

ISSN 2220-3397

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Программная инженерия



Пр 8
2018
Том 9
ИН

Devprom ALM –

первая российская
программная платформа
для всех стадий
разработки ПО



Devprom ALM
для поддержки учебного процесса
предоставляется бесплатно
на основании академической лицензии,
без ограничения по функциональности
и числу пользователей



Приглашаем к сотрудничеству колледжи и вузы,
обучающие студентов по направлению
"Программная инженерия".
Практические задания, выполняемые
с использованием программной
платформы Devprom ALM,
идеально свяжут теоретические основы,
полученные студентами в рамках курса,
с практическими упражнениями,
выполненными в современной программной среде,
широко используемой в индустрии

Программная инженерия

Том 9
№ 8
2018
Прин

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН
Главный редактор
Васенин В.А., д.ф.-м.н., проф.
Редколлегия
Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назиров Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН,
Отделения нанотехнологий и информационных технологий РАН,
МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

Грибова В. В., Клещев А. С., Москаленко Ф. М., Тимченко В. А., Шалфеева Е. А. Расширяемый инструментарий для создания жизне- способных систем с базами знаний	339
Подковальников С. В., Трофимов И. Л., Трофимов Л. Н. Технологические аспекты геоинформационной вычислительной системы для поддержки исследований в области формирования межгосударственных энерго- объединений	349
Туровский Я. А., Суровцев А. С. Программный пакет для автома- тизированной обработки биомедицинских сигналов для поисковых ис- следований	361
Esmaeilishahmirzadi N., Mortezapour H. Preprocessing for Enhancing the Classification of Pulmonary Data Sets using Convolutional Neural Networks	369
Гриф М. Г., Лукоянычев А. В. Программный комплекс для обучения русскому жестовому языку на основе Unity3D	375

Журнал зарегистрирован
в Федеральной службе
по надзору в сфере связи,
информационных технологий
и массовых коммуникаций.
Свидетельство о регистрации
ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

[Http://novtex.ru/prin/rus](http://novtex.ru/prin/rus) E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2018

SOFTWARE ENGINEERING

PROGRAMMAYA INGENERIA

Vol. 9

N 8

2018

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.), Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCHENKO A. B., Dr. Sci. (Phys.-Math.), Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad. RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.), Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.), Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOVS JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.), Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R. , Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A., Shalfeeva E. A. Extensible Toolkit for the Development of Viable Systems with Knowledge Bases	339
Podkovalnikov S. V., Trofimov L. N. The Data Processing and Geo Information System Technological Aspects to Support Research in the Field of Prospective Interstate Electric Power Interconnections Development	349
Turovsky Ya. A., Surovtsev A. S. Software Package for Automatic Data Processing of Biomedical Signals for Exploratory Researches ..	361
Esmaeilishahmirzadi N., Mortezapour H. Preprocessing for Enhancing the Classification of Pulmonary Data Sets using Convolutional Neural Networks	369
Grif M. G., Lukoyanychev A. V. The Program Complex for Training Russian Sign Language Based on Unity3D	375

В. В. Грибова, д-р техн. наук, зам. дир. по научной работе, e-mail: gribova@iacp.dvo.ru,
А. С. Клещев, д-р физ.-мат. наук, гл. науч. сотр.,
Ф. М. Москаленко, канд. техн. наук, ст. науч. сотр.,
В. А. Тимченко, канд. техн. наук, ст. науч. сотр.,
Е. А. Шалфеева, канд. техн. наук, ст. науч. сотр., Федеральное государственное бюджетное учреждение науки Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток

Расширяемый инструментарий для создания жизнеспособных систем с базами знаний

Описана концепция и представлена архитектура расширяемого инструментария для создания и сопровождения систем, основанных на знаниях, который обеспечивает необходимые условия жизнеспособности таких систем. Обоснована важность обеспечения расширения инструментария как его разработчиками, так и пользователями. Дан анализ принципов и механизмов развития инструментария, которые реализованы на облачной платформе IACPaaS.

Ключевые слова: система, основанная на знаниях (СОЗ), технология разработки СОЗ, сопровождение программных систем, инструментальная поддержка разработки СОЗ, среда разработки, язык описания моделей, редактор моделей, редактор решателей задач, генератор редакторов компонентов

Введение

Обеспечение жизнеспособности программных систем (ПС) является одной из ключевых проблем в области инженерии программного обеспечения, актуальность которой растет с каждым годом в связи с увеличением сложности программных систем. Жизнеспособность ПС рассматривается как сохранение их работоспособности и полезности в условиях изменчивости требований пользователей, предметной области, окружающей среды (новые технологии, платформы, методы и протоколы взаимодействия и др.) [1–3]. Достижение этого критерия ПС обеспечивается через механизмы самоадаптации, заложенные в программе, и/или через сопровождение (при участии человека) [4–7].

Затраты на сопровождение в жизненном цикле ПС согласно обзорам, опубликованным в работе [8], занимают более 70 % от общей стоимости затрат. В некоторых публикациях, например, в работе [9], отмечается, что стоимость сопровождения доходит до 90 % всех затрат. Среди факторов, которые влияют на стоимость сопровождения программного обеспечения и его жизнеспособность, отмечается инструментарий, который используется для сопровождения ПС [8, 10, 11]. Таким образом, рассматривая вопрос об обеспечении жизнеспособности ПС, нельзя игнорировать инструментальные средства их разработки. Такие средства, которые также являются ПС, должны развиваться и совершенствоваться, предоставляя новые и улучшенные возможности пользователям.

Программные системы с базами знаний или системы, основанные на знаниях (СОЗ) — особый класс программных систем, требующий специализированных инструментальных средств разработки и сопровождения.

Целью настоящей работы является описание концепции и представление архитектуры расширяемого инструментария для создания и сопровождения СОЗ, а также их компонентов.

1. Роль расширяемости инструментария в обеспечении жизнеспособности СОЗ

Инструментарий для разработки ПС также является ПС, соответственно, его жизнеспособность также важна, как и жизнеспособность любого другого класса программного обеспечения.

Можно выделить два основных подхода к развитию инструментария. *Первый подход* основан на совершенствовании инструментария его разработчиками. В этом случае инструментарий рассматривается как "черный ящик", в который сами разработчики вносят изменения: исправляют ошибки, дефекты, расширяют функциональные возможности, добавляют новые сервисные функции. *Второй подход* основан на расширении инструментария его пользователями. В этом случае развитие возможно как через создание библиотек повторно используемых компонентов, так и через создание программных модулей, реализующих новые или улучшающих су-

ществующие функциональные возможности самого инструментария. Новые модули создаются либо с использованием предоставляемого пользователем API¹, либо через открытие исходного кода (*open source*), который становится доступным через некоторый публичный репозиторий. Если в первом подходе ответственность за все изменения несут разработчики инструментария, то во втором, как показывает практика, ответственность за сопровождение новых функций, за расширения несут как разработчики, так и пользователи (возможны различные механизмы ответственности).

Наиболее полный анализ необходимости расширения инструментария дан в работе [11]. Он основан на более чем 30-летнем опыте использования системы Protégé. В работе отмечается, что без непрерывного развития инструментария невозможно создание и развитие программного обеспечения на его основе, отвечающего постоянно изменяющимся требованиям.

На основе обзора литературы, прежде всего работ [11–13], а также собственного опыта авторов, можно выделить следующие факторы, требующие расширяемости инструментария и влияющие на жизнеспособность ПС.

Появление новых платформ, технологий, интерфейсов. Данный фактор является ключевым и наиболее существенным. Если программный продукт используется 3...5 лет, то за это время появляются новые платформы (либо их версии), новые технологии, новые типы интерфейсов и их модификаций (новые интерфейсные элементы, стили, стандарты и пр.). Эти факторы, как правило, приводят к изменению требований к конечному программному продукту. Их выполнение невозможно без поддержки соответствующих возможностей в инструментарии, с помощью которого этот программный продукт создан. Более того, разработчики инструментария вынуждены "быть на шаг впереди", чтобы при изменении требований к программному продукту инструментарий уже обладал необходимыми возможностями. Так, например, Protégé, с одной стороны, эволюционировала от Xerox Lisp платформы к NeXTSTEP, Windows (затем была разработана кроссплатформенная Java-версия Protégé, позволяющая создавать гибкую архитектуру различных плагинов); с другой стороны, от локального — к клиент-серверному редактору онтологий. Еще один пример — эффективная и распространенная несколько лет назад технология Flash, с использованием которой было реализовано множество программных продуктов. Однако необходимо отметить, что в данный момент технология не поддерживается разработчиками. Такая ситуация приводит к тому, что многие продукты перестают использоваться, приходится разрабатывать новые. Наконец, постоянное усовершенствование средств пользовательских интерфейсов требует непрерывного усовершенствования инструментария, чтобы

в ответ на новые запросы конечных пользователей инструментарий позволял поддерживать такие запросы.

Новые научные разработки (модели, методы и подходы). Наука не стоит на месте, и ее достижения активно внедряют в практику. Если научные разработки направлены на снижение трудоемкости, повышение качества продукта, то они однозначно будут востребованы в новых версиях инструментария. В противном случае разработчикам будет выгоднее разработать новый продукт с использованием другого, более современного инструментария, чем затрачивать огромные ресурсы на сопровождение старого. Прежде всего, к таким научным результатам относятся новые методы программной инженерии (например, методы коллективной разработки программных компонентов), методы инженерии знаний (новые языки, модели и формализмы представления знаний). Так, например, авторы Protégé отмечают включение в инструментарий (как ответ на потребности пользователей-разработчиков) новых методов оценивания онтологий [14], выравнивания онтологий [15], визуализации онтологий [16], эволюцию моделей и формализмов представления Protégé от фреймового подхода, RDF(S) к OWL.

К факторам, требующим развития инструментария, также следует отнести *влияние и положительный пример других успешных проектов и инфраструктур*. Например, авторы Protégé отмечают повлиявшую на них исследовательскую инфраструктуру KADS, редакторы онтологий, разработанные другими группами, такими как OilEd. Эти факторы опосредованно, но влияют на развитие инструментария: если какие-то функции более удобно реализованы у конкурентов (более дружественный интерфейс, наличие средств автоматизации проектирования и др.), то по причине высокой стоимости сопровождения зачастую легче разработать новую программную систему, чем продолжать сопровождать существующую.

Наконец, *отзывы и запросы пользователей* ("отзывы, которые мы получаем от наших пользователей, приводят к новым функциям в Protégé" [11]). При этом важно обеспечить возможность развития инструментария самими пользователями. Создатели Protégé отмечают, что рост сообщества Protégé привел к появлению новых плагинов Protégé, а развитие электронных средств массовой информации помогает им в совершенствовании созданных ими компонентов.

В настоящее время помимо Protégé существует широкий спектр различных специализированных инструментов для разработки СОЗ и их компонентов. Типичными представителями данного класса систем являются: Level5 Object, G2, Clips, Loops, VITAL, KEATS, OSTIS, AT-ТЕХНОЛОГИЯ, RT Works, COMDALE/C, COGSYS, ILOG Rules и др. Отличия различных инструментальных систем связаны с набором предлагаемых разработчику компонентов; с уровнем поддержки этапов жизненного цикла создания и сопровождения СОЗ; с различными технологиями создания; с используемыми формализмами представления знаний и способами

¹ API — Application Programming Interface.

их формирования, отладки и связывания; с механизмами вывода; со средствами формирования пользовательских интерфейсов.

Необходимо констатировать, что все долгоживущие инструментальные средства активно развиваются, постоянно предлагая новые модули и расширения. Например, эволюция развития АТ-ТЕХНОЛОГИИ [17] связана с созданием дополнительных служебных интерфейсных и конфигурационных компонентов, подсистем интеграции с прикладными программами, упрощающих, прежде всего, развитие созданных на ее основе прикладных систем. Система G2 [18] постоянно дополняется новыми методами отладки знаний, формализмами их представления, новыми проблемно-ориентированными расширениями, включая G2 Optegrity, G2 NeurOn-Line, ReThink, DSP, GST, Fault Expert, BatchDesign_Kit. Они представляют собой проблемно- и задаче-ориентированные оболочки экспертовых систем для решения классов задач диагностики, мониторинга, управления, планирования и т. п. в различных предметных областях, упрощая создание и сопровождение систем определенных классов. Большинство систем содержит библиотеки повторно используемых компонентов.

Отмечая, бесспорно, значительные успехи разработчиков инструментария в обеспечении расширяемости создаваемых ими инструментов, можно выделить следующую основную проблему, тормозящую расширение инструментария его пользователями (разработчиками СОЗ). Средства расширения инструментария и средства создания СОЗ основаны на разных методах, технологиях и, как правило, языках программирования. Основная причина такой ситуации связана с тем, что инструментарий для разработки СОЗ не является СОЗ, поэтому используются различные принципы их создания. Это обстоятельство является существенным ограничением и в значительной мере усложняет разработчикам СОЗ задачу расширения инструментария в направлении создания новых технологий, сервисных и системных функций. Исключение составляют библиотеки повторно используемых компонентов, которые также являются средствами расширения инструментария.

Таким образом, комплексное решение проблемы расширяемости инструментария означает его развитие разработчиками и пользователями через создание библиотек повторно используемых компонентов, оболочек экспертовых систем, новых сервисных функций, технологий, средств реализации интерфейса и др. Развитие такого подхода остается важной и актуальной задачей.

2. Основные требования и концептуальная архитектура инструментария для разработки жизнеспособных СОЗ

Выделим следующие ключевые требования к инструментарию для разработки СОЗ.

Развитие инструментария должно осуществляться как основной командой разработчиков инструмен-

тария, так и его пользователями, участвующими в разработке СОЗ. Такое участие может проявляться в создании библиотек повторно используемых компонентов, новых технологий, инструментальных средств поддержки разработки компонентов СОЗ, включая оболочки экспертизных систем.

Наличие единой среды и инструментальной поддержки для развития инструментария и для разработки и сопровождения непосредственно СОЗ. Это требование важно, во-первых, потому, что в результате разработки СОЗ какие-то ее элементы могут напрямую использоваться как расширение инструментария (наиболее очевидный пример — библиотеки повторно используемых компонентов). Во-вторых, разработчики СОЗ могут создать новую "надстройку" для решения своих задач, которая может быть полезна не только им, но и другим потенциальным пользователям инструментария, например, новую технологию и ее инструментальную поддержку, новую оболочку, новый набор инструментов разработки и др. Отдельно стоит вопрос сопровождения разработанных компонентов. Как правило, применяют два основных подхода к такому сопровождению:

- ответственность за сопровождение несут разработчики инструментария, в этом случае по договоренности с разработчиком компонента определяются условия передачи (цена, ответственные за сопровождение и т. д.);
- ответственность за сопровождение несут разработчики компонента и они же определяют условия использования.

Учитывая изложенное выше, предлагается трехуровневая *архитектура расширяемого инструментария* для разработки СОЗ, состоящая из *инструментального ядра, базового и расширяемого наборов инструментальных средств* (далее — *базовый и расширяемый инструментарий*). Рассмотрим эти уровни подробнее.

1. *Инструментальное ядро.* Инструментальное ядро определяет основной принцип построения компонентов как самого инструментария, так и СОЗ (созданных с помощью инструментария), и включает необходимый для этого набор средств (рис. 1). *Основной принцип* заключается в том, что программные и информационные компоненты СОЗ формируются на основе *двухуровневого подхода*. В процессе его реализации сначала формируется декларативная модель компонента, описывающая структуру и набор контекстных условий. Затем такая модель управляет разработкой множества соответствующих компонентов.

Для описания моделей компонентов, независимо от назначения, используется единый декларативный язык — *язык описания моделей*. Для упрощения создания моделей на этом языке в состав ядра входит *редактор моделей*, который позволяет разработчикам создавать их в наиболее простой и удобной форме. В ядре содержится *генератор редакторов компонентов*, предназначенный для автоматической генерации редакторов декларативных компонентов по их моделям. Задачами *генератора редакторов*

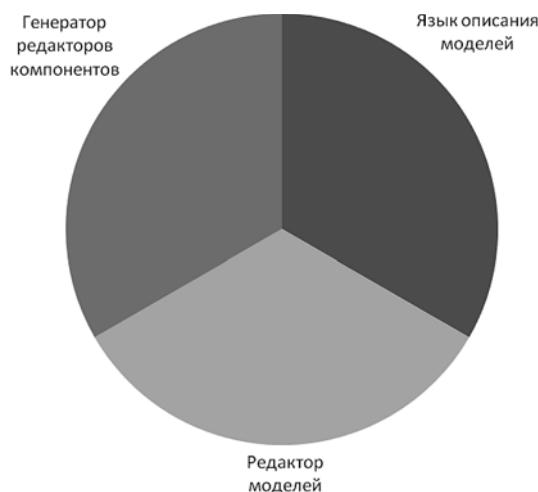


Рис. 1. Структура инструментального ядра

компонентов являются: генерация пользовательского интерфейса и сценария формирования компонента с проверкой заданных в модели контекстных условий, полноты его формирования.

Инструментальное ядро достаточно для создания информационных компонентов по их структурным моделям и управления ими (а также для спецификации программных компонентов любого уровня — создания их декларативных моделей).

Базовый инструментарий. Базовый инструментарий включает набор моделей и программных средств (рис. 2), достаточных для создания СОЗ по базовой технологии и управления ими (включая набор средств инструментального ядра). Поскольку все компоненты формируются по их структурным декларативным моделям, соответственно, данный уровень инструментария включает редакторы для создания компонентов. Такие компоненты формируются *генератором редакторов компонентов* для каждой структурной модели. Кроме указанных выше элементов этот уровень содержит систему управления и



Рис. 2. Структура базового инструментария

внешние программные элементы. Эти элементы используются для создания пользовательского интерфейса СОЗ и императивной части тех компонентов, где декларативного представления недостаточно для задания необходимых функциональных механизмов. Сопровождение элементов базового инструментария осуществляется его *разработчиками*.

3. **Расширяемый инструментарий.** В расширяемый инструментарий (рис. 3) могут включаться *новые технологии разработки СОЗ и их инструментальная поддержка*. К последней относятся: *средства коллективной разработки, оболочки СОЗ (универсальные и специализированные)*, *средства создания адаптивных мультимодальных интерфейсов, библиотек повторно используемых компонентов* (информационных и программных).

Расширение может осуществляться с использованием инструментального ядра, базового инструментария, а также за счет средств и инструментальных механизмов расширяемого инструментария. Таким образом, достигается рекурсивное использование разработанных компонентов расширяемого инструментария. Данный уровень инструментария может расширяться как *разработчиками инструментария, так и его пользователями* (экспертами, инженерами знаний, программистами, аналитиками).

Трехуровневая архитектура представленного выше инструментария обладает следующими основными свойствами:

- единые принципы создания СОЗ (компонентов СОЗ) и инструментария для их разработки;
- двухуровневый подход к формированию компонентов (сначала формируется структурная декларативная модель, затем по ней создается необходимый компонент);
- единый язык и редактор для формирования моделей всех компонентов;
- единый унифицированный внутренний формат всех компонентов;
- автоматическая генерация редакторов для создания компонентов по их моделям.

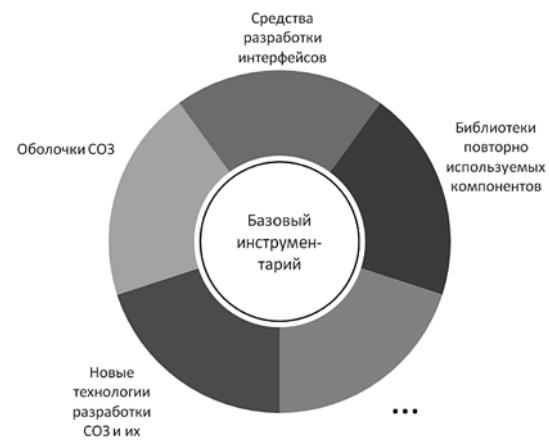


Рис. 3. Структура расширяемого инструментария

Данная архитектура положена в основу облачной платформы IACPaaS [19, 20]. Дальнейшие разделы посвящены детальному описанию состава и структуры каждого уровня архитектуры с особенностями ее реализации на платформе IACPaaS.

3. Инструментальное ядро

Инструментальное ядро состоит из трех базовых компонентов: *языка описания моделей*; *редактора моделей*; *генератора редакторов компонентов*.

Язык описания моделей. Основным компонентом всего инструментария является язык описания моделей. В его создание были заложены следующие основные идеи:

- язык должен быть декларативным, достаточно простым, и в то же время мощным, позволяющим описывать произвольные модели, ориентированные и адаптированные к терминологии и форме, принятой у разработчиков компонентов СОЗ и инструментария для их создания;
- формирование необходимых компонентов разработчиками должно выполняться "сверху вниз", с переходом от общих понятий к деталям, которые уточняют общие понятия и сущности, само описание при этом должно быть целостным, а не "раздробленным" на отдельные сущности с возможностью связывания компонентов между собой;
- поддержание соответствия между моделью и компонентом, формируемым на ее основе.

Язык описания моделей предоставляет средства спецификации моделей в форме связных размеченных корневых иерархических бинарных ориентированных орграфов (далее — орграфов) с возможными петлями и циклами. Разметку имеют как вершины орграфа, так и его дуги. Разметка определяет в том числе семантику правил формирования (создания и модификации) компонентов, накладывая ограничения на их структуру и содержание.

Все вершины орграфа модели можно разбить на два непересекающихся подмножества. Первое подмножество составляют вершины, имеющие поле "имя", значением которого является строка конечной длины, и поле "сорт", значением которого является название сорта. Ко второму подмножеству относятся вершины, имеющие поле "значение", значением которого является константа некоторого сорта.

Каждый сорт характеризуется названием и множеством констант этого сорта. Базовые сорта имеют следующие названия:

- *строковое*, константами которого являются строки конечной длины;
- *целое*, константами которого являются целые числа;
- *вещественное*, константами которого являются вещественные числа;
- *логическое*, константами которого являются "истина" и "ложь";
- *бинарные данные*, константами которого являются массивы байтов конечной длины.

