



Иванов А.Г., Бахвалов С.В.

УДК 681.142.2

# ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ МЕДИЦИНСКОЙ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ СИСТЕМЫ

Проблема профессиональных нейроинтоксикаций (ПНИ) у работников химических промышленных предприятий в Иркутской области актуальна и имеет социально-экономическую значимость. Существующие методики диагностики и прогнозирования ПНИ нуждаются в совершенствовании автоматизированной инфраструктуры, реорганизации процессов обработки информации, во внедрении программных методов и средств обеспечения поддержки регистрационных и аналитических задач. Для решения означенной проблемы НИИ медицины труда и экологии человека (МТ и ЭЧ) ВСНЦ СО РАМН г. Ангарска и Иркутский Государственный Технический Университет (ИрГТУ) объединили усилия и приступили к научно-исследовательской работе, целью которой является разработка и выбор формальных методов и средств поддержки интеллектуальных процессов диагностики и прогнозирования ПНИ у работающих, создание проекта комплексной информационно-аналитической автоматизированной медицинской системы (ИАС), реализующей информационную технологию, предложенную в [1], а также создание научного прототипа экспертной системы (ЭС), обеспечивающего поддержку принятия диагностических решений в рамках проблемы. Исследования проблемной области, специфики предполагаемого режима эксплуатации ИАС, сбор и анализ требований к системе позволили выработать ряд опорных решений в определении архитектуры программного обеспечения комплексной ИАС.

Решение задачи построения архитектуры комплексной медицинской ИАС основывается на внешнем нормативном описании системы и включает идентификацию ряда подчиненных решений, связанных с разработкой статической и динамической структурных моделей, с приведением совокупных ограничений, накладываемых на ПО, модульной декомпозицией, моделей взаимодействия уровней и компонентов, и другого.

Проектируемая ИАС является многопользовательской. На этапе анализа выявлены следующие функциональные роли [1]:

1. Врач-профпатолог - специалист, выполняющий подготовку предварительных диагностических заключений по ПНИ;

2. Клиницист - узкий специалист, выполняющий узконаправленное исследование здоровья пациента и устанавливающий частный диагноз в соответствующей области;

3. Когнитолог - инженер по знаниям, создает и поддерживает базу знаний ЭС; выполняет извлечение и формализацию знаний экспертов;

4. Инженер-математик — данные специалисты выполняют верификацию поставляемых в ИАС данных, а также их статистическую обработку, применяют математические методы анализа медицинских данных;

5. Специалисты в области гигиены труда (Врачи-гигиенисты) — выполняют анализ и оценку условий труда и быта работающих, поставляют данные в ИАС;

6. Невролог-профпатолог поликлинического отделения - врачи клиники НИИ МТиЭЧ выполняющие скрининговое обследование работающих, готовят и передают в ИАС предварительные заключения о состоянии здоровья работающих;

7. Системный администратор — выполняет постоянную техническую поддержку ИАС, оказывает пользователям помощь при работе в ИАС;

8. Разработчик — создает, модифицирует и поддерживает ИАС на уровне кодов приложений и объектов баз данных;

9. Эксперт в области профпатологии нервной системы — привлекается в качестве источника эвристических знаний для ИАС.

10. Социолог — выполняет оценку социально-бытовых условий жизни и социально-психологических характеристик работников;



Пользовательский состав условно разбивается на два класса: *непосредственные* (1-4, 7,8) — взаимодействующие с программной средой ИАС и обеспечивающие прикладные сервисы ИАС и *внешние* (5,6,9,10) — взаимодействующие с ИАС через непосредственных пользователей для предоставления медицинских производственно-гигиенических, социологических данных.

Результаты проведенного анализа требований к ИАС, а также полученных на их основании функциональных и эксплуатационных спецификаций, позволили зафиксировать ряд фундаментальных требований к программной архитектуре ИАС:

– *Использование многослойной ярусной архитектуры для программной системы*, каждое из приложений которой включает либо использует ряд изолированных уровней кода с восходящей вертикальной зависимостью.