Две вершины, являющиеся началом и концом дуги соответственно, могут принадлежать как одному, так и разным орграфам. Для каждой нетерминальной вершины определен *тип набора исходящих дуг*, а именно — "список" или "альтернатива".

Каждая дуга имеет *спецификатор* и *ограничитель*, которые в совокупности с типом набора исходящих дуг определяют правила формирования орграфов компонентов. Значениями ограничителя могут быть: "создание новой вершины"; "ссылка на существующую вершину"; "создание новой вершины — зависимого клона"; все возможные комбинации этих значений, включая пустую (всего восемь возможных значений). Значениями спецификатора могут быть: "копия"; "возможное отсутствие"; "пропуск"; "в точности один"; "непустой список"; "непустое упорядоченное множество"; "непустая последовательность"; "ноль или один"; "возможно пустой список"; "возможно пустое упорядоченное множество"; "возможно пустая последовательность" (значения между собой не комбинируются).

Подробное описание языка приведено в работах [21, 22].

Редактор моделей. Основное назначение редактора моделей заключается в том, чтобы упростить разработку моделей компонентов, обеспечив их интерактивное формирование, не заставляя разработчиков изучать синтаксис языка описания моделей. Разработчик при таком редактировании должен выбирать допустимые в каждом конкретном контексте разработки операторы для формирования необходимой модели. Редактор при этом должен быть ориентирован на широкий круг пользователей, к числу которых могут относиться инженеры знаний (когнитологи) и эксперты предметной области (при разработке СОЗ они формируют модель знаний или ее онтологию), программисты, дизайнеры интерфейса, аналитики и другие разработчики.

Генератор редакторов компонентов. Разработка специализированных редакторов — отдельная трудоемкая задача. Универсальный язык и унифицированное внутреннее представление как моделей, так и отдельных их компонентов, позволяет применить единые принципы генерации редакторов всевозможных по типу компонентов СОЗ. В основу редактора положен процесс порождения компонента СОЗ по его модели. Процесс порождения (формирования) компонента обеспечивает соответствие между этими орграфами и осуществляется "сверху вниз". Процесс редактирования может быть завершен по желанию пользователя в любом состоянии. Порожденный компонент может при этом оказаться формально полным или неполным относительно орграфа его модели. Редактирование орграфа компонента возможно в пределах, которые не нарушают соответствие между орграфами.

Средств инструментального ядра достаточно для создания онтологий, баз знаний и баз данных (с графовой моделью данных). В качестве модели компонента выступает проблемно-ориентированная

модель знаний (или онтология знаний), которая учитывает специфику организации знаний и данных в конкретной предметной области.

4. Базовый инструментарий

Основные задачи базового инструментария заключаются в следующем:

- предоставить разработчику набор средств создания программных компонентов СОЗ и инструментальных сервисов;
- обеспечить управление процессом создания программных компонентов, включая их сборку, связывание с информационными компонентами (например, базой знаний, базой данных, разработанных с использованием инструментального ядра), запуск на выполнение;
- обеспечить организацию инфраструктуры на всех уровнях инструментария.

Платформа IACPaaS реализует агентный подход к созданию решателей. Это означает, что решатель задач состоит из множества агентов, которые обмениваются сообщениями. В соответствии с двухуровневым подходом (сначала разрабатывается модель компонента, а затем компонент), базовый уровень должен включать модели агента, шаблона сообщений и решателя. Учитывая, что для компонентов всех типов используется единый язык для описания моделей, базовый инструментарий включает единые

(универсальные) модели соответствующих компонентов, освобождающие разработчиков компонентов от их создания (рис. 4). Для организации запуска решателей с конкретными наборами входных и выходных данных выделяется также модель сервиса. Модели компонентов и редакторы для их формирования созданы средствами инструментального ядра. Агенты и шаблоны сообщений помимо декларативной части имеют императивный компонент, поэтому в инструментарий включены *среда разработки Java* и *среда разработки JavaScript*. Для разработки пользовательского интерфейса СОЗ используется (по выбору разработчика) Java, HTML, JavaScript. Расширение возможностей этой составляющей инструментария определяется расширяемостью соответствующих библиотек. В настоящее время расширение этого уровня инструментария осуществляется разработчиками платформы.

Таким образом, базовый инструментарий состоит из набора редакторов (*редактора сервисов*; *редактора решателей задач*; *редактора агентов*; *редактора шаблонов сообщений*), системы управления и двух внешних компонентов, а именно *среды разработки Java* и *среды разработки JavaScript*.

5. Расширяемый инструментарий

Расширяемый уровень инструментария, в первую очередь, предназначен для разработчиков СОЗ, кото-

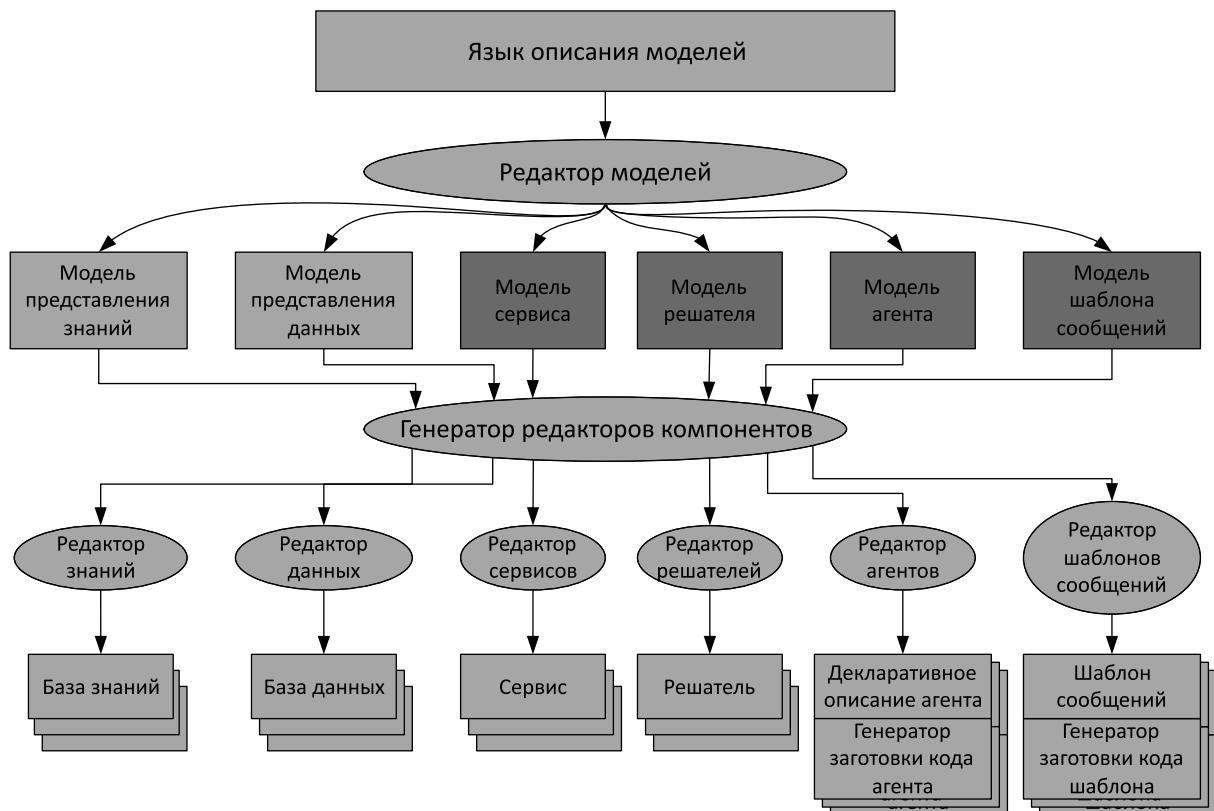


Рис. 4. Использование языка описания моделей для представления декларативной части программных компонентов

▼ сервис по базовой технологии {СПИСОК} (! 'one') (new) ⚡

- ◊ имя сервиса (сорт: Строковое) (! 'one') (new)

▼ требования к информации {СПИСОК} ([=] 'copyumm') (ref-new) ⚡

- ◊ требование к структуре (сорт: Строковое) ([+]'setrnm') (all)
- ◊ требование к содержимому (сорт: Строковое) ([+]'setrnm') (all)

▼ требования к решателю {СПИСОК} ([=] 'copyumm') (ref-new) ⚡

- ◊ нефункциональное требование (сорт: Строковое) ([+]'setrnm') (all)
- ◊ функциональное требование (сорт: Строковое) ([+]'setrnm') (all)

▼ метаинформация входных ИР {СПИСОК} ([=] 'copyumm') (ref-new) ⚡

- ◊ метаИР входного ИР (сорт: Строковое) ([+]'setrnm') (ref)

▼ НОВЫЙ метаИР ([+]'setrnm') (ref-new) ⚡

- потомки отсутствуют

▼ роль пользователя сервиса {СПИСОК} ([+]'setrnm') (ref-new) ⚡

- ◊ задача пользователя (сорт: Строковое) ([+]'setrnm') (all)

▼ метаинформация выходных ИР ([=] 'copyumm') (ref-new) ⚡

- потомки отсутствуют

▼ решатель {АЛЬТЕРНАТИВА} (= 'copy') (ref-new) ⚡

▼ НОВЫЙ решатель (! 'one') (ref-new) ⚡

- потомки отсутствуют

◊ имя решателя (сорт: Строковое) (! 'one') (ref)

▼ входные ИР {СПИСОК} (= 'copy') (ref-new) ⚡

- ◊ имя ИР (сорт: Строковое) ([+]'setrnm') (ref)

▼ НОВЫЙ ИР ([+]'setrnm') (new) ⚡

- потомки отсутствуют

▼ ПРОВЕРИТЬ СООТВЕТСТВИЕ (= 'copy') (ref-new) ⚡

- потомки отсутствуют

▼ Выходные ИР (= 'copy') (ref-new) ⚡

- потомки отсутствуют

▼ ИНТЕГРИРОВАТЬ решатель с ИР (= 'copy') (ref-new) ⚡

Рис. 5. Интерфейс сервиса управления программным проектом по базовой технологии



Рис. 6. Структура портала знаний по медицине

рые могут включать в его состав новые, удобные инструментальные средства для сопровождения СОЗ; новые технологии разработки СОЗ (включая их инструментальную поддержку); специализированные либо универсальные оболочки экспертизных систем. В настоящее время в расширяемый инструментарий включена разработка СОЗ по двум специализированным технологиям: на основе расширенных графовых грамматик [23], а также технологии разработки СОЗ на основе механизмов виртуальной реальности [24]. Пользователями платформы предложены также расширенные средства инструментальной поддержки разработки СОЗ: сервис управления программным проектом по базовой технологии IACPaaS (рис. 5) и инструментальный сервис коллективной разработки СОЗ.

Библиотека повторно используемых компонентов реализована в виде тематических порталов знаний, в которых содержатся онтологии, базы знаний, специализированные решатели, прикладные сервисы, агенты и шаблоны сообщений. Пользователи могут создавать новые порталы знаний в соответствии с необходимой им структурой. На рис. 6 приведен пример структуры портала знаний по медицине.

Заключение

Необходимым условием жизнеспособности любой ПС является наличие расширяемого инструментария, с помощью которого она создается. Чем более долгим является жизненный цикл СОЗ, тем очевидней потребность разработчиков в совершенствовании механизмов их адаптации к изменениям требований пользователей, предметной области, условий эксплуатации. Пользователи инструментария могут совершенствовать его, если средства расширения максимально приближены к средствам разработки самих СОЗ, а трудоемкость этого процесса сопоставима со сложностью разработки соответствующих компонентов СОЗ. Учитывая, что разработчиками СОЗ являются не только программисты, но и инженеры знаний, эксперты, в данном инструментарии заложены механизмы развития, основанные на унифицированном, едином декларативном представлении информационных и программных компонентов с последующей автоматической генерацией редакторов для их формирования.

Все описанные в настоящей работе принципы реализованы на облачной платформе IACPaaS. Более чем пятилетний срок использования данной платформы различными пользователями показал, что потребность в развитии инструментария активно реализуется пользователями. При этом разработанные "для своих нужд" инструментальные средства разработки, а также различные компоненты — онтологии, базы знаний и данных, шаблоны сообщений, агенты, решатели и сервисы как используются этими же пользователями в других разработках, так и становятся доступными для других его пользователей, что реализует на практике компонентный

подход к разработке СОЗ из повторно используемых составляющих. Более того, пользователями платформы создан универсальный и общедоступный инструментальный сервис коллективной разработки СОЗ на основе базовой технологии. Он позволяет распределять задания и отслеживать их выполнение коллективом разработчиков. В качестве расширений доступны также созданные пользователями специализированные оболочки для решения различных классов задач. В настоящее время на платформе созданы следующие порталы знаний: Медицина; Математика; Автономные необитаемые подводные аппараты; Знания по экспериментальной диагностике сельскохозяйственных культур; Защита информации; Педагогическая психология; Технология программирования.

Расширение инструментария разработчиками и пользователями платформы активно продолжается, что позволяет на его основе создавать жизнеспособные СОЗ. В заключение также важно отметить, что жизнеспособность ПС обеспечивается не только расширяемостью инструментария. Ключевую роль играет архитектура и способы представления всех компонентов СОЗ. Агентный подход, формирование компонентов по их декларативным структурным моделям с использованием редакторов, средства автоматизации проектирования, реализованные на облачной платформе IACPaaS и описанные ранее в работах [19, 20], также направлены на достижение этой важной характеристики СОЗ.

Работа выполнена при частичной финансовой поддержке РФФИ (гранты 18-07-01079, 16-07-00340) и программы "Дальний Восток" (проект 18-МНТ-009).

Список литературы

1. Додонов А. Г., Ландэ Д. В. Живучесть информационных систем. Киев: Наукова думка, 2011. 256 с.
2. Кряжич О. А. Обеспечение жизнеспособности информации во времени при ее обработке в СППР // Математические машины и системы. 2015. № 2. С. 170–176.
3. Черников Б. В. Управление качеством программного обеспечения. М.: ФОРУМ-Инфра-М, 2012. 240 с.
4. Izurieta C., Bieman J. M. A multiple case study of design pattern decay, grime, and rot in evolving software systems // Software Quality Journal. 2013. Vol. 21, N. 2. P. 289–323.
5. Черняк Л. Жизнеспособные программы // Открытые системы. СУБД. 2004. № 9. URL: <https://www.osp.ru/os/2004/09/184588/> (дата обращения 25.05.2018).
6. Rowe D., Leaney J. Evaluating evolvability of computer based systems architectures—an ontological approach // Engineering of Computer-Based Systems. Proc. International Conference and Workshop on. IEEE, 1997. P. 360–367.
7. Breivold H. P., Crnkovic I., Eriksson P. J. Analyzing software evolvability // Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE. 2008. P. 327–330.
8. Islam M., Katiyar V. Development of a software maintenance cost estimation model: 4th GL perspective // International Journal of Technical Research and Applications. 2014. Vol. 2, Issue 6. P. 65–68.
9. Dehaghani S. M. H., Hajrahimi N. Which factors affect software projects maintenance cost more? // Acta Informatica Medica. 2013. Vol. 21. N. 1. P. 63–66.

-
10. Рыбина Г. В. Интеллектуальные системы: от А до Я. Серия монографий в трех книгах. Кн. 3. Проблемно-специализированные интеллектуальные системы. Инструментальные средства построения интеллектуальных систем. М.: Научтехлитиздат, 2015. 180 с.
11. Musen M. A. The Protégé Project: A Look Back and a Look Forward // AI Matters. 2015. Vol. 1, Issue 4. P. 4–12. DOI: 10.1145/2757001.2757003.
12. Pressman R. S. Software engineering: a practitioner's approach. McGraw-Hill, 2010. 930 p.
13. Boehm B. Software architectures: critical success factors and fast frivers // IEEE transactions on Software Engineering, 2007. P. 965–971.
14. Mortensen J., Minty E., Januszyk M., Sweeney T., Rector A., Noy N., Musen M. A. Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT // Journal of the American Medical Informatics Association. 2014. Vol. 22, N. 3. P. 640–648. DOI: 10.1136/amiajnl-2014-002901.
15. Ghazvinian A., Noy N. F., Musen M. A. Creating mappings for ontologies in biomedicine: simple methods work // AMIA Annual Symposium Proceedings. American Medical Informatics Association. 2009. Vol. 2009. P. 198–202.
16. Storey M.-A., Musen M. A., Silva J., Best C., Ernst N., Fergerson R., Noy N. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé // Workshop on interactive tools for knowledge capture. 2001. Vol. 73. P. 35–43.
17. Рыбина Г.В. Интеллектуальная технология построения обучающих интегрированных экспертизных систем: новые возможности // Открытое образование. 2017. Т. 21, № 4. С. 43–57.
18. Gensym G2. The World's Leading Software Platform for Real-Time Expert System Application. URL: <http://www.gensym.com/wp-content/uploads/Gensym-l-G2.pdf> (дата обращения 25.05.2018).
19. Грибова В. В., Клещев А. С., Москаленко Ф. М., Тимченко В. А., Федорищев Л. А., Шалфеева Е. А. Облачная платформа IACPaaS: текущее состояние и перспективы развития // Информационные и математические технологии в науке и управлении. 2016. № 2. С. 94–102.
20. Gribova V., Kleschev A., Moskalenko P., Timchenko V., Fedorischev L., Shalfeeva E. The IACPaaS cloud platform: Features and perspectives // Computer Technology and Applications (RPC), 2017 Second Russia and Pacific Conference on. IEEE, 2017. P. 80–84.
21. Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A. A Two-level Model of Information Units with Complex Structure that Correspond to the Questioning Metaphor // Automatic Documentation and Mathematical Linguistics. 2015. Vol. 49, N. 5. P. 172–181.
22. Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A. A Model for Generation of Directed Graphs of Information by the Directed Graph of Metainformation for a Two-Level Model of Information Units with a Complex Structure // Automatic Documentation and Mathematical Linguistics. 2015. Vol. 49, N. 6. P. 221–231.
23. Грибова В. В., Клещев А. С., Москаленко Ф. М., Тимченко В. А., Федорищев Л. А., Шалфеева Е. А. Управляемая графовыми грамматиками разработка оболочки интеллектуальных сервисов на облачной платформе IACPaaS // Программная инженерия. 2017. Т. 8, № 10. С. 435–447.
24. Gribova V. V., Fedorishchev L. A. Software Tool for Virtual Environment Development Based on Cloud Platform // Journal of Computer and Systems Sciences International. 2017. Vol. 56, N. 6. P. 937–944. DOI: 10.1134/S1064230717060065.
-

Extensible Toolkit for the Development of Viable Systems with Knowledge Bases

V. V. Gribova, gribova@iacp.dvo.ru, A. S. Kleshchev, kleshchev@iacp.dvo.ru,
F. M. Moskalenko, philipmm@iacp.dvo.ru, V. A. Timchenko, vadim@dvo.ru,
E. A. Shalfeeva, shalf@iacp.dvo.ru, Institute of Automation and Control Processes,
Far Eastern Branch of the Russian Academy of Sciences (IACP FEB RAS), Vladivostok, 690041,
Russian Federation

Corresponding author:

Gribova Valeriya V., Research Deputy Director, IACP FEB RAS, Vladivostok, 690041, Russian Federation
E-mail: gribova@iacp.dvo.ru

Received on June 5, 2018

Accepted on June 20, 2018

The paper presents the concept and architecture of an extensible toolkit for the development and maintenance of Knowledge Based Systems (KBSs). Such extensibility is a necessary condition for KBS viability in the course of changes of domain, user requirements and operating environment. The importance of provision of the toolkit extensibility by both its developers and users is justified. The key requirements for such a toolkit are identified: the possibility of its extensibility; the existence of a unified environment and instrumental support for the development and improvement of both KBSs and the toolkit itself. A three-level architecture of the extensible toolkit is proposed. It contains such levels as: an instrumental kernel, a basic and an extensible set of tools. The instrumental kernel defines the basic principle for the development of applied and instrumental services (systems): their software and information components are formed on the basis of a two-level approach. The instrumental kernel is sufficient to create information components by their structural models and to manage them and to specify declarative models of software components of all types. The basic toolkit includes a set of models and software sufficient to create and manage a KBS with the use of the basic technology. An extensible toolkit can include the following: new technologies

for KBS development and their instrumental support (including tools for collective development and management), KBS shells, libraries of reusable components. The described concept is implemented within the toolkit of the IACPaaS cloud platform. It includes evolution mechanisms based on a unified declarative representation of information and software components with automatic generation of editors for their formation.

Keywords: knowledge-based system (KBS), KBS development technology, software system maintenance, instrumental support for KBS development, development tools, model description language, model editor, problem solver editor, editor generator

Acknowledgements: This work was partially supported by the Russian Foundation for Basic Research, project nos. 18-07-01079 and 16-07-00340 and by PFI "Far East" project no 18-MNT-009.

For citation:

Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A., Shalfeeva E. A. Extensible Toolkit for the Development of Viable Systems with Knowledge Bases, *Programmnaya Ingeneria*, 2018, vol. 9, no. 8, pp. 339–348.

DOI: 10.17587/prin.9.339-348

References

1. Dodonov A. G., Landeh D. V. *Zhivuchest' informacionnyh sistem* (Survivability of information systems), Kiev, Naukova Dumka, 2011, 256 p. (in Russian).
2. Kryazhich O. A. Obespechenie zhiznesposobnosti informacii vo vremeni pri ee obrabotke v SPPR (Ensuring the viability of the information during its processing in decision support systems), *Matematicheskie mashiny i sistemy*, 2015, no. 2, pp. 170–176 (in Russian).
3. Chernikov B. V. *Upravlenie kachestvom programmnogo obespecheniya* (Software quality management), Moscow, FORUM-INFRA-M, 2012, 240 p. (in Russian).
4. Izurieta C., Bieman J. M. A multiple case study of design pattern decay, grime, and rot in evolving software systems, *Software Quality Journal*, 2013, vol. 21, no. 2, pp. 289–323.
5. Chernyak L. Zhiznesposobnye programmy (A viable program), *Otkrytie sistemy. SUBD*, 2004, no. 9, available of: <https://www.osp.ru/os/2004/09/184588/> (date of access 25.05.2018) (in Russian).
6. Rowe D., Leaney J. Evaluating evolvability of computer based systems architectures—an ontological approach, *Engineering of Computer-Based Systems. Proc. International Conference and Workshop on IEEE*, 1997, pp. 360–367.
7. Breivold H. P., Crnkovic I., Eriksson P. J. Analyzing software evolvability, *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE*, 2008, pp. 327–330.
8. Islam M., Katiyar V. Development of a software maintenance cost estimation model: 4th GL perspective, *International Journal of Technical Research and Applications*, 2014, vol. 2, no. 6, pp. 65–68.
9. Dehaghani S. M. H., Hajrahimi N. Which factors affect software projects maintenance cost more? *Acta Informatica Medica*, 2013, vol. 21, no. 1, pp. 63–66.
10. Rybina G. V. *Intellektual'nye sistemy: ot A do YA. Seriya monografij v trekh knigah. Kn. 3. Problemino-specializirovannye intellektual'nye sistemy. Instrumental'nye sredstva postroeniya intellektual'nyh system* (Intelligent systems: A to Z. A series of monographs in three books. Book 3. Problem-specialized intelligent systems. Tools for building intelligent systems), Moscow, Nauchtekhnitizdat, 2015, 180 p. (in Russian).
11. Musen M. A. The Protégé Project: A Look Back and a Look Forward, *AI Matters*, 2015, vol. 1, issue 4, pp. 4–12, doi: 10.1145/2757001.2757003.
12. Pressman R. S. *Software engineering: a practitioner's approach*, McGraw-Hill, 2010. 930 p.
13. Boehm B. Software architectures: critical success factors and cost drivers, *IEEE transactions on Software Engineering*, 2007, pp. 965–971.
14. Mortensen J., Minty E., Januszyk M., Sweeney T., Rektor A., Noy N., Musen M. A. Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT, *Journal of the American Medical Informatics Association*, 2014, vol. 22, no. 3, pp. 640–648, doi: 10.1136/amiajnl-2014-002901.
15. Ghazvinian A., Noy N. F., Musen M. A. Creating mappings for ontologies in biomedicine: simple methods work, *AMIA Annual Symposium Proceedings. American Medical Informatics Association*, 2009, vol. 2009, pp. 198–202.
16. Storey M.-A., Musen M. A., Silva J., Best C., Ernst N., Ferguson R., Noy N. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé, *Workshop on interactive tools for knowledge capture*, 2001, vol. 73, pp. 35–43.
17. Rybina G. V. Intellektual'naya tekhnologiya postroeniya obuchayushchih integrirovannyh ekspertnyh sistem: novye vozmozhnosti (Intelligent technology for construction of tutoring integrated expert systems: new aspects), *Open Education*, 2017, vol. 21, no. 4, pp. 43–57 (in Russian).
18. Gensym G2. The World's Leading Software Platform for Real-Time Expert System Application, available at: <http://www.gensym.com/wp-content/uploads/Gensym-l-G2.pdf> (date of access 25.05.2018).
19. Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A., Fedorishchev L. A., Shalfeeva E. A. Oblachnaya platforma IACPaaS: tekushchee sostoyanie i perspektivy razvitiya (IACPaaS cloud platform: Current state and evolution trends), *Information and mathematical technologies in science and management*, 2016, no. 2, pp. 94–102 (in Russian).
20. Gribova V., Kleschev A., Moskalenko P., Timchenko V., Fedorischev L., Shalfeeva E. The IACPaaS cloud platform: Features and perspectives, *Computer Technology and Applications (RPC), 2017 Second Russia and Pacific Conference on. IEEE*, 2017, pp. 80–84.
21. Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A. A Two-level Model of Information Units with Complex Structure that Correspond to the Questioning Metaphor, *Automatic Documentation and Mathematical Linguistics*. 2015, vol. 49, no. 5, pp. 172–181.
22. Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A. A Model for Generation of Directed Graphs of Information by the Directed Graph of Metainformation for a Two-Level Model of Information Units with a Complex Structure, *Automatic Documentation and Mathematical Linguistics*, 2015, vol. 49, no. 6, pp. 221–231.
23. Gribova V. V., Kleschev A. S., Moskalenko Ph. M., Timchenko V. A., Fedorischev L. A., Shalfeeva E. A. Upravlyayemaya grafovymi grammatikami razrabotka obolochek intellektual'nyh servisov na oblachnoj platforme IACPaaS (A Graph Grammar Managed Development of Intelligent Service Shells on the IACPaaS Cloud Platform), *Programmnaya Ingeneria*, 2017, vol. 8, no. 10, pp. 435–447 (in Russian).
24. Gribova V. V., Fedorishchev L. A. Software Tool for Virtual Environment Development Based on Cloud Platform, *Journal of Computer and Systems Sciences International*, 2017, vol. 56, no. 6, pp. 937–944, doi: 10.1134/S1064230717060065.

С. В. Подковальников, канд. техн. наук, зав. лабораторией, e-mail: spodkovalnikov@isem.irk.ru,
И. Л. Трофимов, ст. инженер, e-mail: t_john88@isem.irk.ru, **Л. Н. Трофимов**, вед. программист,
e-mail: trofi@isem.irk.ru, Институт систем энергетики им. Л. А. Мелентьева Сибирского отделения
Российской академии наук, г. Иркутск

Технологические аспекты геоинформационной вычислительной системы для поддержки исследований в области формирования межгосударственных энергообъединений

Рассмотрена геоинформационная вычислительная система, предназначенная для прогнозирования развития межгосударственных электроэнергетических объединений с использованием оптимизационной модели ОРИРЭС. Предложены оригинальная технология хранения информации в интегрированной объектно-ориентированной базе данных, а также технология обработки и представления данных в интерфейсе системы. Проведен ряд вычислительных экспериментов, в ходе которых получены удобные для анализа агрегированные формы представления конечных результатов решения оптимизационной модели.

Ключевые слова: геоинформационная вычислительная система, оптимизационная модель, обработка данных, объектная база данных, электростанция, электроэнергетическая система, межгосударственное электроэнергетическое объединение

Введение

Формирование межгосударственных электроэнергетических объединений (МГЭО) — глобальная тенденция развития мировой электроэнергетики. Факторы, способствующие развитию МГЭО, обусловлены расширением внешнеэкономических связей между странами, дефицитом мировых природных энергетических ресурсов, несовпадением их центров добычи и потребления, неравномерностью сезонных и суточных электрических нагрузок. В результате формирования МГЭО повышается надежность электроэнергетических систем, появляется возможность широкомасштабного вовлечения в электроэнергетические балансы возобновляемых источников энергии, образуются значительные системные эффекты и др. Формирование МГЭО требует обоснования его технико-экономической эффективности, прогноза дальнейшего развития, удовлетворения экономических интересов и соблюдения технических стандартов стран-участников и учета многих других аспектов [1].

Для проведения подобного рода исследований требуется большая предварительная работа по сбору и анализу огромного количества информации — энергетической, технико-экономической, экологической и пр., сложные технические и экономические расчеты, значительные интеллектуальные и

информационные ресурсы. Без наличия развитых информационных технологий, отражающих специфику данной предметной области, исследования занимают огромное количество времени и средств. Многоаспектные результаты расчетов также трудно воспринимаемы специалистами без специализированного интерфейса для обработки и представления данных.

Исследованиями в области формирования МГЭО занимаются многие зарубежные ученые и международные организации. К их числу относятся: Global Energy Interconnection Development and Cooperation Organization (GEIDCO, Китай), Mongolia Energy Policy Regulatory Department (Монголия), Asia Pacific Energy Research Centre (международная организация APERC), United Nations Economic and Social Commission for Asia and the Pacific (комиссия ООН по социально-экономическому развитию Азиатско-Тихоокеанского региона — ЭСКАТО ООН). Сюда же можно отнести различные исследовательские институты, например, Renewable Energy Institute (Япония), Korea Electrotechnology Research Institute (Южная Корея), China Electric Power Planning & Engineering Institute (Китай), Global Energy Interconnection Research Institute (Китай) и ряд других организаций аналогичной направленности [2—5].

К известным геоинформационным системам (ГИС), которые используются в России и за рубе-

жом для изучения вопросов развития энергетики, относятся ESRI ArcGIS, MapInfo, GIS Energy. Наиболее широко известные ГИС, такие как Google maps, Wikimapia, OpenStreets, Yandex-карты не предназначены для работы с большим количеством разнородных данных по энергетике. Все указанные выше системы созданы в основном для решения универсальных информационно-аналитических задач, в которых отсутствует вычислительная часть, необходимая для проведения соответствующих исследований.

Как известно, в основе большинства отечественного и зарубежного программного обеспечения, предназначенного для работы с большими массивами информации, лежат стандартные методы обработки данных, такие как OLAP, Data Mining, а также средства для их хранения, к числу которых в основном относятся реляционные базы данных. Такие средства хранения и обработки данных предназначены для универсальных решений, но не позволяют в полной мере реализовать поставленные перед авторами задачи. В Институте систем энергетики им. Л. А. Мелентьева (ИСЭМ СО РАН) решаются различные задачи, ориентированные на конкретные расчеты для исследования перспектив и эффективности формирующихся и развивающихся межгосударственных энергообъединений. При этом МГЭО и входящие в них электроэнергетические системы (ЭЭС) моделируются как сложные по структуре объекты, которые описываются большим набором различных динамически меняющихся параметров. При отсутствии специализированного программного обеспечения подготовка исходных данных и обработка полученных результатов при решении отмеченных выше задач долгое время выполнялись путем трудоемких расчетов, проводимых практически вручную с использованием программы Microsoft Excel.

Поэтому появилась необходимость создания и использования специализированного программного обеспечения для проведения комплексных научных исследований в области формирования и прогнозирования развития МГЭО.

Для решения поставленной цели необходимо было, прежде всего, разработать технологию (способ) сбора и обработки слабоструктурированных данных, к которым относятся разрозненная информация, собранная из различных источников, алгоритмы ее преобразования и хранения. Также было необходимо реализовать интеграцию собранных информационных ресурсов в геоинформационной вычислительной системе с объектно-ориентированной базой данных.

Ближайшие по функциональным возможностям аналоги данной системы, известные авторам, — это энергетический портал "Asia Pacific Energy Portal", разработанный при участии ЭСКАТО ООН [6], и Китайская ГИС, разработанная при участии GEIDCO [7], содержащие большое количество энергетических данных по электростанциям и линиям электропередачи по всему миру. Перечисленные аналоги не имеют вычислительной части и в контексте настоящей статьи могут называться лишь геоинфор-

мационными системами. Разрабатываемая авторами геоинформационная вычислительная система помимо визуализации энергетических данных предназначается для проведения экспертами оптимизационных расчетов и построения обоснованных прогнозов развития МГЭО с использованием математической оптимизационной модели.

1. Геоинформационная вычислительная система

Для повышения эффективности и качества научных исследований в области формирования межгосударственных электроэнергетических объединений, которые проводятся в ИСЭМ СО РАН, авторами разрабатывается специализированная геоинформационная вычислительная система (ГИВС). Наряду с решением информационно-аналитических задач такая система позволяет исследователям проводить различного рода вычисления, строить прогнозы развития МГЭО с использованием математической оптимизационной модели.

Задачи, которые решают с использованием ГИВС, весьма разнородны. Условно их можно разделить на перечисленные далее три группы.

I. Информационно-аналитические задачи, к которым относятся:

- хранение в единой структуре энергетической информации, собранной из различных источников;
- анализ данных за различные временные периоды в целях выявления тенденций развития электроэнергетики, в частности развития определенных типов энергоисточников, технологий генерации электроэнергии и использования различных видов топлива;
- графическое и картографическое отображение данных.

II. Расчетно-прогностические (вычислительные) задачи, для решения которых используется разработанная в ИСЭМ СО РАН модель Оптимизации Развития и Режимов Электроэнергетических Систем (ОРИРЭС). Целью их решения являются:

- оценка эффективности развития межгосударственных электрических связей;
- оценка уровня интеграции и эффективности МГЭО;
- оптимизация структуры генерирующих мощностей, режимов электроэнергетических систем и МГЭО;
- исследования развития ЭЭС различных стран.

III. Научно-популяризационные задачи, которые направлены на привлечение интереса научного сообщества к результатам исследований. На базе ГИВС разработан внешний интернет-сервис (веб-интерфейс ГИВС). Через этот интерфейс решают следующие задачи:

- продвижение идеи электроэнергетической интеграции среди специалистов российских и иностранных электроэнергетических организаций, а также среди всех заинтересованных физических и юридических лиц;



Рис. 1. Функциональные блоки ГИВС

• популярное представление результатов проводимых научных исследований в сети Интернет.

Основные структурно-функциональные блоки ГИВС и схема потока данных между ними представлены на рис. 1.

Блок "Источники данных" выполняет функцию сбора данных с разнородных информационных ресурсов и их преобразования в единую структуру базы данных, используемую в ГИВС. В нем реализованы программные алгоритмы обработки форм статистической отчетности с использованием семантической (смысловой) обработки информации по заданным шаблонам (параметрам, соответствующим структуре данных). Для работы с данными, полученными (скачанными) из энергетических интернет-ресурсов, таких как Global Energy, International Energy Agency, Energy Pedia Portal [8–10] и др., авторами написаны специальные алгоритмы верификации (различные проверки на целостность и дублетность–повторяемость) и преобразования данных (реструктуризация, индексация объектов для базы данных).

Для хранения информации в единой структуре в рамках ГИВС авторами разработана "Объектно-ориентированная база данных". Данные представляют собой систему или набор файлов специального формата, содержащего параметры объекта базы данных, его текстовые и числовые значения со специальными разделителями. Разработанная авторами объектно-ориентированная база данных в ГИВС позволяет собирать, хранить и обрабатывать множество разнородной информации, взятой из различных несвязанных источников, но преобразованной в единообразный вид — структуру, удобную для машинной обработки.

Еще один функциональный блок ГИВС предназначен для работы с оптимизационной моделью ОРИРЭС. Специальный интерфейс ГИВС позволяет экспертам настраивать и изменять параметры модели ОРИРЭС, число узлов и ограничений, запускать "оптимизатор" и генерировать конечный набор таблиц с результатами оптимального решения. Далее полученный набор таблиц, определяющий конкретный

сценарий развития МГЭО для расчетного года, также сохраняется в структуре данных ГИВС. Это дает возможность исследователю получать новые решения оптимизационной модели и сравнивать различные сценарии развития МГЭО.

Для визуального представления и анализа информации из объектно-ориентированной базы данных в ГИВС разработан блок "Интерфейс представления данных". Данный функциональный блок ГИВС предназначен для решения информационно-аналитических задач. В него входят разработанный авторами конструктор запросов к объектно-ориентированной базе данных в терминах предметной области; программный модуль — интерфейс для работы с картами; программный модуль для построения специальных графиков и диаграмм. В качестве подложки (фона) используются карты, построенные с помощью свободно распространяемого API (Application Program Interface) Google Maps, на которые накладываются полупрозрачные слои, отображающие различные энергетические показатели. Интерфейс позволяет масштабировать карту и просматривать тренды изменения показателей за различные временные периоды, например, в ретроспективе за последние 20 лет. Данные тренды можно просматривать на построенных картах, а также в виде графиков или диаграмм.

Авторами был также разработан и запущен в сети Интернет специальный блок ГИВС "Аналитический интернет-сервис" для представления результатов исследований. С использованием веб-интерфейса в этот блок по запросу пользователя экспортируются сформированные в ГИВС карты, графики, таблицы и диаграммы, с сохранением некоторых динамических функций, реализованных средствами веб-программирования. Например, пользователи веб-интерфейса ГИВС могут просматривать динамику изменения энергетических показателей на картах за различные годы.

Работа пользователей в ГИВС состоит из нескольких этапов. На предварительном этапе исследователи занимаются непосредственным поиском актуальной, но чаще всего не систематизированной информации в различных источниках и интернет-ресурсах. Данный этап является наиболее трудоемким с точки зрения использования человеческих ресурсов. Далее найденная информация анализируется в ГИВС с помощью специальных программных алгоритмов на предмет соответствия семантическим шаблонам (наборам объектов определенного типа и их параметров) в структуре базы данных, при необходимости шаблоны редактируются либо создаются новые (под новый тип объектов). При этом общий вид/структура объектов базы данных не меняется, могут изменяться только набор объектов, их параметров, значений этих параметров и отношений между объектами. Объектами в базе данных ГИВС могут быть представлены электростанции, линии электропередачи, электроэнергетические системы и их объединения, а также страны.

Затем происходит автоматическая загрузка информации в объектную структуру базы данных. При

каждом новом типе объектов данная операция проводится лишь один раз. В случае, когда приходит обновленная информация по существующим в базе данных объектам, например, новая статистика за текущий год, то в соответствии с уже разработанным шаблоном специальная процедура автоматически распределяет новые значения по соответствующим ячейкам в структуре объектов базы данных.

На следующем этапе может работать уже другой пользователь ГИВС — исследователь-эксперт, занимающийся составлением расчетных схем и прогнозов развития МГЭО с использованием математической модели ОРИРЭС. Для этого в ГИВС запускается специальный интерфейс, в котором эксперт может настраивать различные параметры модели, задавать расчетную схему с набором узлов и запускать встроенный оптимизатор. Многие параметры, отображаемые в данном интерфейсе, берутся из соответствующих объектов базы данных, например, установленные мощности электростанций или стоимость определенного вида топлива в некоторой стране и т. д.

На выходе из оптимизатора в ГИВС формируется набор таблиц с оптимальными значениями параметров МГЭО для расчетного года (экономических, технических и пр.), полученных в результате решения оптимизационной задачи модели ОРИРЭС.

В ГИВС предусмотрено использование полученных решений в табличном виде, например, экспорт в Microsoft Excel для дальнейшей работы с таблицами, либо представление отдельных выбранных показателей в графическом интерфейсе ГИВС для построения интерактивных карт, графиков и диаграмм в целях проведения ретроспективного и перспективного анализа данных.

Возможен также экспорт обработанной информации во внешний аналитический интернет-сервис в целях использования результатов исследований научным сообществом в сети Интернет.

При реализации ГИВС были использованы перечисленные далее технологии.

В качестве основы построения ГИС-системы и объектно-ориентированной базы данных были использованы среда разработки Embarcadero Delphi XE, API Google Maps, SHP- и XML-файлы, содержащие географическую и другую информацию об объектах изучения. Для создания веб-интерфейса ГИВС были применены современные веб-технологии, основанные на PHP, Ajax, Jquery, JavaScript, MySQL, Bootstrap. Оптимизационная модель ОРИРЭС представлена в виде набора математических формул и уравнений, решаемых в ГИВС посредством общей алгебраической системы моделирования GAMS. Параметры модели задаются и записываются в/из объектно-ориентированной базы данных для их последующей обработки и представления в таблично-графическом интерфейсе ГИВС.

Применяемые в разработке ГИВС информационные технологии позволяют в полной мере реализовывать поставленные задачи. Разрабатываемая ГИВС

интегрирует в себе алгоритмы сбора/загрузки и обработки слабоструктурированных данных (на этапе их загрузки и реструктурирования в объектный вид), а также технологии графического и картографического представления данных. Для представления конечной информации в картографическом интерфейсе используются совмещенные слои ГИС-карт с основными составляющими электроэнергетической инфраструктуры стран и электроэнергетических объединений.

2. Технология хранения и представления информации в ГИВС

Как уже отмечалось выше, для хранения и обработки информации в ГИВС используется разработанная авторами объектно-ориентированная база данных (ООБД). Суть технологии хранения данных в ООБД заключается в том, что вся информация, поступающая/записываемая в систему, структурируется с помощью специальных программных алгоритмов и хранится в виде уникальных объектов базы данных, описывающих энергетические параметры объектов реального мира. Виды объектов ООБД могут быть различными и независимыми — электростанция, линия электропередачи, регион, страна, электроэнергетическое объединение и пр. [11].

Каждый объект ООБД представляет собой запись, содержащую уникальный идентификатор объекта, набор описывающих его параметров и значе-

ния каждого параметра в текстовом или числовом виде, хранящиеся по годам, месяцам, суткам и т. д. (рис. 2).

На рис. 2 представлен список объектов ООБД типа "страна". На рис. 2 также продемонстрирован отдельный файл, открытый с помощью простого текстового редактора, в котором можно увидеть фрагмент параметров и значений выбранного объекта, записанных в специальном формате. В данном примере показан фрагмент данных объекта "Russia", его текстовые параметры: идентификатор (ID), год начала собираемых данных (year_start), название (name_rus), принадлежность к определенному региону (ch_APEC) и числовые параметры: количество экспортированного угля за год (exp_coal), экспорт нефти (exp_oil), экспорт газа (exp_gas), экспорт электроэнергии (exp_el), показатели по импорту и пр. Все значения даны в килотоннах нефтяного эквивалента, они взяты из таблиц энергетического баланса стран мира. Данный список параметров неограничен по длине и может занимать несколько страниц.

Внутренняя структура файлов ООБД не только удобна для машинной обработки, но и понятна человеку. Универсальность и простота структуры данных делает их практически не зависимыми от самой программной части среды/системы и процедур обработки. Не составляет большого труда разобраться в данной структуре и написать процедуры обработки на другом языке программирования, например, при создании веб-интерфейса для ГИВС.

ID#Russia
year_start#1980
name_rus#Россия
ch_APEC#APEC
imp_coal# 35087 28011 23719 17143 16684 14012 12215 12553
imp_oil# 18592 18874 10748 10351 4852 8546 6771 4024 5603
imp_gas# 56422 11123 5642 5320 1612 2907 2794 3618 2426 33
imp_el# 3013 3018 2383 2123 2034 1580 1063 615 710 719 756
exp_coal# 40423 25525 27867 17763 16584 18774 17464 16651
exp_oil# 222907 175271 143161 128193 127358 122947 127081
exp_gas# 201662 154529 157727 138102 148924 153975 158650
exp_el# 3728 4055 3780 3734 3797 3266 2739 2308 2260 1937

Рис. 2. Форма записи объектов в ООБД

Технология хранения информации в виде объектов имеет преимущество перед реляционными базами данных по следующим причинам.