– *Возможность параллельного исполнения программ*, жизненный цикл основного комплекса которых определяется и управляется контейнерами сервера приложений. При этом целесообразно отображение данного требования в модели MVC (Model-View-Controller, Модель-Вид-Контроллер) [2];

Также сформулирован ряд общих требований к программной реализации, создаваемой ИАС: *стандартизация решений*; *масштабируемость*; *гибкость* (замена компонент одного программного слоя или яруса не должны влечь необходимости внесения изменений в компоненты других слоев); *конфигурируемость*, в т.ч. способность к легкому изменению состава компонент, в том числе без останова сервисов и приложений, в «горячем» режиме; *управляемость* (архитектура ИАС должна всегда оставаться доступной для контроля, учета, планирования); *обеспечение кросс-платформенности системы*; *обеспечение надежности системы* на уровне архитектуры системы; *обеспечение производительности системы* (за счет поддержки распределенных вычислений и многопоточности); *поддержка защиты информации (ЗИ)*. Архитектура должна способствовать реализации методов ЗИ.

На основании сформулированных требований в качестве основной программной технологии для ИАС выбирается технология Java2 от компании Sun Microsystems, в качестве архитектурной платформы - J2EE (Java 2 Enterprise Edition), на сегодняшний день имеющая статус одного из признанных мировых

стандартов в области разработки сложных корпоративных приложений. Архитектура системы, созданной на J2EE, способна полностью удовлетворять сформулированным требованиям [3]. Кроме этого, достоинствами платформы J2EE, являются: полная документированность, наличие ряда базовых программных средств — серверов приложений, интегрированных сред разработки, компонентных контуров — приобретение и использование которых не влечет прямых финансовых затрат, и, вместе с тем, обеспечивает независимость разрабатываемых приложений от аппаратно-программных платформ.

Для построения основной серверной программной инфраструктуры аналитических и регистрационных приложений ИАС было выбрано следующее базовое программное обеспечение:

– Базовый набор java-разработки JDK (Java Development Kit) v5.0 Update 15, в составе имеющий виртуальную java-машину (JVM — Java Virtual Machine), необходимую для работы всей java-инфраструктуры ИАС;

– Сервер приложений JBossAS v4.2.0.GA, предназначен для развертывания функциональной и представительной логики приложений ИАС; характеризуется открытым кодом (Open Source) и общественной лицензией (GPL — General Public License)

– Сервер баз данных Oracle 10g XE (свободно распространяемая версия), предназначен для хранения медицинских, гигиенических, производственных данных, вовлеченных в информационные процессы ИАС;

В качестве контура повторно используемых компонентов определяются:

– средство JBoss Seam 2.0.1 GA — каркас, предоставляющий компонентную модель для использования в приложениях JBossAS на любом из ярусов системы.

– J2EE-совместимый каркас GWT (Google Web Toolkit) framework, предназначенный для разработки графических интерфейсов корпоративных приложений на основе технологии AJAX (Asynchronous JavaScript and XML).

Для компоновки API (Application Programming Interface, интерфейс прикладного программирования) ядра ЭС используются два базовых программных продукта:

– Jess (Java Expert System Shell) Rule Engine v7.0p1 — полнофункциональная оболочка экспертной системы, продукт компании Sandia National Laboratories], включающий набор функциональных классов



Java, расширение JessDE для платформы Eclipse, предоставляющее визуальную оболочку для разработки и тестирования базы знаний; Jess Rule Engine может быть использован как самостоятельное приложение, и как набор Java API для интеграции во внешний код Java.

– NPX FuzzyJ Toolkit v1.10 – система Java API для представления и управления нечеткой информацией, реализующая аппарат нечеткой логики и нечетких множеств Л.Заде

Для поддержки процессов разработки и сопровождения ИАС выбираются:

– Среда разработки Red Hat Developer Studio v1.0.0 CR1, основанная на платформе Eclipse, интегрированная с сервером JBossAS.