- Количество и длина записей в предложенной авторами структуре данных не ограничены. Соответственно, в одной строке для определенного параметра объекта можно хранить бесконечное количество его выходных значений, например, почасовую нагрузку в течение целого года или периода лет, без необходимости создания дополнительных реляционных таблиц.

- Может быть использован свободный тип/формат значений параметров, а именно дробный, целочисленный, или текстовый, хранящийся в неограниченном диапазоне значений. Например, частое изменение названий энергетических объектов по годам может быть представлено в виде параметра "название", текстовые значения которого записаны в одной строке со специальными разделителями. Значения такого параметра могут быть взяты из ООБД за любой требуемый год, или за несколько лет, например, чтобы увидеть тенденцию изменения названий.

- Отсутствие большого числа реляционных таблиц и индексов в случае работы с большими списками энергетических объектов разного вида или несвязанных между собой типов объектов.

- Необходимость компактного представления исчерпывающей (полной имеющейся) информации об энергетическом объекте в одном файле специального формата. Такое представление дает уверенность в целостности данных и исключает потенциальные потери, ошибки данных, возникающие при сложных SQL-запросах, "собирающих" значения одного или нескольких объектов из различных реляционных таблиц.

Авторами разработаны специальные программные алгоритмы и пользовательский интерфейс для ввода/вывода информации из ООБД с использованием графического и картографического представления данных. Разработаны программные процедуры верификации данных и проверки на целостность. Для исключения дублирования объектов, собранных из различных источников, авторами разработаны алгоритмы сравнения, в которых используются простые методы семантического и синтаксического разбора текста. Существует также возможность визуального поиска объектов (в случае электростанций) по географическим координатам на картах Google Maps — авторами были выявлены случаи, когда географические координаты электростанции намеренно или случайно были "сдвинуты" на несколько километров от их реального места расположения. Большинство крупных объектов-электростанций в ООБД имеют реальные географические координаты и могут быть представлены на картах в картографическом интерфейсе ГИВС.

Для проведения качественных исследований необходимо иметь максимально полную информацию об изучаемых объектах или процессах. В частно-

сти, для определения современного состояния некоторой электроэнергетической системы в регионе необходимо иметь информацию об установленных электрогенерирующих мощностях данной ЭЭС. При поиске и получении такой информации исследователь сталкивается с субъективными и объективными трудностями, поскольку источники информации, как правило, разрознены и находятся на различных интернет-ресурсах. Для преодоления этой трудности авторами был разработан полуавтоматизированный способ сбора/записи слабоструктурированной информации из различных источников и размещения ее в объектную структуру с последующей верификацией и анализом. Конкретный алгоритм загрузки и преобразования информации в ООБД зависит от определенных видов источников данных. Авторы не берут в рассмотрение случайные источники с информацией, хранящейся в виде текста из каких-либо статей или новостных лент, хотя для этого в ГИВС предусмотрена возможность "ручного" добавления/редактирования данных. Программно-реализованные алгоритмы (далее — алгоритмы) сбора данных написаны под конечный набор источников, которые являются наиболее крупными интернет-ресурсами, базами данных и хранилищами энергетической информации. Наиболее крупными и популярными источниками энергетической информации, востребованными исследователями во всем мире, являются Международное Энергетическое Агентство (International Energy Agency — IEA), British Petroleum (BP), Росстат, Евростат, Stats of America, интернет-ресурс Global Energy Organization и др. Общая схема загрузки данных в ООБД представлена на рис. 3.

Под каждый источник информации исследователь выбирает определенный шаблон данных из структуры ООБД — указывает определенный тип/вид объекта и его параметры (это могут быть электростанции в определенной стране, страны, линии электропередачи, ЭЭС и т. д.). Затем запускается специальная программа, реализующая алгоритм, соответствующий выбранному шаблону, в котором выполняются процедуры сравнения объектов (для исключения дублетности), их названий, параметров и последующие процедуры записи данных в структуру ООБД. Система ГИВС постоянно развивается, с появлением новых источников информации разрабатываются новые шаблоны, содержащие советующие типы объектов из ООБД.

Например, для загрузки в ООБД большого объема информации, представленной в виде онтологий на открытом ресурсе DBpedia/EniPedia.org, авторами был разработан специальный алгоритм, реализующий синтаксический анализ XML-формата данных, экспортированных из указанного интернет-ресурса, и распределяющий параметры объектов-электростанций и их значения в структуру ООБД [12]. Для определенного набора стран, необходимых для исследований (прежде всего это страны Азиатско-Тихоокеанского региона), из указанных источников в ООБД были записаны несколько тысяч объектов-

Выбор шаблона из структуры ООБД
и запуск программного алгоритма скачивания,
соответствующего определенному источнику

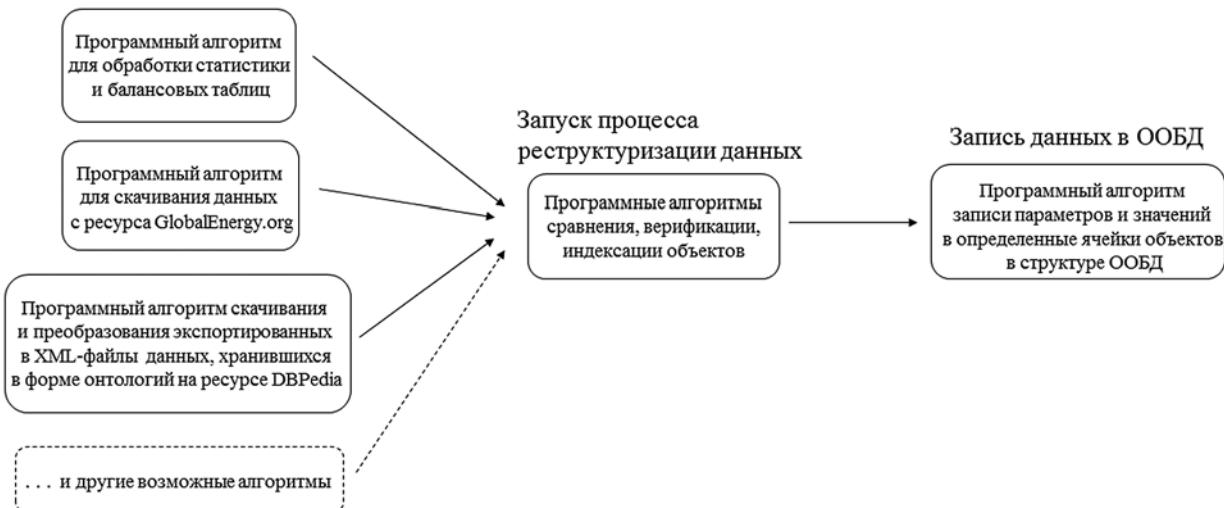


Рис. 3. Схема процесса загрузки и преобразования данных в ООБД

электростанций с их реальными географическими координатами и технико-экономическими показателями.

Для работы со статистической информацией из энергетических балансов, заданных в специальном международном формате, авторами разработан другой алгоритм, предназначенный для считывания таблиц Microsoft Excel (в которые сводятся полученные энергетические балансы), и автоматического установления соответствия столбцов и строк балансовых таблиц параметрам и соответствующим значениям объектов в структуре ООБД.

Таким образом, начало последовательности действий пользователя ГИВС по сбору и загрузке данных в ООБД зависит от конкретного типа источника информации. Однако, учитывая тот факт, что необходимая для исследований информация регулярно обновляется и требует актуализации в ООБД, разработанные однажды алгоритмы запускаются неоднократно, и пользователю необходимо лишь выбрать нужный из них, советующий указанному источнику. Для верификации объектов, которые загружены из различных внешних источников в ГИВС, используются алгоритмы сопоставления полученных новых объектов с уже существующими в ООБД.

В результате использования программ, реализующих разработанные авторами алгоритмы загрузки данных из различных источников, пользователями ГИВС формируется и пополняется тематическая ООБД. Она содержит уникальную, собранную в единой структуре информацию, использование которой многократно расширяет возможности и повышает качество проведения научных исследований.

Для графического интерфейса ГИВС разработаны специальные алгоритмы выгрузки данных, позволя-

ющие работать с информацией из ООБД, анализировать ее, редактировать и проводить различного рода вычислительные расчеты. Результаты таких расчетов также могут быть загружены в ООБД для их последующего использования.

Дальнейшая обработка данных из ООБД проводится с помощью специального языка запросов (разработанная авторами надстройка над SQL, использующая термины предметной области для удобства работы пользователей-экспертов). Такой язык позволяет извлекать параметры объектов и осуществлять их сортировку, разного рода выборки и агрегирование. Объекты обладают расширяемым набором параметров, которые могут добавляться в процессе работы с ООБД. В частности, для электростанций были добавлены параметры их картографической привязки и возможность представления на картах в ГИС [13, 14].

Имея информацию о точном месте расположения объектов, можно агрегировать данные на различных уровнях, что открывает возможности для масштабирования модели ОРИРЭС — формирования разных уровней детализации расчетной схемы. Собранная информация может быть агрегирована по различным странам, регионам, национальным энергосистемам или их подсистемам. На рис. 4 показано агрегирование установленных мощностей электростанций объединенных электроэнергетических систем (ОЭС) Северного и Северо-Восточного Китая, полученное в картографическом интерфейсе ГИВС.

Первичный список объектов-электростанций в ООБД был сформирован из открытого источника в сети Интернет "EniPedia.org". Электроэнергетические системы с их агрегированными мощностями, показанными на рис. 4, также, как и отдельные электростанции, хранятся в ООБД в виде объектов и



Рис. 4. Агрегирование установленных мощностей электростанций ОЭС Северного и Северо-Восточного Китая

их параметров. Для этого типа объектов введен ряд отличных от электростанций параметров, например, графики суточной нагрузки, агрегированные объемы электрогенерирующих мощностей по типам станций и пр. Все объекты ООБД имеют как численные параметры, так и справочно-информационные (текстовые), например, название объекта, название региона, в котором он находится и пр.

Значения параметров объектов ООБД также используются в качестве входных данных для оптимизационной модели ОРИРЭС, являющейся важной функциональной частью ГИВС.

3. Представление результатов прогнозных расчетов в ГИВС

Математическая модель ОРИРЭС (далее Модель), позволяющая проводить исследования, направленные на поиск оптимальных стратегий развития и

режимов работы ЭЭС, используется при проведении расчетов различных перспективных вариантов формирования межгосударственных электроэнергетических объединений и входящих в них ЭЭС. При этом в Модели учитываются капитальные вложения, топливные затраты, годовые эксплуатационные издержки, представлены переменные, характеризующие установленные и рабочие мощности разных типов электростанций, пропускные способности электрических связей и другие факторы. Модель является линейной, что допустимо, учитывая долгосрочный характер решаемых задач и значительную мощность моделируемых энергосистем. Модель также является многоузловой, что позволяет учесть территориальную распределенность ЭЭС и МГЭО. В качестве узлов расчетной схемы принимаются районные, объединенные или национальные ЭЭС. Исследования с помощью модели проводятся на перспективу для заданного расчетного года (например, 2035 г.).

Целевой функцией модели являются затраты по МГЭО в целом, приведенные к годовой размерности. Они включают топливные затраты на производство электроэнергии различными типами электростанций, затраты на ввод новых мощностей и их эксплуатацию (инвестиционная и эксплуатационная составляющие), и стоимость ввода новых электрических линий — связей между узлами. Оптимальное решение определяется минимумом приведенных затрат по энергообъединению в целом.

Целевая функция оптимизируется в рамках определенных ограничений, в том числе балансовых. Например, рабочая мощность электростанций с учетом экспорта и импорта должна быть равна заданной нагрузке в каждом узле во все часы расчетного года, а суммарная установленная мощность электростанций в каждом узле должна быть не меньше максимальной нагрузки расчетного года (с учетом необходимого резервирования и обмена мощностью с другими узлами). Все указанные параметры модели настраиваются через специальный интерфейс ГИВС.

Система ГИВС использует общую алгебраическую систему моделирования GAMS (General Algebraic Modeling System) [16], которая запускается из интерфейса ГИВС, считывает заданные в интерфейсе параметры модели и формирует оптимальное решение на выходе (в виде набора определенных таблиц с соответствующими оптимальными значениями параметров модели). Графический блок ГИВС позволяет визуально оценить и интерпретировать выбранные параметры на картах, графиках, а также построить сводные таблицы. Результаты расчетов с использованием оптимизационной модели на выходе из GAMS характеризуются чрезвычайно большим объемом первичных данных (несколько тысяч строк).

На рис. 5. изображен пример малой части выходных данных из GAMS.

Многоаспектные результаты расчетов трудно представимы без специализированного таблично-графического интерфейса. Для восприятия такого объема информации экспертом и его дальнейшего анализа требуются специализированные процедуры по агрегированию результатов решения в различные виды сводных таблиц и диаграмм.

Авторами разработан специальный подход к представлению результатов расчетов с использованием оптимизационной модели. Объемные матрицы первичных подробных выходных данных переводятся в агрегированные шаблоны таблиц с результатами. В качестве примера приведем агрегированные таблицы с готовыми результатами прогнозирования развития электроэнергетических систем стран Северо-Восточной Азии на 2035 г. [15]. В табл. 1 приведены балансы электроэнергии по узлам/странам. В табл. 2 приведена структура выработки электроэнергии для различных типов электростанций в узлах.

Сформированные в данном примере агрегированные формы таблиц, содержащие некоторые результаты прогнозных расчетов для 2035 г. с использованием оптимизационной модели ОРИРЭС, понятны и удобны для анализа экспертами-энергетиками, которые являются основными (целевыми) пользователями ГИВС.

Разработанная авторами ГИВС значительно упрощает формирование подобных таблиц, способствует сокращению трудозатрат на поиск, проверку и формирование параметров для оптимизационной модели. Она помогает визуально представлять результаты полученных расчетов в графическом и картографическом видах, повышает качество проводимых исследований.

	A	B	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Operations capacity,															
2																
3	ОЭС Сибири															
4	Seasons	Hrs	ГЭС СибД	ГЭС СибП	ГЭС ВостД	ГЭС ВостП	ГАЭС	КЭС уг.	КЭС газ	ТЭЦ уг	ТЭЦ газ	ТЭЦ нефть	ТЭС уг.	ТЭС газ	ТЭС нефть АЭС	
5	W	1	18033,74		147,2			7745,4	835,92	14828,4	1737,9				2160	
6	W	2	18446,2		147,2			7745,4	835,92	14828,4	1737,9				2160	
7	W	3	18990,64		147,2			7745,4	835,92	14828,4	1737,9				2160	
8	W	4	19173,39		147,2			7745,4	835,92	14828,4	1737,9				2160	
9	W	5	19051,74		147,2			7745,4	835,92	14828,4	1737,9				2160	
10	W	6	18624,43		147,2			7745,4	835,92	14828,4	1737,9				2160	
11	W	7	18956,76		147,2			7745,4	835,92	14828,4	1737,9				2160	
12	W	8	19687,17		147,2			7745,4	835,92	14828,4	1737,9				2160	
13	W	9	20186,03		147,2			7745,4	835,92	14828,4	1737,9				2160	
14		10	20339,83		147,2			7745,4	835,92	14828,4	1737,9				2160	
15	W	11	20304,16		147,2			7745,4	835,92	14828,4	1737,9				2160	
16	W	12	20107,32		147,2			7745,4	835,92	14828,4	1737,9				2160	
17	W	13	19876,63		147,2			7745,4	835,92	14828,4	1737,9				2160	
18	W	14	19413,23		147,2			7745,4	835,92	14828,4	1737,9				2160	
19	W	15	18426,65		147,2			7745,4	835,92	14828,4	1737,9				2160	
20	W	16	17355,98		147,2			7745,4	835,92	14828,4	1737,9				2160	
21	W	17	16294,72		147,2			7745,4	835,92	14828,4	1737,9				2160	
22																
23	F	1	16821,36	110,4				6884,8	770,33	14004,6	1641,35				2160	
24	F	2	17316,19	110,4				6884,8	770,33	14004,6	1641,35				2160	
25	F	3	17569,08	110,4				6884,8	770,33	14004,6	1641,35				2160	
26	F	4	16841,09	110,4				6884,8	770,33	14004,6	1641,35				2160	
27	F	5	16297,33	110,4				6884,8	770,33	14004,6	1641,35				2160	
28	F	6	16181,92	110,4				6884,8	770,33	14004,6	1641,35				2160	
29	F	7	16634,01	110,4				6884,8	770,33	14004,6	1641,35				2160	
30	F	8	16469,13	110,4				6884,8	770,33	14004,6	1641,35				2160	
31																
32	Seasons	Hrs	ГЭС СибД	ГЭС СибП	ГЭС Вост											
33	F	1	16821,36	110,4				6884,8	770,33	14004,6	1641,35				2160	
34	F	2	17316,19	110,4				6884,8	770,33	14004,6	1641,35				2160	
35	F	3	17569,08	110,4				6884,8	770,33	14004,6	1641,35				2160	
36	F	4	16841,09	110,4				6884,8	770,33	14004,6	1641,35				2160	
37	F	5	16297,33	110,4				6884,8	770,33	14004,6	1641,35				2160	
38	F	6	16181,92	110,4				6884,8	770,33	14004,6	1641,35				2160	
39	F	7	16634,01	110,4				6884,8	770,33	14004,6	1641,35				2160	
40	F	8	16469,13	110,4				6884,8	770,33	14004,6	1641,35				2160	
41	F	9	16491,17	110,4				6884,8	770,33	14004,6	1641,35				2160	
42	F	10	16787,94	110,4				6884,8	770,33	14004,6	1641,35				2160	
43	F	11	17769,95	110,4				6884,8	770,33	14004,6	1641,35				2160	
44	F	12	18660,28	110,4				6884,8	770,33	14004,6	1641,35				2160	
45	F	13	18819,35	110,4				6884,8	770,33	14004,6	1641,35				2160	
46	F	14	17493,44	110,4				6884,8	770,33	14004,6	1641,35				2160	
47																

Рис. 5. Фрагмент выходных данных в формате Microsoft Excel — результатов решения оптимизационной задачи на модели ОРИРЭС

Таблица 1

Баланс электроэнергии по узлам (МВт·ч в год)

Название ОЭС	Произведено электроэнергии	Заряд ГАЭС*	Выдача (экспорт)	Приток (импорт)	Итого в узле
Восток (Россия)	87 698	—	-114 175	99 356	72 879
Сибирь (Россия)	408 048	—	-70 410	—	337 638
Сахалин (Россия)	7509	—	-64 829	63 289	5969
Япония	1 070 323	-19 344	-3103	137 869	1 185 745
Северная Корея	106 125	—	-92 784	67 080	80 421
Южная Корея	848 501	-4042	-85 566	86 081	844 974
Монголия	23 210	—	-2428	900	21 682
Северный Китай	2 969 707	-8198	-61 397	6798	2 906 910
Северо-Восточный Китай	1 130 992	-5515	-80 094	95 603	1 140 986

* ГАЭС — гидроаккумулирующая электростанция.

Таблица 2

Выработка электроэнергии различными типами электростанций в узлах (МВт·ч в год)

Название ОЭС	ТЭС			АЭС	ГЭС	ГАЭС	ВЭС	СЭС	Произведено электроэнергии
	Уголь	Газ	Нефть						
Восток (Россия)	31 245	13 471	—	—	42 982	—	—	—	87 698
Сибирь (Россия)	168 842	8962	17 832	—	212 412	—	—	—	408 048
Сахалин (Россия)	2448	5061	—	—	—	—	—	—	7509
Япония	233 824	199 511	266 845	123 226	167 184	15 475	19 719	44 539	1 070 323
Северная Корея	19 094	27 050	—	4124	53 498	—	1188	1171	106 125
Южная Корея	306 746	138 325	326 937	3774	14 117	3233	42 188	13 181	848 501
Монголия	17 930	—	—	—	4 086	—	1013	181	23 210
Северный Китай	2 682 344	60 693	85 872	—	23 777	6558	70 945	39 518	2 969 707
Северо-Восточный Китай	986 844	1677	24 074	—	50 296	4412	51956	11 733	1 130 992
Итого	4 449 317	454 750	721 560	131 124	568 352	29 678	187 009	110 323	6 652 113

Примечание: ТЭС — тепловая электростанция; АЭС — атомная электростанция; ГЭС — гидроэлектростанция; ВЭС — ветровая электростанция; СЭС — солнечная электростанция.

Заключение

Авторами разработана и совершенствуется специализированная геоинформационная вычислительная система для поддержки научных исследований в области формирования МГЭО, не имеющая аналогов в России и за рубежом. Разработанная авторами ГИВС кроме анализа и визуализации данных энергетического характера, позволяет экспертам проводить оптимиза-

ционные расчеты и строить прогнозы развития МГЭО, используя математическую оптимизационную модель ОРИРЭС. Ранее параметры для модели ОРИРЭС настраивались с использованием макросов Microsoft Excel, и любая доработка или изменение расчетной схемы требовали больших трудозатрат и времени. Разрабатываемая авторами ГИВС способствует автоматизации данного процесса, графической визуализации расчетов и повышению качества проводимых исследований.

Авторами предлагаются специализированные алгоритмы сбора/загрузки данных в объектно-ориентированную структуру, а также оригинальный способ хранения и обработки данных в ООБД, визуальное представление результатов расчетов с использованием оптимизационной модели ОРИРЭС в агрегированных таблицах, на графиках и картах.

В ходе работы над проектами по электроэнергетической кооперации России и стран Евразии будет проведена практическая апробация разрабатываемых алгоритмов и информационных технологий для выполнения прогнозных расчетов развития национальных и межгосударственных электроэнергетических систем. Разработанные авторами информационные технологии являются универсальными и могут применяться для исследований МГЭО в различных регионах мира.

Работа выполнена при финансовой поддержке гранта РФФИ № 18-07-00495 А.

Список литературы

1. Беляев Л. С., Подковальников С. В., Савельев В. А., Чудинова Л. Ю. Эффективность межгосударственных электрических связей. Новосибирск: Наука, 2008. 240 с.
2. Seung M. Role of Korea for Developing the East Asia Super Grid // 20th International Conference on Electrical Engineering. Jeju, Republic of Korea, 15–19 June 2014. Jeju, 2014. P. 45–52.
3. Yoon J., Park D., Kim H. Y. The Pre-feasibility Results of NEAREST between the ROK, and the DRPK, and RF // Proceedings of the 6th Intern. conf. Asian Energy operation: Forecast and Realities. Irkutsk, 7–11 September 2008. Irkutsk, 2008. P. 59–67.
4. Chimeddorj D. Power Sector of Mongolia, Regional cooperation policy // Northeast Asia Regional Power Interconnection Forum. (Beijing, China, 26–27 October 2016). Beijing, 2016. P. 3–26.
5. Ryo Omatsu. Interim Report by Asia International Grid Connection Study Group // 10th International Conference on Asian Energy Cooperation "AEC 2017". E3S Web of Conf. 2018. Vol. 27. P. 9–18.
6. Asia Pacific Energy Portal: Interactive data and policy information. URL: <http://asiapacificenergy.org/> (дата обращения 20.11.2017).
7. Xuming Liang. Application and research of global grid database design based on geographic information // Global Energy Interconnection Development and Cooperation Organization. 2018. Vol. 1, N. 1. P. 87–95.
8. Global Energy Storage Database. URL: <http://www.energystorageexchange.org/> (дата обращения 17.12.2017).
9. International Energy Agency. URL: <http://www.iea.org> (дата обращения 23.11.2017).
10. Energy Pedia. Portal: Power Plants: URL: https://enipedia.org/wiki/Portal:Power_Plants (дата обращения 15.01.2018).
11. Chudinova L. Yu., Podkovalnikov S. V., Trofimov I. L. Multilateral cooperation for power interconnection in Northeast Asia // 10th International Conference on Asian Energy Cooperation "AEC 2017". E3S Web of Conf. 2018. Vol. 27. P. 20–28.
12. Trofimov I. L. Using metadata to query the database on thermal economy of Russia through the Internet // Mathematical and Informational Technologies "MIT-2013". Vrnjacka Banya, Serbia, 05–14 September, 2013. University of Pristina, 2014. P. 706–714.
13. Трофимов И. Л., Трофимов Л. Н. Современные проблемы поиска и верификации данных об электростанциях Китая и других стран // Информационные и математические технологии в науке и управлении. 2017. № 4 (8). С. 120–128.
14. Трофимов Л. Н., Трофимов И. Л. Интернет-сервис для анализа статистических данных в области энергетики на примере мирового рынка нефти // Энергетика России в XXI веке. Инновационное развитие и управление: сб. ст. Всерос. конф. Иркутск, 1–3 сентября 2015. Иркутск: ИСЭМ СО РАН, 2015. С. 170–174.
15. Podkovalnikov S. V., Savelev V. A., Chudinova L. Yu. Prospects of electric-energy cooperation between Russia and Northeast Asian countries // Studies on Russian Economic Development. 2015. Vol. 26, Issue 4. P. 403–412.
16. Bussieck M. R., Meeraus A. General Algebraic Modeling System (GAMS). Chapter 8 / Kallrath J. (eds). Modeling Languages in Mathematical Optimization. Applied Optimization. Springer, Boston, MA. 2004. Vol. 88. P. 137–157.

The Data Processing and Geo Information System Technological Aspects to Support Research in the field of Prospective Interstate Electric Power Interconnections Development

S. V. Podkovalnikov, spodkovalnikov@isem.irk.ru, **L. N. Trofimov**, trofi@isem.irk.ru, Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, 664033, Russian Federation

Corresponding author:

Trofimov Ivan L., Senior Engineer, Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, 664033, Russian Federation,
E-mail: t_john88@mail.ru

Received on May 23, 2018

Accepted on July 19, 2018

The article describes Data Processing and Geo Information System (DPGIS), which is being developed in Melentiev Energy Systems Institute (ESI) of Siberian Branch of the Russian Academy of Sciences. The main purpose of developing software is to increase effectiveness of study and forecasting prospective interstate electric power connections. To study scenarios of expansion and operation of prospective electric power interconnections, special mathematical optimization model developed at ESI is used. Original technologies of data collection and storage in the Integrated Object-Oriented Database are offered. Technologies of data processing and representation in DPGIS Interface are considered. A computational experiment by DPGIS was performed; it has resulted in obtaining the most convenient aggregated forms for analysis and representation of optimization model solutions. Because of DPGIS application, efficiency and quality of scientific research in the field of prospective interstate electric power connections are increased.

Keywords: geo information system, optimization model, data processing, object database, power plant, electric power system, interstate electric power interconnection

Acknowledgements: This work was supported by the Russian Foundation for Basic Research, project no 18-07-00495 A.

For citation:

Podkvalnikov S. V., Trofimov L. N. The Data Processing and Geo Information System Technological Aspects to Support Research in the Field of Prospective Interstate Electric Power Interconnections Development, *Programmnaya Ingeneria*, 2018, vol. 9, no. 8, pp. 349–360.

DOI: 10.17587/prin.9.349-360

References

1. **Belyaev L. S., Podkval'nikov S. V., Savel'ev V. A., Chudinova L. Yu.** *Effektivnost' mezhgosudarstvennyh elektricheskikh svyazej* (Efficiency of Interstate Electric Interconnections), Novosibirsk, Nauka, 2008, 240 p. (in Russian).
2. **Seung M.** Role of Korea for Developing the East Asia Super Grid, *20th International Conference on Electrical Engineering*, Jeju, Republic of Korea, 15–19 June 2014, Jeju, 2014, pp. 45–52.
3. **Yoon J., Park D., Kim H. Y.** The Pre-feasibility Results of NEAREST between the ROK, and the DRPK, and RF, *Proceedings of the 6th Intern. conf. Asian Energy Cooperation: Forecast and Realities*. Irkutsk, 7–11 September 2008, Irkutsk, 2008, pp. 59–67.
4. **Chimededorj D.** Power Sector of Mongolia, Regional cooperation policy, *Northeast Asia Regional Power Interconnection Forum*, Beijing, China, 26–27 October 2016, Beijing, 2016, pp. 3–26.
5. **Ryo Omatsu.** Interim Report by Asia International Grid Connection Study Group, *10th International Conference on Asian Energy Cooperation "AEC 2017"*. E3S Web of Conf., 2018, vol. 27, pp. 9–18.
6. **Asia Pacific Energy Portal.** Interactive data and policy information, available at: <http://asiapacificenergy.org/>
7. **Xuming Liang.** Application and research of global grid database design based on geographic information, *Global Energy Interconnection Development and Cooperation Organization*, 2018, vol. 1, no. 1. P. 87–95.
8. **Global Energy Storage Database**, available at: <http://www.energystorageexchange.org/>
9. **International Energy Agency**, available at: <http://www.iea.org>
10. **Energy Pedia.** Portal: Power Plants, available at: https://enipedia.org/wiki/Portal:Power_Plants.
11. **Chudinova L. Yu., Podkvalnikov S. V., Trofimov I. L.** Multilateral cooperation for power interconnection in Northeast Asia, *10th International Conference on Asian Energy Cooperation "AEC 2017"*. E3S Web of Conf., 2018, vol. 27, pp. 20–28.
12. **Trofimov I. L.** Using metadata to query the database on thermal economy of Russia through the Internet, *Mathematical and Informational Technologies "MIT-2013"*, Vrnjacka Banya, Serbia, 05–14 September 2013, University of Pristina, 2014, pp. 706–714.
13. **Trofimov I. L., Trofimov L. N.** Sovremennyye problemy poiska i verifikacii dannyh ob elektrostancijah Kitaja i drugih stran (Modern problems of data search and data verification on power plants in China and other countries), *Informacionnye i matematicheskie tehnologii v naуke i upravlenii*, Irkutsk, 2017, vol. 4, no. 8, pp. 120–128 (in Russian).
14. **Trofimov L. N., Trofimov I. L.** Internet-servis dlja analiza statisticheskikh dannyh v oblasti jenergetiki na primere mirovogo rynka nefti (Internet service for statistical data analysis in the field of energy sciences, the example of the world oil market), *Jenergetika Rossii v XXI veke. Innovacionnoe razvitiye i upravlenie*, Irkutsk, 2015, pp. 170–174 (in Russian).
15. **Podkvalnikov S. V., Savel'ev V. A., Chudinova L. Yu.** Prospects of electric-energy cooperation between Russia and Northeast Asian countries, *Studies on Russian Economic Development*, 2015, vol. 26, issue 4, pp. 403–412.
16. **Bussieck M. R., Meeraus A.** General Algebraic Modeling System (GAMS). Chapter 8, Kallrath J. (eds), *Modeling Languages in Mathematical Optimization. Applied Optimization*, Springer, Boston, MA, 2004, vol. 88, pp. 137–157.

Я. А. Туровский, канд. мед. наук, доц., e-mail: yaroslav_turovsk@mail.ru,
Воронежский государственный университет,

А. С. Суровцев, аспирант, e-mail: alexander.surovtsev.vrn@gmail.com,
Воронежский государственный университет инженерных технологий

Программный пакет для автоматизированной обработки биомедицинских сигналов для поисковых исследований

Представлен программный пакет для автоматизированной обработки больших объемов биомедицинских сигналов, основным назначением которого является проведение поисковых исследований. Организация процесса обработки данных задается вычислительной схемой, т. е. определением набора вычислительных модулей и потоков данных между ними. Работа с данными рассматривается как совокупность трех этапов — создание выборки, препроцессинг и процессинг. В рамках решения задач обработки данных записей электроэнцефалограмм реализовано более 50 вычислительных модулей, позволяющих проводить частотно-временной, факторный, корреляционный и статистический виды анализа цифровых медико-биологических сигналов. Показан пример обработки записей электроэнцефалограмм, который заключается в формировании набора масштабированных во времени сигналов для проведения корреляционного анализа. Возможность расширять функции программного пакета за счет добавления новых модулей открывает широкие перспективы его дальнейшего использования при проведении поисковых исследований на основе анализа цифровых медико-биологических сигналов.

Ключевые слова: цифровая обработка биомедицинских сигналов, электроэнцефалограмма, поисковые исследования медико-биологических данных

Введение

Анализ показателей активности органов и систем органов человека, таких как электроэнцефалограмма (ЭЭГ), электромиограмма (ЭМГ) и вариабельность сердечного ритма (ВСР) широко используется в настоящее время. Спектр задач, которые решают с использованием анализа биомедицинских данных, достаточно широк. К их числу относятся задачи клинической медицины, которые направлены на выявление нарушений или отклонений в человеческом организме, задачи построения систем контроля деятельности человека-оператора в целях предотвращения аварийных ситуаций [1], например, на опасных химических производствах. Решение последних сводится к поиску предикторов отклонения состояния оператора [2], что достаточно сложно с учетом его индивидуальных особенностей. В качестве примера рассмотрим круг задач, в которых предполагается использование ЭЭГ. Следует отметить, что записи некоторых биомедицинских параметров представляют собой набор нескольких сигналов. Электроэнцефалограмма — результат активности большого числа микросистем нейронов. Отведение (канал) ЭЭГ задает участок головного мозга, биоэлектрическая активность которого регистрируется. После оцифровки ЭЭГ в виде набора цифровых сигналов

может быть обработана с привлечением алгоритмов цифровой обработки сигналов (ЦОС).

Работу с биомедицинскими данными в поисковых исследованиях, например, в оценке параметров деятельности человека-оператора на опасных производствах, можно разложить на три этапа: создание выборки данных, препроцессинг и процессинг. Создание выборки подразумевает либо регистрацию данных в соответствии с целью или задачей эксперимента, либо получение этих данных из каких-то баз. Препроцессинг является подготовкой данных к обработке. На этапе процессинга выполняются действия, направленные на обработку данных с последующим выделением интересующих результатов. Поэтому развитие ЦОС и компьютерных наук в значительной мере определяют подходы и методы в анализе биомедицинских сигналов (БМС).

Поисковые исследования БМС часто сводятся к перебору большого количества различных управляющих параметров, присутствующих в тех или иных алгоритмах и направленных на решение задач, связанных с обработкой значительного объема данных. Нередко потребность в использования набора различных алгоритмов при выполнении поисковых исследований вносит дополнительные требования к исследователю. Такие требования выражаются

в наличии у исследователя навыков ЦОС и высокой квалификации в программировании.

Рассматривая современный уровень развития программ и алгоритмов в задачах анализа БМС, следует отметить, что существует много программного обеспечения (ПО) для анализа биомедицинских сигналов, таких как ЭЭГ, ВСР и ЭМГ. Одним из широко распространенных модулей, используемых для обработки ЭЭГ, является модуль EEGLAB [3] к пакету прикладных программ MATLAB. Особенностью данного модуля является наличие широкого спектра средств визуализации биомедицинских данных. Для проведения вычислений требуется подключение дополнительных модулей к MATLAB, что увеличивает сложность в разработке, тестировании и поддержке ПО. Библиотеки numpy и scipy [4] языка программирования Python, а также библиотеки языка R [5] обладают большим набором алгоритмов ЦОС. Однако такие решения (готовые библиотеки сторонних разработчиков) в ряде случаев могут быть сложны. В первую очередь, речь идет о содержательной интерпретации для пользователей, не имеющих специализированного технического или физико-математического образования.

Материалы и методы

В свете перечисленного выше, целью работы, результаты которой представлены в статье, является разработка программного пакета для автоматизированной обработки биомедицинских сигналов в поисковых исследованиях.

Для эффективной работы такого пакета одними из главных требований, которым должно соответствовать такое ПО, являются расширяемость и возможность распараллеливания вычислений с последующим переносом вычислений части алгоритмов, например, на высокопроизводительные графические процессоры CUDA от NVidia [6] или на вычислительные кластеры. С учетом этого требования в качестве языка программирования был выбран язык C# (C Sharp) [7]. В первую очередь, такой выбор обусловлен тем, что он кроссплатформенный. Это означает, что разработанное с помощью этого языка ПО можно запускать на исполнение в окружении различных операционных систем таких семейств, как Windows, Linux и Mac OS.

Учитывая современные представления о структуре медико-биологических сигналов, необходимо включать в набор программ модули по работе с периодическими сигналами как в частотном, так и во временном доменах [9], а также набор программ по проведению корреляционного анализа. Включение статистических методов, как параметрических, так и непараметрических расширяет возможности поисковых исследований. На рис. 1 изображена диаграмма потоков данных (DFD) [8] между вычислительными модулями разработанного программного пакета. Прямоугольниками показаны сущности (записи хранилища данных). Овалы задают процессы. Потоки

данных показаны стрелками. Прямоугольники и овалы, не выделенные полужирным контуром, могут быть исключены из цепочек вычислений. Множество вариантов обработки данных образуется маршрутами из стрелок, причем начало и конец каждого маршрута должны быть прямоугольниками.

Реализация программного обеспечения

Процесс обработки данных представлен в виде последовательности вычислительных модулей (создания выборки данных, препроцессинга и процессинга). Каждый модуль обязан задавать число слотов для входных и выходных связей. Число входных слотов указывает количество файлов, которое требуется подать на вход данному модулю. Число выходных слотов показывает количество файлов, полученных в результате вычисления.

Рассмотрим пример модуля, который вычисляет прямое непрерывное вейвлет-преобразование. Один из вариантов реализации данного модуля предполагает, что этот модуль будет иметь один входной слот и один выходной. В единственный входной слот модуля записывается имя файла с сигналом, который требуется преобразовать. Выходной слота этого модуля после его исполнения будет содержать имя файла с матрицей непрерывного вейвлет-разложения.

Схема вычислений задается связями между вычислительными модулями, по которым передаются наборы имен файлов. Со стороны пользователя процесс вычисления выглядит следующим образом — сначала пользователь составляет схему вычислений, а потом запускает ее на выполнение. Составление схемы заключается в наборе последовательности вычислений из различных модулей и задании связей между выходными и входными слотами модулей, а также в указании значений управляющих переменных каждого из модулей в последовательности. Процесс обработки данных, запущенный на выполнение, проходит два шага до завершения. Первым шагом является верификация связей в схеме и валидация значений управляющих переменных. Вторым шагом является поочередное исполнение модулями составленной схемы. Причем каждый модуль выполняет серию вычислений. Ошибка на одном из шагов (валидация и выполнение модулей) ведет к внебратному завершению вычислений с обязательным указанием причины, которая повлекла прекращение работы программы.

Если рассматривать процесс обработки данных как совокупность преобразований набором модулей, то продукт выполнения модуля может быть представлен как совокупность "результата" и "дополнительного эффекта" (*side effect*, побочное действие) [10]. "Результатом", согласно такому рассмотрению (процесс обработки данных как совокупность преобразований модулями, потоками информации между модулями есть коллекции файлов), является дескриптор (указатель) на данные, например, название файла, а "дополнительным эффектом" — новые



Рис. 1. Диаграмма потоков данных между вычислительными модулями разработанного программного пакета

данные, записанные в хранилище. Термин "дополнительный эффект" использован в данном случае по той причине, что при рассмотрении процесса с точки зрения вычислительных модулей, основным "результатом" являются дескрипторы данных (имена файлов), а вычисление новых данных является вторичным. Однако этот термин используется только с этой точки зрения. Иными словами, процесс обработки данных с точки зрения модулей схемы — трансформация потока имен файлов в соответствии со значениями управляющих переменных. Таким образом, имя выходного файла определяется именами файлов, поданных на входные слоты, содержащими этих файлов и значениями управляющих переменных.

Было отмечено, что "дополнительный эффект" в рассматриваемом случае означает данные, записанные на жесткий диск. Так как нет никаких ограничений на то, чтобы один и тот же набор файлов был подан на вход двум модулям, необходимо обеспечить механизмы, позволяющие избежать

коллизии имен файлов внутри схемы вычислений, чтобы предотвратить перезапись данных. Такое требование может быть достигнуто путем введения ограничения, согласно которому большинство модулей, имеющих "дополнительный эффект", должны требовать от объекта класса настроек уникальное, как минимум в рамках данного процесса вычислений, имя директории, в которую следует сохранять данные.

В соответствии с изложенным выше, продуктом выполнения некоторых модулей может являться только "результат", выраженный в списке файлов. Продуктом выполнения других модулей может являться только "дополнительный эффект", представленный данными, запись которых была выполнена на рассматриваемым модулем. Продуктом работы остальных программ является наличие как "результата" (коллекции файлов), так и "дополнительного эффекта" (новых данных, записанных на диск). На рис. 2 показан пример графического интерфейса для одного из модулей вычислений.



Рис. 2. Пример графического пользовательского интерфейса одного из модулей вычислений:

1 — область для указания номеров модулей и выходных слотов, из которых будет осуществляться перенаправление данных; 2 — номер модуля в последовательности вычислений; 3 — область с элементами задания управляющих переменных; 4 — область элемента, в которой содержится описание исполняемого модуля; 5 — название модуля; 6 — число выходных слотов программы

Результаты и обсуждение

Решение задач обработки данных сигналов ЭЭГ потребовало реализации более 50 вычислительных модулей. Рассмотрим основные из них.

Модули создания выборки и проведения препроцессинга представлены следующим набором: модули по формированию списка файлов для обработки, модуль по конвертированию файлов в EDF-формат [11] (для удобства передачи данных в программы биомедицинских исследований для дальнейшей визуализации и обработки) и модули ЦОС.

Вычислительные модули ЦОС препроцессинга: модуль для свертки сигналов, модуль для растяжения или сжатия сигнала, модуль децимации, удаления линейного тренда и расчета скользящей средней. Модуль для растяжения или сжатия сигналов представляет собой реализацию метода формирования масштабированных во времени последовательностей, который позволяет выполнить формирование пула сигналов для проведения корреляционного анализа [12].

Набор модулей процессинга представлен модулями прямого и обратного преобразования Фурье; модулями прямого и обратного непрерывного вейвлет-преобразования; модулями по работе с цепочками локальных минимумов и максимумов матрицы непрерывного вейвлет-разложения (позволяют выполнить восстановление сигнала в заданной частотно-временной области); модулем, проводящим факторный анализ, модулями по расчету корреляций Пирсона и Спирмана; модулями для расчета статистических критериев (Манна-Уитни, Вилкоксона, Фридмана и Краскела-Уоллиса) [13].

Модули прямого и обратного непрерывного вейвлет-преобразований, а также модули прямого и обратного Фурье-преобразований выполнены с использованием механизмов, представляющих возможность проводить расчеты в том числе и с использованием вычислительных ресурсов графических процессоров CUDA от NVidia. Сравним время, которое необходимо для расчета тестовых данных. Длину тестового сигнала для прямого непрерывного вейвлет-преобразования возьмем равной 5000 отсчетов. Прямое преобразование будем рассчитывать для 2048 масштабов. Обратное непрерывное преобразование выполним для матриц размером 5000×2048 . В качестве вейвлетов в обоих случаях будем использовать вейвлеты Morlet и Wave. Четырехъядерный процессор Intel 7700hq справляется с данными задачами (как с прямым, так и с обратным непрерывными вейвлет-преобразованиями) в среднем за 40 мин. Видеокарте NVidia GTX 1070 с 8 Гбайтами оперативной видеопамяти и 2048 CUDA-ядрами требуется в среднем 8 с для таких расчетов (прямого и обратного непрерывных вейвлет-преобразований). Результаты тестов показали уменьшение времени, которое необходимо на расчеты, примерно в 300 раз при использовании вычислительных ресурсов видеокарт. Таким образом, реализована возможность увеличения массива данных, подвергнутого частотно-временному анализу при проведении поисковых исследований, в сотни раз, если для расчетов использовать видеокарты NVidia, при том же времени обработки, что и при использовании для расчетов современных процессоров персональных компьютеров и ноутбуков.

Пример схемы обработки записей ЭЭГ

В качестве примера рассмотрим расчет сжатия/растяжения сигналов. На рис. 3 показана диаграмма потока данных с выделенным маршрутом их преобразования, соответствующим данному расчету. Рис. 4 показывает вид графического интерфейса разработанного программного пакета. На рис. 5 показан графический пользовательский интерфейс для всех пяти вычислительных модулей, участвующих в вычислении одного из вариантов получения растянутых/сжатых во времени сигналов. Как показано на рис. 4 и 5, процесс вычисления растянутых/сжатых во времени сигналов можно выполнить с помощью цепочки вычислений, образованной из пяти элементов.