– Eclipse-совместимое средство функционального расширения JessDE, предоставляющее визуальную среду для разработки и поддержки базы знаний;

– Eclipse-совместимое средство функционального расширения GWT Designer, предназначенное для разработки и поддержки графических пользовательских интерфейсов.

Рассмотрим распределение функций, выбранного программного обеспечения по ярусам, разрабатываемой системы. При этом выделяются ярусы: представления и клиентского уровня, функциональной логики, интерфейсов ядра подсистемы ЭС, доступа к данным.

Результаты анализа требований к ИАС, а также исследование специфики ее проектируемых функций, позволили сформулировать для пользовательских интерфейсов ряд обобщенных нормативных характеристик:

- *Функциональное соответствие решаемым задачам.* Даже поверхностный анализ активности пользователей в ИАС определяет достаточно сложные сценарии их взаимодействия с программной средой, что требует разработки графического интерфейса с событийной моделью адекватной сложности;

- *Стандартность и типизация.* Архитектура и дизайн интерфейсов для всех медицинских приложений к ИАС, охватывающие ее клиентский и представительский уровни, объединены определенным типовым решением, универсальной технологией, принятой в качестве внутреннего стандарта.

- *Прозрачность и гужественность.*

- *Интерактивность в режиме реального времени.*

Анализ приведенных выше и иных смежных требований позволяет определить следу-

ющие решения. Основной технологией взаимодействия между клиентским и представительским ярусами ИАС определяется AJAX, использующая протоколы http и https для организации передачи асинхронных сообщений между клиентом и сервером веб-приложения. Особенность AJAX состоит в отказе от прямого http-диалога веб-обозревателя и веб-сервера, и замене его использованием объекта XMLHttpRequest, предоставляемого моделью обозревателя и управляемого кодом JavaScript на стороне клиента. Данный код и составляет высококачественный пользовательский интерфейс, предоставляющий пользователю развитую событийную модель, обеспечивающий непрерывность его работы и организующий все взаимодействия с серверной частью путем отправки запросов XMLHttpRequest и обработки откликов сервера в функциях обратного вызова.

Для практической реализации AJAX-интерфейсов был выбран инструмент GWT, позволяющий выполнить цикл разработки кода уровней клиента и представления, включая отладку и тестирование, на языке Java2, в выбранной визуальной среде разработки. Развертывание включает компиляцию клиентской части из Java2 в JavaScript и HTML. Серверная часть – удаленный GWT-сервис - формируется в виде jar-библиотеки, содержащей Java-класс сервлета, отвечающего за взаимодействие между процессами соответствующего приложения на уровне функциональной логики и соответствующим клиентским интерфейсом.

На рис. 1 приводится модель, комбинирующая представление процесса компиляции и развертывания GWT-программы, а также логической организации удаленного GWT-сервиса и взаимодействия между клиентским и представительским уровнями.

На ярусе функциональной логики в соответствии каждой прикладной функции, выявленной в областях применения ИАС для пользователей разных типов, ставится подмножество сеансовых компонентов на сервере приложений, образующих соответствующий сервис пользователя. В основе каждого такого сервиса находится центральный сеансовый компонент с поддержкой состояний (EJB3 stateful session bean), обеспечивающий базовую логику сервиса и сохранение контекста между клиентскими соединениями. Запросы пользователя о реализации специфической логики приложений, такой, как, например, многомерный анализ медицинских данных для

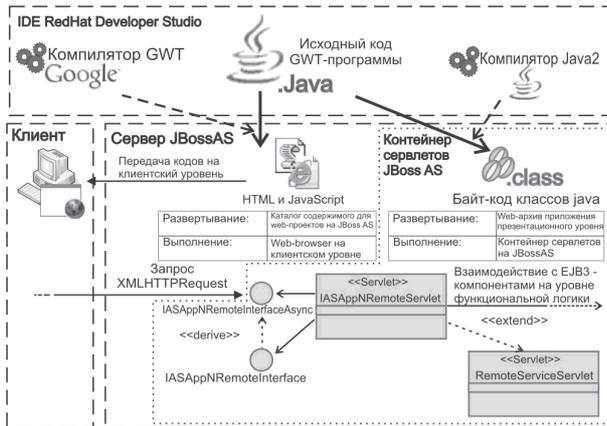


Рис. 1. Концепция архитектуры клиентского и представительского ярусов ИАС

пользователя группы 4, выполняются обращением к библиотекам компонент, расширяющих функциональность базового сервиса и реализуемых в виде сессионных компонентов без поддержки состояний (EJB3 stateful session bean). Доступ к компонентам-сущностям (EJB3 entity beans) уровня постоянства данных (data persistence) выполняется из центрального сессионного компонента для каждого из прикладных сервисов. Доступ к каждому из сессионных компонентов уровня логики обеспечивается соответствующими сервлетами удаленных GWT-сервисов представительского яруса. Для сессионных компонентов применяется управление жизненным циклом от контейнера.

Пользователь-администратор, помимо Web- и JMX (Java Management Extension)-консолей, также использует специальный сервис, который включает необходимую функциональность для управления доступом пользователей ИАС к различным ее процессам, компонентам и сервисам, и реализует иные административные задачи. Кроме того, как видно из рис. 2, разграничение доступа к компонентам сервисов поддержано на уровне архитектуры ИАС: пользователь каждого типа может использовать только соответствующий набор компонентов.

Сверху вниз следуют цепочки компонентов для комплекса приложений медиков - узких специалистов, врача-диагноста ПНИ, системного администратора и пользователей группы математической обработки. Интерфейсы и сервисы в ИАС инженера по знаниям и разработчика компонуется базовыми приложениями.

Все события в ИАС регистрируются для нужд администрирования. Управляемый JMS (Java Message Service)-сообщениями компо-

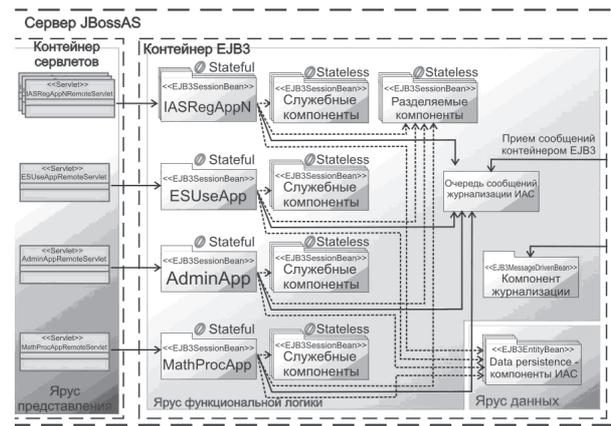


Рис. 2. Концепция архитектуры яруса функциональной логики ИАС

нент журнализации регистрирует в системном журнале сообщения, посылаемые сеансовыми stateful-компонентами в специальную очередь. Передача сообщений управляется EJB3-контейнером.

Классы, задействуемые приложением экспертной системы (далее ЭС) включаются в состав ear-архива комплекса приложений ИАС и развертываются на сервере JBoss AS в качестве API, составляющего функциональное ядро ЭС. Подсистема, компонуемая классами Jess Rule Engine и NRC FuzzyJ Toolkit, представляется завершенным, полнофункциональным решением для API обсуждаемой ЭС в силу следующих свойств[4]:

- Согласно определению задач, возложенных на компонент ЭС и реализуемых на уровне функциональной логики в ИАС, каждая из таковых получает в соответствие исчерпывающий набор функций, реализованный программно, в классах API;
- Предлагаемые средства изначально совместимы на уровне API и внутренних объектов, и предназначены для работы — концептуально — в одной проблемной области, и — программно и логически - в системной взаимосвязи;
- Объекты API предлагаемых средств возможно использовать в качестве единого интерфейса к ядру ЭС, включая ссылки на соответствующие классы в код приложения уровня бизнес-логики. На рисунке 3 приводится модель организации единого API ядра для ЭС.