Ранее было отмечено, что процесс работы с биомедицинскими сигналами при проведении поисковых исследований можно разделить на три этапа — создание набора данных, препроцессинг и процессинг. Покажем какие элементы цепочки вычислений соответствуют каждому из этапов. Этап создания набора данных представлен "вычислителем списка файлов".

Препроцессинг составлен из модуля прямого преобразования Фурье и двух модулей растяжения или сжатия сигналов (по одному на вещественные и комплексные компоненты). Процессинг состоит из модуля обратного преобразования Фурье.

Таким образом, разработанный программный пакет позволяет пользователям формировать программы для проведения поисковых исследований

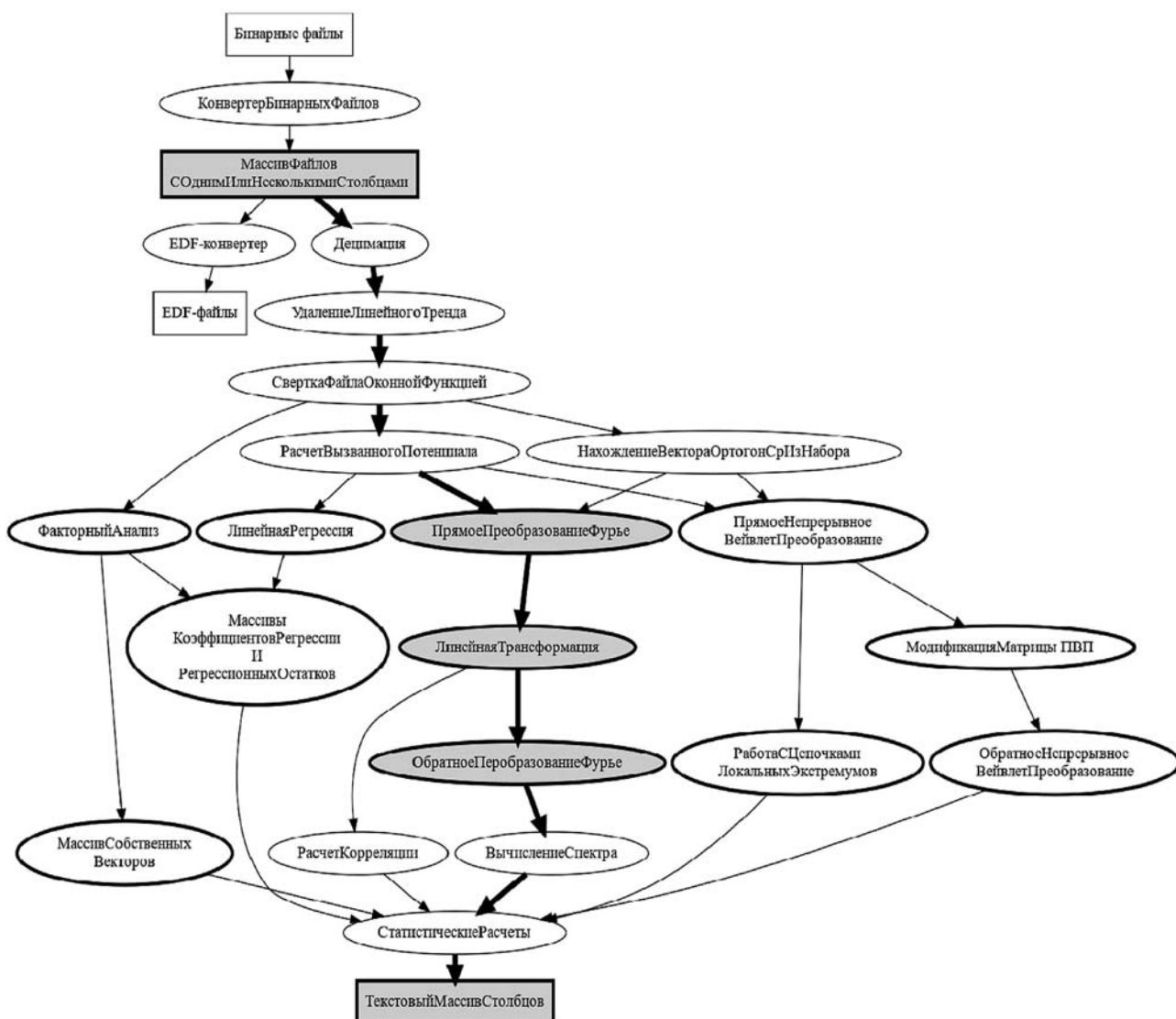


Рис. 3. Диаграмма потока данных с выделенным маршрутом их преобразования: выделен маршрут, который соответствует расчету растянутых/сжатых во времени сигналов; полужирными стрелками показан поток данных; прямоугольники и овалы с серым цветом показывают сущности и процессы соответственно, вовлеченные в данный расчет

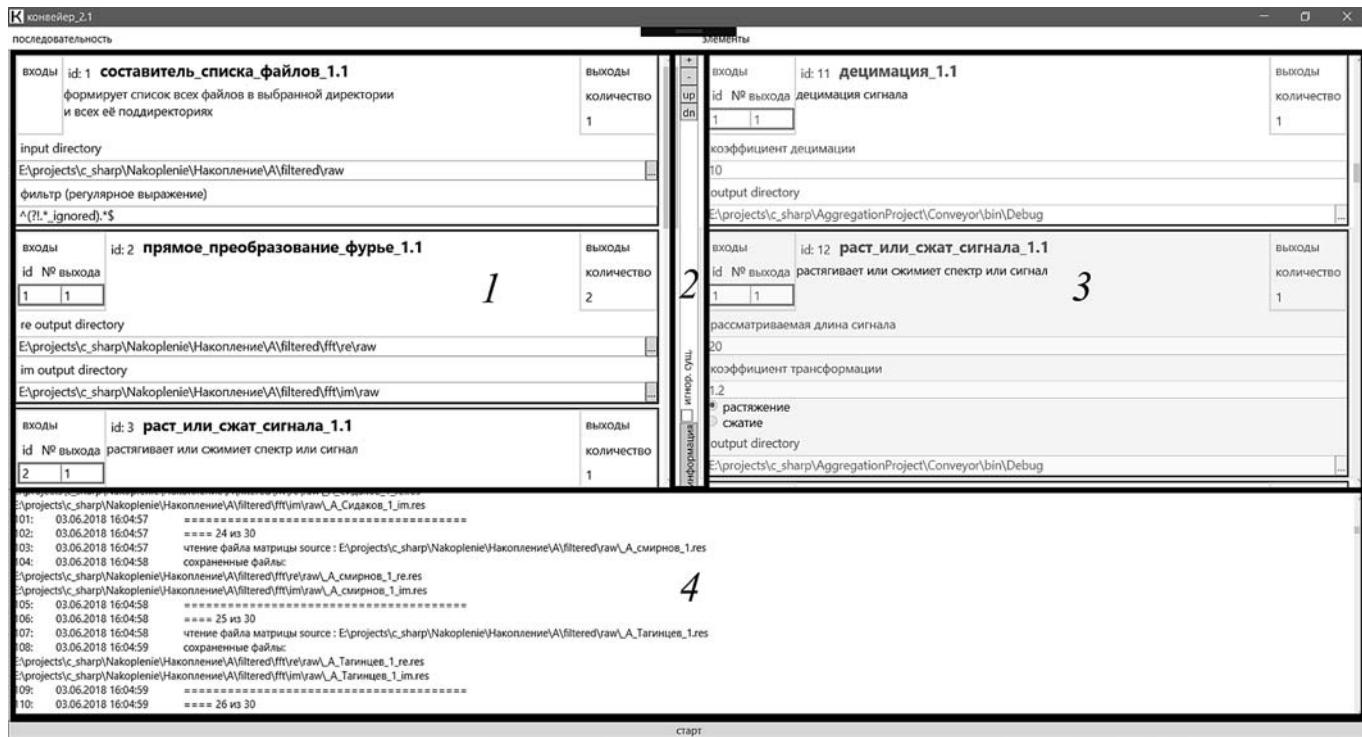


Рис. 4. Расчет растянутых во времени сигналов:

1 — область с построенной цепочкой вычислений; 2 — элементы управления; 3 — список доступных модулей для построения цепочек вычислений; 4 — область, в которую выводится диагностическая информация

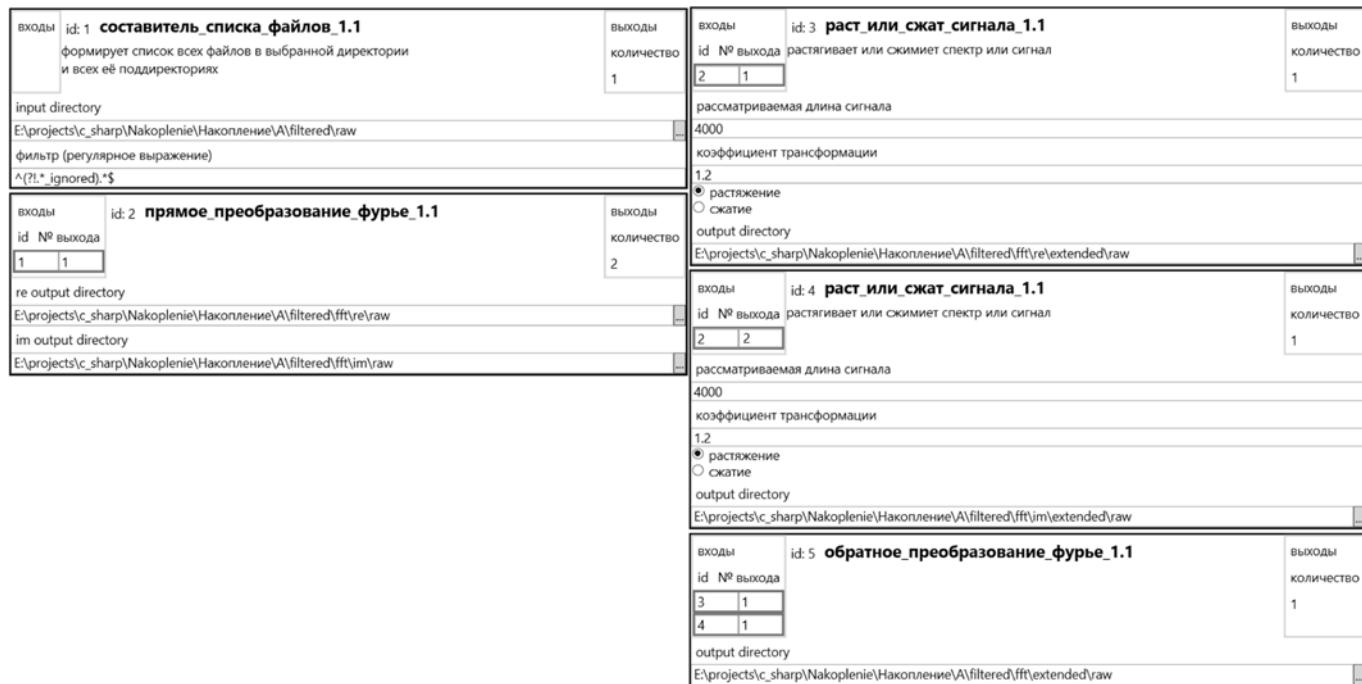


Рис. 5. Графический пользовательский интерфейс элементов цепочки вычислений одного из вариантов реализации вычислений для получения растянутых/сжатых во времени сигналов

БМС, что показано на примере получения растянутых/сжатых во времени сигналов. Причем процесс обработки данных проходит автономно и позволяет обрабатывать большие объемы данных.

Заключение

Разработан программный пакет для автоматизированной обработки медико-биологических сигналов при проведении поисковых исследований. Программный пакет состоит из управляющей программы и набора вычислительных модулей. Вся логика по организации и проведению процесса вычислений содержится в управляющей программе. Организация процесса вычислений заключается в построении вычислительной схемы в рамках программы. Проведение вычислений — запуск вычислительных модулей на выполнение и передача данных между ними (вычислительными модулями). Требования к каждому вычислительному модулю регламентированы и довольно просты. Это позволяет абстрагироваться от организации процесса вычислений и сосредоточиться на логике рассматриваемого этапа вычислений при разработке новых вычислительных модулей, что открывает широкие перспективы дальнейшего использования данного программного пакета. Кроме этого, удалось добиться увеличения производительности вычислений за счет реализации нескольких наиболее требовательных к вычислительным ресурсам алгоритмов. К их числу относятся: прямое и обратное непрерывные вейвлет-преобразования; прямое и обратное преобразования Фурье на высокопроизводительных процессорах CUDA от NVidia.

Список литературы

1. Долгов Д. И., Ильина И. Е. Эргатические системы управления как фактор повышения конкурентоспособности предприятия // Общество: политика, экономика, право. 2011. № 1. С. 47–52.
2. Сороко С. И., Бекшаев С. С., Рожков В. П. ЭЭГ Корреляты генофенотипических особенностей возрастного развития мозга у детей аборигенного и пришлого населения северо-востока России // Российский физиологический журнал им. Сеченова. 2012. № 1. С. 3–26.
3. EEGLab Documentation. URL: https://sccn.ucsd.edu/wiki/EEGLAB_Wiki#Documentation.
4. scipy documentation. URL: <https://docs.scipy.org/doc>
5. R documentation. URL: <https://www.rdocumentation.org/>
6. CUDA toolkit documentation. URL: <https://docs.nvidia.com/cuda>
7. Troelsen A. Pro C# 5.0 and the .Net 4.5 Framework. Apress. 2012. 1487 p.
8. Rumbaugh J., Rumbaugh J., Jacobson I., Booch G. The Unified Modelling Language. Reference Manual. Second edition, 2004. 722 p.
9. Кузнецов С. Ю., Боровик Д. В., Виноградова О. Л. Использование вейвлет преобразования для анализа поверхности ЭМГ // Физиология мышечной деятельности. 2010. № 1. URL: <http://phmag.imbp.ru/articles/Kuznetcov.pdf>.
10. Petricek T., Skeet J. Functional Programming for the Real World with examples in F# and C#. Manning, 2009. 560 p.
11. Kemp B., Olivian J. Full specification of EDF+. 2003. URL: <https://www.edfplus.info/specs/edfplus.html>.
12. Туровский Я. А., Борзунов С. В., Вахтин А. А., Суровцев А. С., Алексеев А. В. Алгоритмы генерации масштабируемых во времени сигналов в задачах обработки медицинских данных // Цифровая обработка сигналов. 2018. № 1. (XX-1). С. 238–241.
13. Bickel P., Diggle P., Fienberg S., Krickeberg K., Ilkin I., Wermuth N., Zeger S. A Distribution-Free Theory of Nonparametrics Regression. Springer, 2002. 647 p.

Software Package for Automatic Data Processing of Biomedical Signals for Exploratory Researches

Ya. A. Turovsky, yaroslav_turovsk@mail.ru, Voronezh State University, Voronezh, 394051, Russian Federation,
A. S. Surovtsev, alexander.surovtsev.vrn@gmail.com, Voronezh State University of Engineering Technologies, Voronezh, 394065, Russian Federation

Corresponding author:

Turovsky Yaroslav A., PhD, Voronezh State University, 394051, Voronezh, Russian Federation,
E-mail: yaroslav_turovsk@mail.ru

Received on June 16, 2018

Accepted on July 17, 2018

This paper presents a software package for automatic processing of large amount of biomedical signals. The key purpose of this package is to conduct exploratory researches. The organization of the data processing is defined by specifying of computational scheme, that is, by determining a set of computational modules and data flows between this set of modules. Data processing is considered as a sum of three stages — sampling, preprocessing, and processing. While doing tasks of processing of electroencephalogram data more than 50 computational modules are

created. These modules can be used to do the time-frequency, factor, correlation, and statistical analysis of biomedical data. There is a sample of electroencephalogram records processing, which is aimed in the formation of a set of time-scaled digital signals for carrying out the correlation analysis. The possibility of extending the functions of this program package due to the addition of new computational modules opens up wide prospects for the further usage of it in the exploratory researches of biomedical signals.

Keywords: digital processing of biomedical signals, electroencephalogram, exploratory research of biomedical data

For citation:

Turovsky Ya. A., Surovtsev A. S. Software Package for Automatic Data Processing of Biomedical Signals for Exploratory Researches, *Programmnaya Ingeneria*, 2018, vol. 9, no. 8, pp. 361–368.

DOI: 10.17587/prin.9.361-368

References

1. Dolgov D. I., Ilina E. I. Ergaticheskie sistemy upravleniya kak faktor povisheniya konkurentospособnosti predpriyatiya, *Obshchestvo, politika, ekonomika, pravo*, 2011, no. 1, pp. 47–52 (in Russian).
2. Soroko S. I., Beshkaev C. C., Rozhkov V. P. EEG Korrelyaty genofenotipicheskikh osobennostey vozrastnogo razvitiya mozga u detey aborigennogo i proshlogo naseleniya severo-vostoka Rossii, *Rossiyskiy fiziologicheskiy zhurnal im. Sechenova*, 2012, no. 1, pp. 3–26 (in Russian).
3. **EEGLab** Documentation, available at: https://sccn.ucsd.edu/wiki/EEGLAB_Wiki#Documentation.
4. **scipy** documentation, available at: <https://docs.scipy.org/doc>.
5. **R documentation**, available at: <https://www.rdocumentation.org/>
6. CUDA toolkit documentation, available at: <https://docs.nvidia.com/cuda>.
7. Troelsen A. *Pro C# 5.0 and the .Net 4.5 Framework*, Apress, 2012, 1487 p.
8. Rumbaugh J., Jacobson J., Booch G. *The Unified Modelling Language*. Reference Manual, 2004, Second edition, 722 p.
9. Kuznetsov S. U., Borovik D. V., Vinogradova O. L. Ispolzovaniye veyvlet preobrazovaniya dlya analiza poverhnosti EMG, *Fiziologiya mishechnoy deyatelnosti*, 2010, no. 1, available at: <http://phmag.imbp.ru/articles/Kuznetcov.pdf> (in Russian).
10. Petricek T., Skeet J. *Functional Programming for the Real World with examples in F# and C#*, Manning, 2009, 560 p.
11. Kemp. B., Olivan J. *Full specification of EDF+*, 2003, available at: <https://www.edfplus.info/specs/edfplus.html>.
12. Turovsky Ya. A., Borzunov S. V., Vahtin A. A., Surovtsev A. S., Alekseev A. V. Algoritmy generatsii mashtabiruemych vo vremeni signalov v zadachah obrabotki medicinskikh dannih, *Cifrovaya obrabotka signalov*, 2018, no. 1. (XX-1), pp. 238–241 (in Russian).
13. Bickel P. A., Diggie P., Fienberg S., Krickerberg K., Olkin I., Wermuth N., Zeger S. *Distribution-Free Theory of Nonparametrics Regression*, Springer, 2002, 647 p.

ИНФОРМАЦИЯ

Начинается подписка на журнал "Программная инженерия" на первое полугодие 2019 г.

Оформить подписку можно через подписные агентства
или непосредственно в редакции журнала.

Подписной индекс по объединенному каталогу

"Пресса России" — 22765

*Адрес редакции: 107076, Москва, Стромынский пер., д. 4,
Издательство "Новые технологии",
редакция журнала "Программная инженерия"*

Тел.: (499) 269-53-97. Факс: (499) 269-55-10. E-mail: prin@novtex.ru

N. Esmaeilishahmirzadi, Postgraduate Student, nasibe.smaeili@gmail.com,
Lobachevsky State University, Department of Computer Science, Nizhny Novgorod, Russian
Federation, **H. Mortezaour**, hr.mortezapour@gmail.com, Postgraduate Student Ferdowsi
University of Mashhad, Department of Computer Engineering, Mashhad, Iran

Corresponding Author:

Esmaeilishahmirzadi Nasibeh, Postgraduate Student, Lobachevsky State University, Nizhny Novgorod, Russian Federation,
E-mail: nasibe.smaeili@gmail.com

Preprocessing for Enhancing the Classification of Pulmonary Data Sets using Convolutional Neural Networks

Received on July 17, 2018
Accepted on August 13, 2018

With growing lung cancer in the world the requirement of computer-aided diagnosis systems (CADs) has increased. In this work we present a method for enhancement of the data collection of pulmonary images using the method of data augmentation and filters in image processing. Our work consists of two components. We firstly present a novel method of image enhancement and secondly classify images into either nodule or non-nodule. For nodule classification the Residual convolutional neural network is proposed, which achieved better performance than state-of-the-art networks in our dataset.

Images are extracted from the LUNA16 database derived from the LIDC-IDRI database. For image enhancement we use compound images obtained from image processing filters. These filters are like Gabor and Average and rotate at 90 and 180 degrees to produce images similar to LUNA16 images. We produce 8 images similar to the original image and combine them into one image. We thus produce a new dataset. After testing various convolutional networks on a new dataset we found that the Residual convolutional neural network (ResNet) provides an accuracy of 96.214 % in the classification task. The accuracy demonstrates that the proposed method performs desirably and acceptably compared to other methods.

Keywords: pulmonary nodule, lung images, residual neural network, deep learning, convolutional neural network CNN

Introduction

Lung cancer is a leading cause of death by cancer in both men and women in the United States, accounting for 27 % of cancer deaths in 2014 [1]. The increase of pulmonary disease, especially lung cancer, has increased the need of a modern system for early diagnosis of the disease.

Lung cancer is caused by uncontrollable irregular growth of cells in lung tissue. These lung tissue abnormalities are often called lung nodules. They are small and roughly spherical masses of tissue, usually about 5 to 30 mm long [2]. A quantitative analysis of pulmonary nodules using computer tomography (CT) is the main focus of early diagnosis of cancer [3].

Therefore one of the main challenges in lung images is the classification of images to high precision that makes other operations and retrieval of images easier. Denoising and enhancement are also the main factors required in medical Imagery [4].

In this paper we present a technique for enhancement of datasets and classification of pulmonary nodules by using convolutional neural network. Image classification is the process of dividing an image into two or more groups. Image classification helps us to obtain a more detailed analysis of a task. We have classified images using deep learning methods. The database used in this paper

is derived from the largest reference database available for pulmonary nodules, called LIDC-IDRI [5].

Literature Review

The use of deep [6], in particular convolutional, neural networks has recently increased in medical diagnostics. In 2015 Roth et al. [7] used convolutional neural networks to detect spleen sclerotic metastasis, lymph nodes, and colon polyps. Their research has shown that neural networks are highly conducive to using CADs systems. They also increased the sensitivity and recall rates between 15 and 30 % on each issue. In 2016 Setio et al. [8] proposed a new computer diagnostic (CAD) system for pulmonary nodules using multidimensional convolutional networks that automatically learn discriminative features from training data. The sensitivity of this system is 85.4 and 90.1 % in 1 and 4 positive errors per scan. In the same year Sun et al. [9] designed deep learning algorithms and, including Convolutional Neural Network (CNN), implemented Deep Belief Networks (DBNs), Stacked Denoising Autoencoder (SDAE), and compared traditional methods of computer diagnosis and feature learning on the LIDC-IDRI dataset for the purpose of classifying pulmonary nodules. The

result of their research showed the accuracy of 0.797 in convolutional neural networks, which slightly increased compared to the traditional method with the accuracy of 0.7940. Anthimopoulos et al. [10] used convolution neural networks to evaluate and classify lung disease patterns. Their proposed network consists of 5 convolutional layers with a 2×2 kernel and the Leaky ReLU activity function. The accuracy of 0.8561 shows the success of convolutional neural networks in classifying pulmonary patterns.

Ramaswamy et al. [11] experimented with some CNN such as AlexNet and GoogleNet on the LUNA16 dataset. Their best model successfully classifies 89.6 % of the actual nodules and significantly reduces false positives when compared to previous attempts of the classification task.

Methodology

Our work in this paper consists of two parts: enhancement of collection of data and classification of images in an optimal manner. Reduction of false positive in the classification section is also a goal of our work.

Our proposed method has used the data augmentation method and various filters to enhance the dataset. Eight similar images are produced of each image and they are combined into one image. A new dataset has been tested on different convolutional neural networks for the purpose of classifying images. Its accuracy is then compared to similar tasks.

Convolutional Neural Network. Profound network learning was also facilitated by emerging the residual learning framework. The ResNet [12] architecture was introduced in 2015 and won many competitions. This architecture with a depth of 152 layers was the deepest architecture by 2015 and remains one of the deepest architectures till present day.

Deep residual networks (ResNets) [13] consist of many stacked Residual Units. Each unit can be expressed in a general form:

$$\begin{aligned} yl &= h(xl) + F(xl, WI), \\ xl + 1 &= f(yl). \end{aligned}$$

Where xl and $xl + 1$ are input and output of the l -th unit and F is a residual function. In [12] $h(xl) = xl$ is an identity mapping and f is a ReLU [14] function.

Many years before the introduction of ResNet better performance could be achieved by increasing the depth and width of the network. These cases were repeatedly confirmed in previous years by architectures such as AlexNet [15] and GoogleNet [16]. Due to the problem of vanishing the gradient the network practically had a high training error. But ResNet proved that a deep network with optimal efficiency can be found in practice.

The advantage of ResNet is that it is easily optimized, and high accuracy with deep layers can be achieved in this architecture. Their paper [13] described how they optimized the network and proposed a method that makes training easier and improves generalization by residual block. We use their method in this work. This network was tested on the LUNA16 dataset, and with a little manipulation we arrived at an acceptable accuracy which is better than other networks.

In [11] which was performed on the LUNA16 dataset in 2016, have been examined two architectures of the AlexNet and GoogleNet networks. We have examined ResNet in the following manner to be comparable with that work.

Data Enhancement. Data augmentation [17] includes methods to improve the learning process by increasing its dataset and preventing overfitting. This method uses rotation, zoom, and shift to produce images similar to dataset images. Both original and produced images are fed into the convolutional neural network, and the network learns the image data from several angles or filters. Thus for a dataset of size N we generate a dataset of size $2N$ or more [17].

Our focus is on data enhancement, combining data augmentation with image processing filters, and combining into an image. In this case, by adding new features to the image we will improve the distinction by convolutional networks. Images in the 64×64 format are rotated 180° clockwise. The resulting image will be the same as the image of the dataset, which is 180° clockwise. Both nodule and non-nodule images are then rotated 90° counterclockwise. The reason is that the neural network can see the images in different situations with different rotation and learn them better. The other idea is using linear and nonlinear filters and high-pass and low-pass filters. The Gabor [18] filter is a powerful tool for texture analysis because of the multiple separability properties in the spatial and frequency domains. The two-dimensional Gabor function is as follows:

$$\begin{aligned} g(x, y; \lambda, \theta, \psi, \sigma, \gamma) &= \\ &= \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \psi\right)\right), \end{aligned}$$

where λ , θ , ψ , σ , γ represents the wavelength of the sinusoidal factor; θ represents the orientation of the normal to parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma/standard deviation of the Gaussian envelope, and γ is the spatial aspect ratio and specifies the ellipticity of the support of the Gabor function.

Gabor functions have been widely used in various computer vision applications, such as texture analysis and edge detection, feature extraction, etc. The Gabor filter is linear and local. If its parameters are properly adjusted, we have a very good function identifying texture features.

The Gabor filter is based on two parameters of scale and rotation. Using different scales and rotations produce a number of filter banks that use filters to extract features from the image and their applications. There are certain parameters that affect the output of the Gabor filter. The first one is the size of the Gabor filter kernel (k). Like many convolution kernels, the filters should be square because of kernel uniformity. We used 3 types of Gabor filters with different parameters. We have a 10×10 kernel in 2 states and a 5×5 kernel in one state. Sigma (σ) is the standard deviation of the Gaussian function used in the Gabor filter. We considered sigma values in all 3 states equal to 5. This parameter controls the Gaussian width used in the Gabor kernel. The most important Gabor filter parameter is the theta (θ) which shows the main direction for the application of the Gabor filter. This parameter decides which image features will be in the response. For example, a theta with a value of zero means

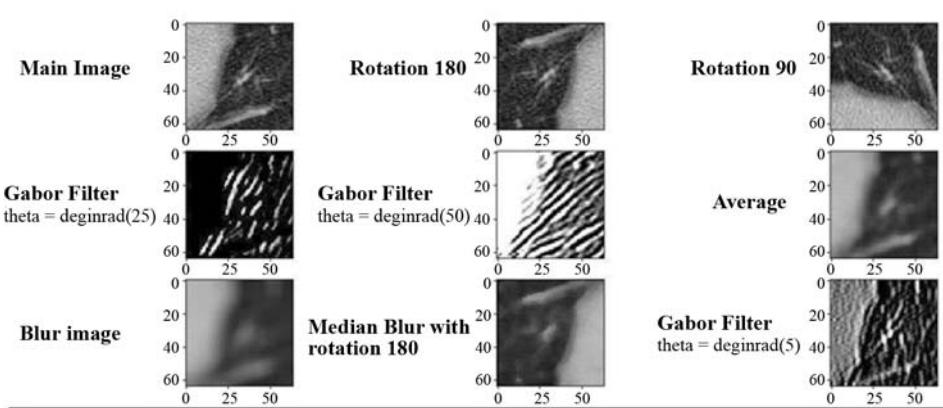


Fig. 1. Different filters on image

that the filter only responds to features revealing themselves along a horizontal axis. To obtain the features at different angles in the first case it was 25 degrees, in the second case 50 degrees, and in the third 5 degrees to produce different images and features for detection by the Gabor filter (fig. 1).

Another effective parameter in the Gabor filter, λ , is the length of sinusoidal wave in the above equation, which we consider equal to 5 for each of the three cases.

The next filter used to improve the dataset is the Average [19] filter. This low-pass filter makes the image smooth by getting the average of all pixels. This filter is used for noise reduction. To do this, use the 5×5 kernel to perform convolutional operations with the image. Eventually we produced new images using two blur filters with a kernel of 10 and a median blur image with s kernel of 5 along with the rotation of 180° . Fig. 1 depicted different filters of the images of the dataset. We produced 8 images of each image and 9 images with the original image counted in. We combined these distinct images and gathered them in an image. So our previous 64×64 images were converted into 192×192 images. Remarkable is the way in which these images are replaced. We tested different modes of various filters. We reached the best way to locate the images together. By changing the location of these images the precision changed. We have examined this problem in the experimental section. We helped to create

low-level features in order for the network to learn more easily and can distinguish between nodule and non-nodule images. The remarkable thing is that until this stage the number of images of the entire dataset had not been added. We would have the same number of images in the nodule and non-nodule class, but the nature of the collection has been changed and converted into new data with new features. After completing this section we will use data augmentation methods to increase the new dataset. It should be noted that this step is carried out inside the RasNet and will cause the creation of nodule and non-nodule images in the network. So the increase will be done on newly created images. Examples of new data collections are shown in fig. 2.

Experimental result

We performed extensive experiments to validate our algorithm. The dataset used in this work was collected from the largest publicly available reference database for lung nodules: the LIDC-IDRI [5]. We used LUNA16 annotation files to classify the images.

In the training section of the network we used 85 epochs with the Adam optimizer for each model and our number of batches equalled to 16.

Dataset. We also used the patient lung CT scan dataset with labeled nodules from the Lung Nodule Analysis 2016 (LUNA16) Challenge. The LUNA16 [20] database is selected from the publicly available LIDC/IDRI database, consists of 888 CT volumes, and was provided as MetaImage (.mhd) images that can be accessed and downloaded from the LUNA16 website [21].

The CT scans from the LUNA16 dataset consist of 512×512 slices. The annotation files have the Csv format and contain the image areas for nodule candidates. These annotation files are provided the diameter of nodules and corresponding x, y, z location of nodules and non-nodules in the scans. Nodules < 3 mm are omitted and only nodules > 3 mm are considered meaningful.

We read the raw scan by using the Simple Insight Segmentation and Registration Toolkit (SimpleITK)¹ [22] and converted it into an array. We read the annotation files in

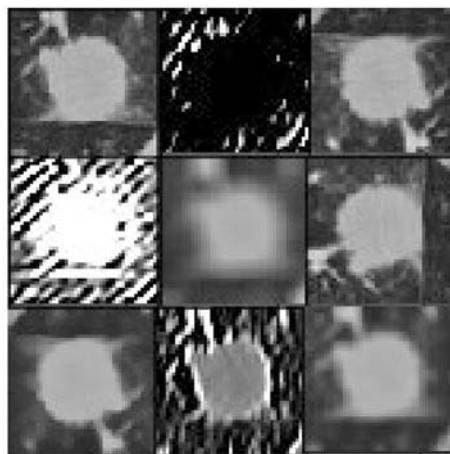


Fig. 2. Example of a new generated dataset

¹ Simple Insight Segmentation and Registration Toolkit.

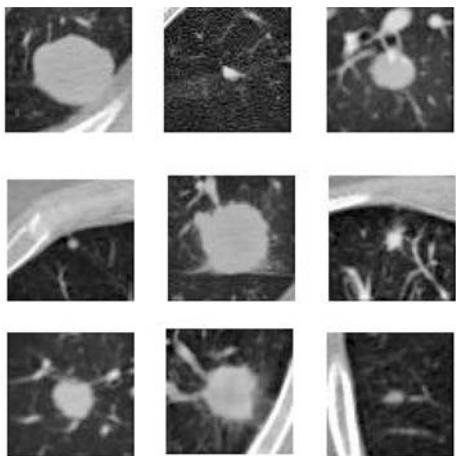


Fig. 3. Examples of the nodule class after preprocessing. Nodules are located in the center of each image

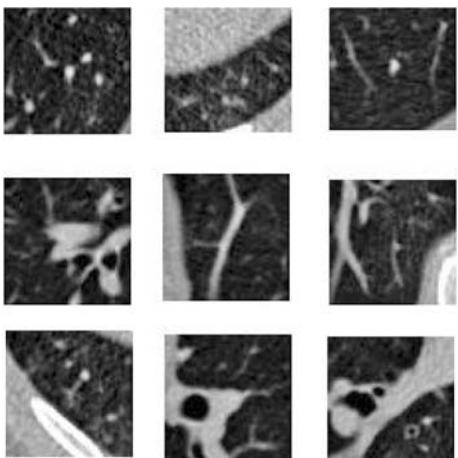


Fig. 4. Examples of the non-nodule class after preprocessing

the Csv format. Candidate locations in annotation files in orld coordinates were converted into noninteger voxel coordinates. The images were normalized and cropped around the candidate nodule to generate 64×64 Jpg files and we saved them in nodule and non-nodule folders. As we can see in figure 3, nodules are located in the centers of images. The diameter of nodules moves in the range of 3.253443196 mm to 32.27003025 mm. Examples of the class of non-nodule after preprocessing are depicted in fig. 4. We obtained 1182 nodules and 547680 non-nodules. Therefore we do not need to work with the whole image, only with pieces of the image which have the necessary information. Given that the ratio of the number of non-nodule images to the number of nodule images is 500 to 1, we decided not to use all the non-nodule images and only use 1182 images of both classes. 70 % of the data are used for training and 30 % for testing.

Image classification. Our goal in this paper is to identify and categorize images into two classes — nodule and non-nodule. We enhanced the dataset of images in order to better classify the images with higher accuracy. We measured the reliability of our datasets without preprocessing

on different convolutional neural networks such as Alex, GoogleNet, DenseNet, or ResNet.

The method of this is pretrained learning and no hyperparameters have changed. In the classification task the highest accuracy was related to the ResNet which is 88 %. Table 1 compares the accuracy of these networks. After the higher accuracy was achieved by ResNet we improved it by enhancing the dataset and manipulating some parameters.

Table 1
Comparison of different CNN models accuracy without preprocessing

Different Networks	Acc.
AlexNet	85.02
GoogleNet	86.03
DenseNet	88.20
ResNet	88.70

The newly generated dataset was trained by the implementation of ResNet and we classified the images into two classes — nodule and non-nodule. We used the Python application and the Keras high-level neural networks API written in Python and capable of running on top of TensorFlow for implementation. The pretrained ResNet was trained on CIFAR10. We fit it on its new dataset and looked for the optimal number of layers for our proposed system. Our batch size is 16 due to the limited amount of our resources. The used loss function is categorical cross entropy.

All experiments were run for 85 epochs at the learning rate of 0.0001 using Adam Optimization. The highest test accuracy at all the epochs was reported as the best score.

We experimented with 110 layers which had 54 two-layer Residual Units consisting of 3×3 convolutional layers. Our activation function was ReLU [14] and we used criteria such as precision e and recall to make it possible to compare our results with previous work. Their formulas are:

$$\text{Precision} = \frac{T_p}{T_p + F_p}, \quad \text{Recall} = \frac{T_p}{T_p + F_n}.$$

Table 2 shows the performance of the proposed system in two criteria of precision and recall. As we can see, the precision and recall in proposed system is acceptable.

Table 2
Precision and Recall in proposed model

Network	Precision, %	Recall, %
Proposed	94.52	98.10

Data Enhancement. Our main focus in this work is on improving the data in order to improve accuracy of classification. The main task was to prioritize various filters of the image. We chose 4 examples of states of filters and tested them on our system. In fig. 5 these images can be seen in a different arrangement.

In state 1 (fig. 5, a) the order of the filters and rotations is: 1 — the original image; 2— image with a median blur

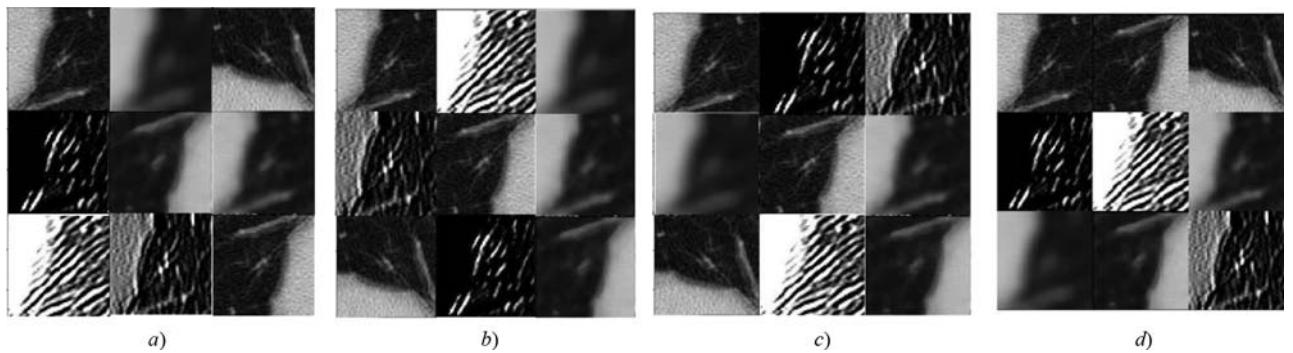


Fig. 5. Four states positioning images with filters and rotations

filter; 3 — the image rotated at 90°; 4 — image with the Gabor filter with theta = 25, 5 — and 6 — images with median blur and the rotation of 180° and average; 7 — images with the Gabor filter with theta = 5 and images with the rotation of 180°.

We examined state 2 of the images which are arranged as follows: 1 — the original image; 2 — image with the Gabor filter with theta = 50; 3 — image with a median blur filter; 4 and 5 — images with the Gabor filter with theta = 5 and 180° rotation; Images with an average filter, the rotation of 90°, the Gabor filter with theta = 25, and median blur with the rotation of 180° are located respectively.

As shown in fig. 5, c, in State 3 after the original image are images with the Gabor filter with theta = 25, Gabor filter with theta = 5, median blur, the rotation of 180°, average, the rotation of 90°, Gabor filter with theta = 50, and median blur with the rotation of 180°.

In State 4 (fig. 5, d) after the original image are located images with the rotation of 180°, the rotation of 90°, the Gabor filter with theta = 25, the Gabor filter with theta = 50, average, median blur, median blur with the rotation of 180° and the Gabor filter with theta = 5.

The experimental result shows that the arrangement of the images in state 4 is the best possible among these four orders. Because of the high accuracy that this arrangement requires in our work we introduced it as an optimal state. Table 3 shows the comparison of the 4 states.

Table 3

Comparison of different position of images with filters

States	Accuracy, %	Precision, %
1	89.90	88.14
2	90.69	90.90
3	92.74	90.88
Proposed	96.214	94.52

Discussion

In this section we discuss the advantages of our approach to other methods. Paper [11] named three networks: Alexnet, Google Net, and VGG as able to get good results

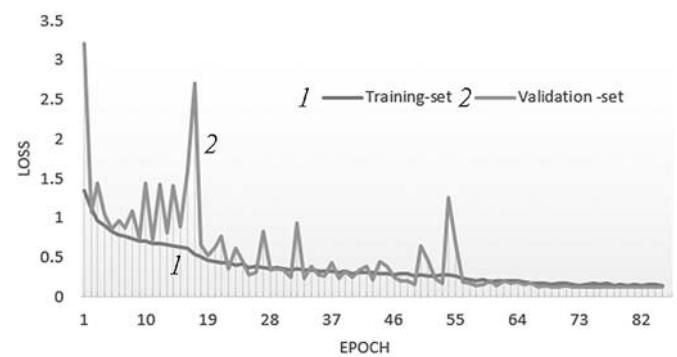


Fig. 6. Loss function of proposed method

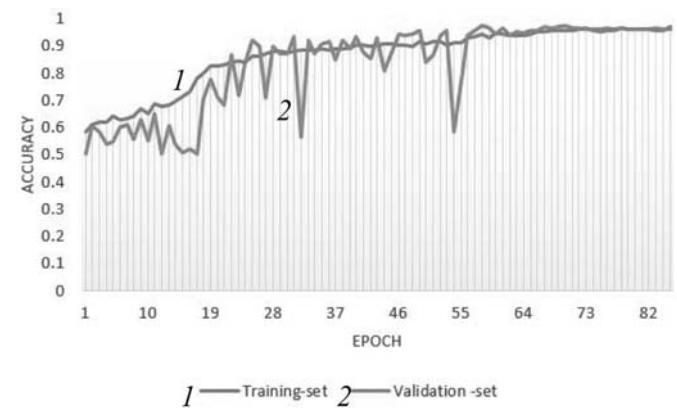


Fig. 7. Accuracy of proposed method

in the Pulmonary Nodule classification task. Their open source dataset is LUNA16, so we compared our results with their results.

According to our research the ResNet has the highest accuracy on the LUNA16 dataset without improving the images. We improved the image of the LUNA16 dataset to 96.336 %, which is acceptable with this volume of data. Fig. 6 shows the loss and fig. 7 indicates accuracy in test and train data. Table 4 shows the false positives and false negatives of the proposed system. Compared to paper [11] the value of our accuracy has improved. These results are visible in table 5.

Table 4
Comparison different CNN models

CNN model	Sens, %	PPV, %	Spec, %	NPV, %
AlexNet	80.8	82.1	97.0	97.0
GoogleNet	79.2	86.8	98.1	97.0
DenseNet	90.53	90.37	90.70	90.12
ResNet	98.10	94.52	94.32	98.03

Table 5

Confusion matrix of proposed method

		T	
		T'	F
Predicted	T'	311	18
	F'	16	299
Actual			

Conclusion

Early diagnosis of pulmonary nodules helps to diagnose the disease and prevent death by the disease. Diagnosis

For citation:

Esmaeilishahmirzadi N., Mortezapour H. Preprocessing for Enhancing the Classification of Pulmonary Data Sets using Convolutional Neural Networks, *Programmnaya Ingeneria*, 2018, vol. 9, no. 8, pp. 369–374.