Программный аппарат применения нечеткой логики и нечетких множеств, предоставля-

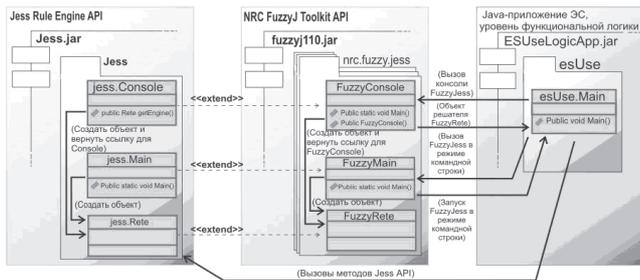


Рис. 3. Модель организации API-ядра ЭС его интеграции в ИАС

емый NRC FuzzyJ Toolkit, может быть использован автономно от Jess, равно как и средства Jess могут быть применены без привлечения поддержки нечеткой информации, однако целесообразно совместное использование этих средств, как слоистой системы. Средство и технология, предоставляемые в этом случае, имеют рабочее название FuzzyJess: фактически, логика системы Jess расширяется возможностями корреспондирующих классов системы NRC FuzzyJ Toolkit [4].

Архитектура яруса данных ИАС трехслойна и состоит из компонентов-сущностей, комплекса API доступа к данным в составе JBossAS, а также J2EE и JDBC-совместимого сервера баз данных Oracle 10g XE. Компоненты-сущности определены в рамках проектирования ИАС и соответствуют крупным автономным сущностям, представляемым логическими записями БД. На рис. 4, иллюстрирующем архитектурный принцип яруса данных ИАС, приводится соответствие компонентов-сущностей четырем крупным прикладным сервисам ИАС. Для компактности модели все выделенные сущности символизируются шестью логически группами, также в зависимостях сессионных компонентов не отображаются home-интерфейсы компонентов-сущностей.

Комплекс Java API яруса данных также трехслоен и последовательно включает: диспетчер персистентности — ORM-средство Hibernate, высокоуровневой интерфейс управления транзакциями JTA и интерфейс соединений с БД — JDBC. Такая инфраструктура позволяет отделить функциональность управления транзакциями, ORM-преобразования и поддержки персистентности от развертываемого кода ИАС, сконцентрировав в компонентах-сущностях последней логику предметной области по обработке данных. Для построения выражений запросов к БД могут быть использованы такие адаптации языка

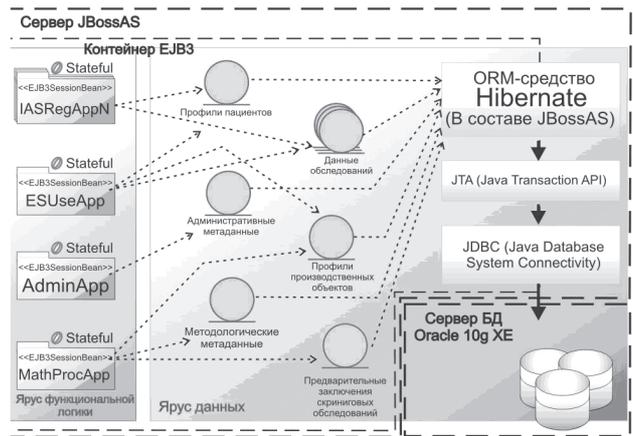


Рис. 4. Концепция архитектуры яруса данных ИАС

SQL, как EJB-QL и HQL. Выбор означенного сервера БД удовлетворяет полученным спецификациям ИАС.

Результирующая модель информационно-аналитической системы, представленная на рис. 5, сочетает представления ИАС как многослойной программной системы с MVC-представлением.

Таким образом, при решении задачи информатизации клинко-диагностического процесса в НИИ МТ и ЭЧ г. Ангарска выполняется проектирование комплексной информационно-аналитической системы, одной из главных функций которой устанавливается экспертная поддержка диагностики и прогнозирования профессиональных нейроинтоксикаций у работников химических предприятий. В рамках проектного процесса предложен и обоснован подход к построению архитектуры программной системы. Полученное решение,

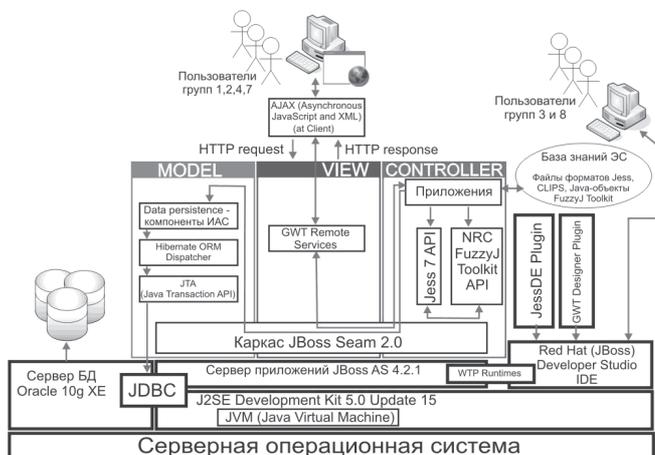


Рис. 5. Многоуровневое MVC-представление информационно-аналитической системы



ключевые элементы которого изложены в настоящей статье, основывается на зарекомендованных практикой стандартах и технологиях программной инженерии, реализует компонентный подход к разработке сложных корпоративных приложений и их комплексов, а также сочетает базу надежности и производительности с экономичностью и минимальной стоимостью владения результирующей системой.

#### БИБЛИОГРАФИЯ

1. Иванов А.Г. Подходы к разработке информационной технологии идентификации и прогнозирования профессиональных нейроинтоксикаций у работающих. /Труды XII Байкальской Всероссийской конференции «Информационные и математические технологии в науке и управлении».
2. Хавар Заман Ахмед, Кэри Е. Амриш, Разработка корпоративных Java-приложений с использованием J2EE и UML.: Пер. с англ.- М.: Издательский дом "Вильямс", 2002.- 272 с.
3. Ноутон П., Шидт Г., Java2: Пер. с англ. - СПб.: БХВ-Петербург, 2001.- 1072 с.
4. Иванов А.Г. Обзор современных программно – языковых средств экспертных систем в контексте задачи проектирования медицинской информационно – аналитической системы идентификации и прогнозирования профессиональных нейроинтоксикаций у работающих. /Труды XIII Байкальской Всероссийской конференции «Информационные и математические технологии в науке и управлении». Часть II. – Иркутск: ИСЭМ СО РАН, 2008. – 116-124 с.

Степаненко А.С.

УДК 159.9:004.8

## СОЦИАЛЬНО-КУЛЬТУРНЫЕ ПЕРЕДПОСЫЛКИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА. СИСТЕМНЫЕ АСПЕКТЫ

Искусственный интеллект возник благодаря социокультурным предпосылкам, которые сложились в рамках европейской культуры.

Выяснению социокультурных предпосылок искусственного интеллекта способствует концепция социокода, обоснованная в трудах ростовского философа и историка науки М.К. Петрова [1, 2]. Методологической основой такой концепции является положение, согласно которому человек в определенном смысле является микрокосмом общества, что выражается в развитии его Я в связи со сменой схем социального кодирования (социокода). Действительно, исследования последних десятилетий по типологии культуры обнаружили существование трех основных структурных ключей социального кодирования, а именно: *лично-именной, профессионально-именной и универсально-понятийной социокоды* [1].

I. *Лично-именной социокод* означает, что приобщение человека к социальной деятельности происходит при помощи имени-различителя, нагруженного определенным набором социальных ролей и обязанностей. Этот вечный ролевой набор транслируется в актах последовательной **передачи имени** от одного к другому. Человек проходит цикл: детство – зрелость – старость, поэтому имя – адрес у него меняется трижды – «трижды рождается и нарекается». В первый раз он как дитя человеческое получает неформальное обучение под руководством взрослых и стариков, во второй раз имя дается в обряде посвящения (это взрослое или охотничье имя несет набор ролевых функций активности члена племени) и в третий раз, доживший до старости, получает стариковское имя с его функцией социальной памяти и трансляции, свое же имя передает молодому. Это тип социокода присущ «пер-