DOI: 10.17587/prin.9.369-374

References

1. Siegel R., Ma J., Zou Z., Jemal A. Cancer statistics, *Cancer journal for clinicians*, 2014, vol. 64, no. 1, pp. 9–29.
2. Mahersia H., Zaroug M., Gabralla L. Lung cancer detection on CT scan images, a review on the analysis techniques, *International journal of advanced research in Artificial Intelligence*, 2015, vol. 4, no. 4, pp. 38–45.
3. Aerts H. J., Velazquez E. R., Leijenaar R. T., Parmar C., Grossmann P., Cavalho S., Bussink J., Monshouwer R., Haibe-Kains B., Rietveld D., Hoebers F. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach, *Nature Communications*, 2014, vol. 5, pp. 4006.
4. Khan W. Image Segmentation Techniques: A Survey, *Journal of Image and Graphics*, 2013, vol. 1, no. 4, pp. 166–170.
5. Armatto S. G., McLennan G., Bidaut L., McNitt-Gray M. F., Meyer C. R., Reeves A. P., Zhao B., Aberle D. R., Henschke C. I., Hoffman E. A., Kazerooni E. A., The lung image database consortium (LIDC) and image database resource initiative (IDRI), a completed reference database of lung nodules on CT scans, *Medical physics*, 2011, vol. 38, no. 2, pp. 915–931.
6. LeCun Y., Kavukcuoglu K., Farabet C. Convolutional networks and applications in vision, Circuits and Systems (ISCAS), *Proceedings of 2010 IEEE International Symposium on*, IEEE, 2010, vol. 2010, pp. 253–256.
7. Roth H. R., Lu L., Liu J., Yao J., Seff A., Cherry K., Kim L., Summers R. M. Improving computer-aided detection using convolutional neural networks and random view aggregation, *IEEE transactions on medical imaging*, 2016, vol. 35, no. 5, pp. 1170–1181.
8. Setio A. A. A., Ciompi F., Litjens G., Gerke P., Jacobs C., Van Riel S. J., Wille M. M. W., Naqibullah M., Sánchez C. I., van Ginneken B. Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks, *IEEE transactions on medical imaging*, 2016, vol. 35, no. 5, pp. 1160–1169.
9. Sun W., Wenqing Zheng B., Qian W. Computer aided lung cancer diagnosis with deep learning algorithms, *Medical Imaging, Computer-Aided Diagnosis*, 2016, vol. 9785, pp. 97850Z.
10. Anthimopoulos M., Christodoulidis S., Ebner L., Christe A., Mougiakakou S. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network, *IEEE transactions on medical imaging*, 2016, vol. 35, no. 5, pp. 1207–1216.
11. Ramaswamy S., Truong K. Pulmonary Nodule Classification with Convolutional, Neural Networks, *Stanford reports*, 2016, vol. 2, no. 5, pp. 2–9.
12. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
13. He K., Zhang X., Ren S., Sun J. Identity mappings in deep residual networks, *European Conference on Computer Vision*, Springer, Cham, 2016, pp. 630–645.
14. Nair V., Hinton G. E. Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
15. Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 2012, pp. 1097–1105.
16. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going deeper with convolutions, *Cvpr*, 2015, pp. 1–9.
17. Perez L., Wang J. The effectiveness of data augmentation in image classification using deep learning, Technical report, 2017, no. 300.
18. Fogel I., Sagi D. Gabor filters as texture discriminator, *Biological cybernetics*, 1989, vol. 61, no. 2, pp. 103–113.
19. Montenegro D., Gonzalez J. *Average filtering: theory, design and implementation*, Nova Science Publisher, 2016, 198 p.
20. Setio A. A.A., Traverso A., De Bel T., Berens M. S., van den Bogaard C., Cerello P., Chen H., Dou Q., Fantacci M. E., Geurts B., van der Gugten R. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge, *Medical image analysis*, 2017, vol. 42, pp. 1–13.
21. LUNA16 website, available at: <https://luna16.grand-challenge.org/download/>
22. Lowekamp B.C., Chen D. T., Ibáñez L., Blezek D. The design of SimpleITK, *Frontiers in Neuroinformatics*, 2013, vol. 7, pp. 45.

of pulmonary diseases by a physician requires a lot of time and attention. To reduce the time, increase the accuracy and efficiency, and accelerate the diagnosis computer diagnostic systems have been developed with the help of physicians to reduce the death rate by the disease.

Our challenge in classification of lung nodules is very sensitive because in medical images the number of nodules is lower than that of non-nodules. So the network tends to detect non-nodule images and is less likely to detect nodule images. Images are taken from the LUNA16 dataset, which is derived from the LIDC-IDRI database.

Our work in this paper consists of two parts: enhancement of the dataset images and optimal classification of images into nodule and non-nodule.

Our system has used a data augmentation method and image processing filters to enhance the datasets. The newly generated dataset includes new images of which every image contains 9 images and features. These new images have been trained on the different convolutional neural networks and experimental results show that ResNet has performed better in our dataset.

The proposed method compares to other methods with an accuracy of 96.214 and has shown acceptable performance.

М. Г. Гриф, д-р техн. наук, проф., e-mail: grifmg@mail.ru, **А. В. Лукоянычев**, аспирант, e-mail: dizzystyle@yandex.com, Новосибирский государственный технический университет

Программный комплекс для обучения русскому жестовому языку на основе Unity3D

Представлены результаты разработки программного комплекса для обучения русскому жестовому языку. Рассмотрены особенности использования нотационной записи для управления компьютерным анимационным персонажем, демонстрирующим дактилемы и слова русского жестового языка. Описаны возможности программного комплекса для создания справочника жестового языка и уроков, перспективы его дальнейшего использования.

Ключевые слова: русский жестовый язык, мультимедийный словарь, нотации Димскис, Unity3D, анимированный персонаж-аватар

Введение

В Российской Федерации реализуется целевая программа "Доступная среда", которая предусматривает создание и внедрение в практику различного рода средств социальной адаптации инвалидов по слуху [1]. Основой адаптации слабослышащих людей является обучение их жестовому языку (ЖЯ). Внедрение современных компьютерных технологий позволяет автоматизировать этот процесс. Одним из наиболее эффективных средств человека-машинного взаимодействия и обучения ЖЯ являются мультимедийные компьютерные программы, поэтому создание информационных приложений, способных работать с ЖЯ, является одной из приоритетных задач при работе с глухими и слабослышащими людьми. Возможным вариантом компьютерного синтеза ЖЯ является использование компьютерного 3D-персонажа (аватара).

Для управления анимационным персонажем необходимо определенным образом представить ЖЯ, чтобы компьютер мог обрабатывать и синтезировать жесты. Для описания жеста по его признакам существует несколько различных систем нотаций (жестовой транскрипции), позволяющих зафиксировать представление жеста в записи, которые обладают разной степенью подробности и иконичности и ориентированы на описание различных явлений в жестовых языках [2]. Немногочисленные работы в России, которые посвящены компьютерному сурдопереводу, в основном используют нотационную запись HamNoSys [3, 4]. Предлагаемые нотации имеют свои достоинства и недостатки [2]. Идеальной общепринятой нотации ЖЯ нет. По этой причине ведутся работы по применению всех рассмотренных нотаций.

Для русского жестового языка (РЖЯ) перспективной является нотация, разработанная Л. Димскис [5]. В этой нотации выделяется более 50 конфигураций

рук, около 50 характеристик места исполнения жеста и более 100 характеристик локализации.

Нотационная запись Димскис обладает следующими свойствами, необходимыми для реализации автоматизированного компьютерного сурдоперевода:

- "фотографические" символы конфигурации пальцев;
- ограниченное число знаков в нотации;
- знаки нотации Димскис являются основными транскрипциями РЖЯ;
- возможность использования анимированного персонажа для отображения РЖЯ;
- символьная запись, которая значительно уменьшает объем словаря и повышает скорость воспроизведения на мобильных устройствах.

Для реализации 3D-модели сурдопереводчика был выбран мощный графический движок Unity3D, основным преимуществом которого является кросс-платформенность и доступность [6]. Этот движок имеет возможность сборки приложения под несколько популярных операционных систем как на мобильных устройствах, так и на персональных компьютерах.

Управление анимационным персонажем

Ответственным этапом при создании аватара является обоснованное проведение риггинга — процесса подготовки персонажа к анимации. Такой процесс включает создание и размещение внутри рига (трехмерной модели) виртуального скелета, установки иерархической зависимости между суставами и возможной их трансформации. Скелетная анимация позволяет управлять большим числом подвижных элементов в фигуре человека с помощью небольшого числа управляющих элементов. От сложности этой структуры напрямую зависит гибкость и реалистичность получаемой анимации. Однако большое чис-

ло подвижных элементов в структуре рига приводит к усложнению управления им. Другим важным этапом при создании аватара является скрининг, процесс назначения связей между участками поверхности анимируемой фигуры и элементами рига. На этом этапе определяются группы взаимосвязанных элементов, подвижные части которых должны подвергаться воздействию трансформации при изменении положения отдельной кости скелета. Оптимальным вариантом для анимации аватара является скелет, в упрощенной форме соответствующий обычному человеческому скелету.

Основными подвижными элементами разрабатываемого аватара являются руки, остальные части тела остаются в статическом состоянии. Структура виртуальных костей должна в наибольшей степени соответствовать структуре костей реальной руки.

Анатомическое строение руки достаточно простое (рис. 1, а). Рука прикрепляется к туловищу посредством костей плечевого пояса, суставов и мышц. Она состоит из трех частей — плеча, предплечья и кисти. Сгибание в локте дает рукам большую подвижность, увеличивая их амплитуду и функциональные возможности. Кисть состоит из множества подвижных суставов, благодаря которым человек может показывать пальцем в нужном направлении.

Плечо и кисть соединяются посредством плечевой кости, локтевой и лучевой костей. Все три кости между собой соединяются с помощью суставов. В локтевом суставе руку можно сгибать, разгибать и поворачивать. Обе кости предплечья соединяются подвижно, поэтому во время движения в суставах лучевая кость вращается вокруг локтевой кости. Кисть можно повернуть на 180°.

Кисть состоит из ладони, пяти пальцев и пястной кости (рис. 1, б). Мизинец, безымянный, средний и указательный пальцы имеют одинаковую структуру (три фаланги), в отличие от большого пальца (две фаланги). Первая пястная кость участвует в движении большого пальца, следовательно, можно рассматривать большой палец как состоящий из трех суставов.

В соответствии с анатомическим строением руки человека, если четыре пястные кости можно рассматривать как единое целое, то кисть может быть описана движением, состоящим из 16 частей, а остальные части руки можно описать тремя подвижными элементами, соответствующими трем суставам: локтевому, предплечью и плечевому.

Аватар создан в пакете Blender3D. В библиотеке пакета есть готовые персонажи, однако даже они требуют адаптации для решения поставленной задачи.

Состояние движения руки можно контролировать путем управления 19 подвижными элементами для одной руки. В таблице приведено соответствие между частями руки человека и подвижными элементами аватара.

Наличие такого числа подвижных элементов позволяет качественно описать движение руки. Каждый элемент руки имеет пространственное описание.

Существуют два основных типа планирования движения созданных подвижных элементов — это прямая кинематика и инверсная кинематика. При использовании прямой кинематики передвижение вышестоящего элемента иерархии вызывает перемещение всех нижестоящих элементов. При использовании обратной кинематики алгоритм получается ровно противоположный.

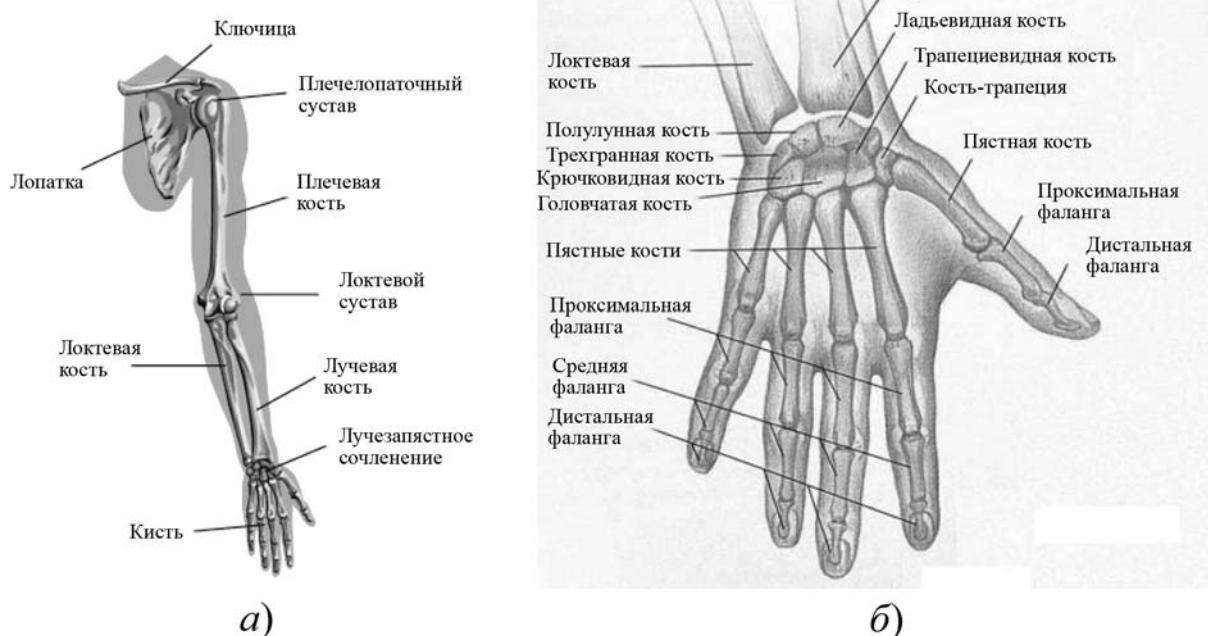


Рис. 1. Анатомическое строение руки (а) и кисти (б)

Соответствие частей руки и подвижных элементов аватара

Часть руки человека	Правая рука	Левая рука
Мизинец	RightHandPinky1 RightHandPinky2 RightHandPinky3	LeftHandPinky1 LeftHandPinky2 LeftHandPinky3
Безымянный палец	RightHandRing1 RightHandRing2 RightHandRing3	LeftHandRing1 LeftHandRing2 LeftHandRing3
Средний палец	RightHandMiddle1 RightHandMiddle2 RightHandMiddle3	LeftHandMiddle1 LeftHandMiddle2 LeftHandMiddle3
Указательный палец	RightHandIndex1 RightHandIndex2 RightHandIndex3	LeftHandIndex1 LeftHandIndex2 LeftHandIndex3
Большой палец	RightHandThumb1 RightHandThumb2 RightHandThumb3	LeftHandThumb1 LeftHandThumb2 LeftHandThumb3
Ладонь (кисть руки)	RightHand	LeftHand
Рука (локтевой сустав)	RightArm	LeftArm
Предплечье	RightForeArm	LeftForeArm
Плечо	RightShoulder	LeftShoulder

Суть прямой кинематики состоит в том, что воздействие передается по иерархической цепочке сверху вниз, т. е. дочерние сегменты движутся относительно родительских сегментов. Сначала положение и/или ориентацию меняет родительский сегмент. Это изменение влияет на положения и/или ориентацию всех остальных дочерних сегментов. Далее изменяется положение следующего сегмента в цепочке, при этом изменяется положение всех последующих дочерних ему сегментов, а родительские сегменты остаются неподвижными. Это означает, что при повороте плечевого сустава все другие суставы руки, такие как локоть, запястье и пальцы, меняют свое положение.

Глобальное положение любого подвижного элемента, имеющего родителей, рассчитывается при прямой кинематике в соответствии с уравнением [7]

$$P_i = P_{i-1} + \text{rotate}\left(D_i, P_{i-1}, \sum_{k=0}^{i-1} \alpha_k\right),$$

где P_i — новое положение i -го сустава (конечной точки); P_{i-1} — положение $(i-1)$ -го сустава, rotate — поворот i -го сустава относительно $(i-1)$ -го на суммарный угол поворота всех предшествующих суставов; D_i — длина i -й кости; k — число суставов, участвующих в движении; α_k — угол поворота k -го элемента (сустава).

В Unity3D имеются функции для автоматического расчета траектории движения каждого элемента. При этом генерируются кривые движения/вращения/масштаба для каждого элемента, участвующего

в анимации. В качестве опорных значений Unity3D использует узловые точки и описание траектории. Этими параметрами регулируется плавность движения и объем занимаемой памяти. Кроме указанных параметров, Unity3D может регулировать скорость отображения, которая определяется реперными метками, установленными разработчиком на оси времени таймера анимации. Таким образом, создание анимации движения руки аватаром-человеком заключается в поиске оптимального решения, т. е. является компромиссом между плавностью и естественностью отображения, с одной стороны, и объемом занимаемой памяти и загрузкой процессора, с другой стороны.

Создание качественной анимации — процесс, требующий учитывать все факторы влияния отдельных элементов друг на друга. Следует иметь в виду анатомические особенности человека в реальном мире и соответствующую взаимосвязь элементов.

Движок Unity3D позволяет разработчику самостоятельно рассчитывать траекторию движения и при этом регулировать параллельность передвижения составных частей руки. Относительно небольшое число подвижных элементов у аватара позволило заменить стандартный модуль Unity3D на модуль с регулируемым более мелким шагом. Это дает возможность реализовать естественное и плавное передвижение суставов руки [8].

Еще одной проблемой при создании анимации жеста является соблюдение выбранного стиля движений. Для этого в редакторе анимации необходимо предусмотреть возможность копирования ранее созданных элементов анимации и их простой адаптации для реализации новых жестов.

Все нотационные записи у Димскис разбиты на пять групп:

- 1) положение пальцев рук (56 символов);
- 2) направление ладони (20 символов);
- 3) направление пальцев (13 символов);
- 4) место исполнения жеста (54 символа);
- 5) направление и характер движения исполняемого жеста (105 символов).

Большие группы для удобства работы разбиты на несколько более мелких, связанных тематически.

Движения выполняются правой, левой или одновременно двумя руками. Для этого существуют символы П, Л, ЛП. Если действия в двухручном жесте выполняются не одинаковые, то сначала описывается пассивная рука, а затем активная. Для синхронного одновременного движения двух рук используются отдельные анимационные файлы и соответствующие им символы. Таким образом, используются три разновидности файлов: для левой руки, для правой руки и одновременно для двух рук.

В нотации Димскис все слова и буквы в основном начинаются с пятисимвольного описания положения руки и пальцев. Нотация состоит из следующих символов:

- первый элемент — характер исполнения жеста (одноручный, двуручный);



Рис. 2. Пример записи жеста — слово "суббота" в нотации Димскис

- второй элемент — положение пальцев;
- третий элемент — направление ладони;
- четвертый элемент — направление пальцев;
- пятый элемент — место расположения жеста.

Для примера рассмотрим запись в нотации Димскис слова "суббота" (рис. 2):

- первый знак (буква П) указывает на то, что жест выполняется правой рукой;
- второй знак иконографически указывает форму руки — большой и мизинец прямые, остальные прижаты к ладони;
- третий символ схематически изображает разворот ладони — она направлена от себя, от корпуса говорящего;
- четвертый знак указывает на ориентацию ладони — пальцами вверх;
- пятый символ означает, что жест исполняется на уровне груди перед собой.

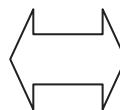
После пятого символа идут дополнительные символы, которые отображают динамику выполнения жеста.

Нотация Димскис не является полностью компьютерно-ориентированной. Некоторые нотации даются в сокращенном виде, без указания места выполнения, что при реализации недопустимо. Поэтому нотации дополняют необходимыми знаками для полного описания слова.

Для удобства работы с Unity3D разработано программное обеспечение, которое позволяет переводить созданный в Unity3D анимационный клип в скриптовый текстовый формат и наоборот — текстовый формат такого описания в анимационное движение. Создание скриптового языка позволило упростить понимание отображения движений аватара, опера-

```
{
  "Description": "",
  "PlayMode": 0,
  "curves": [
    {"TargetName": "RightHandMiddle1",
     "modelRotationNodes": [
       {"angle": {
         "x": 0.000010031,
         "y": 7.340751171,
         "z": 7.340751171 },
        "time": 0.3 } ]},
  ...
}
```

a)



тивно корректировать файл и сократило объем файлов клипов. Это особенно актуально для мобильной реализации. Пример данного скриптового файла и файла в формате Unity3D приведен на рис. 3.

В рамках одного анимационного файла параллельно осуществляется движение подвижного элемента по трем координатам (рис. 4, б). В одном файле может проводиться движение нескольких элементов за различное время (с разной скоростью), например пальцы (рис. 4, в), кисть и рука.

Возможность Unity3D параллельно выполнять несколько потоков анимации является наиболее существенной при создании конфигурации пальцев, где подвижных элементов и степеней свободы много, а изменение времени выполнения позволяет создать наиболее естественное выполнение движения.

Использование отдельных нескольких записей движения одного элемента в одном анимационном файле позволяет, во-первых, создавать сложные движения (например, дактилем букв Д, З, Й, Ъ, где требуется демонстрация движения руки по определенной траектории), во-вторых, за счет изменения времени выполнения *t* изменять скорость демонстрации жеста (рис. 5). Это достигается изменением параметра *time* на одинаковое значение, для равномерного движения, или на различное, для замедления и ускорения движения.

Для реализации движения некоторых элементов требуется изменение только одной или двух координат, например, поворот кисти, подъем, опускание кисти. Такое изменение достигается за счет введения специальной координаты 0.0. Действие значения 0.0 показано на примерах "ладонь к себе" и "вверх" (рис. 6).

Кроме специального символа 0.0 для отмены перемещения подвижного элемента используется "фиктивное" движение, т. е. конечные координаты совпадают с координатами начального положения. При демонстрации отдельных дактилем и слов некоторые элементы (суставы) не двигаются, например,

```
path;type;property;time;value;inTangent;
inTangent;outTangent
ValveBiped_Bip01_Pelvis/ValveBiped_Bip01_Spine/ValveBiped_Bip01_Spine1/ValveBiped_Bip01_Spine2/ValveBiped_Bip01_Spine4/ValveBiped_Bip01_L_Clavicle/ValveBiped_Bip01_L_UpperArm/ValveBiped_Bip01_L_Forearm/ValveBiped_Bip01_L_Hand/ValveBiped_Bip01_L_Finger1/ValveBiped_Bip01_L_Finger11;UnityEngine.Transform:m_LocalRotation.w;0.3; 0.000010031; 7.340751171; 7.340751171; 0;
```

б)

Рис. 3. Пример скриптового (а) и анимационного (б) файлов движения аватара

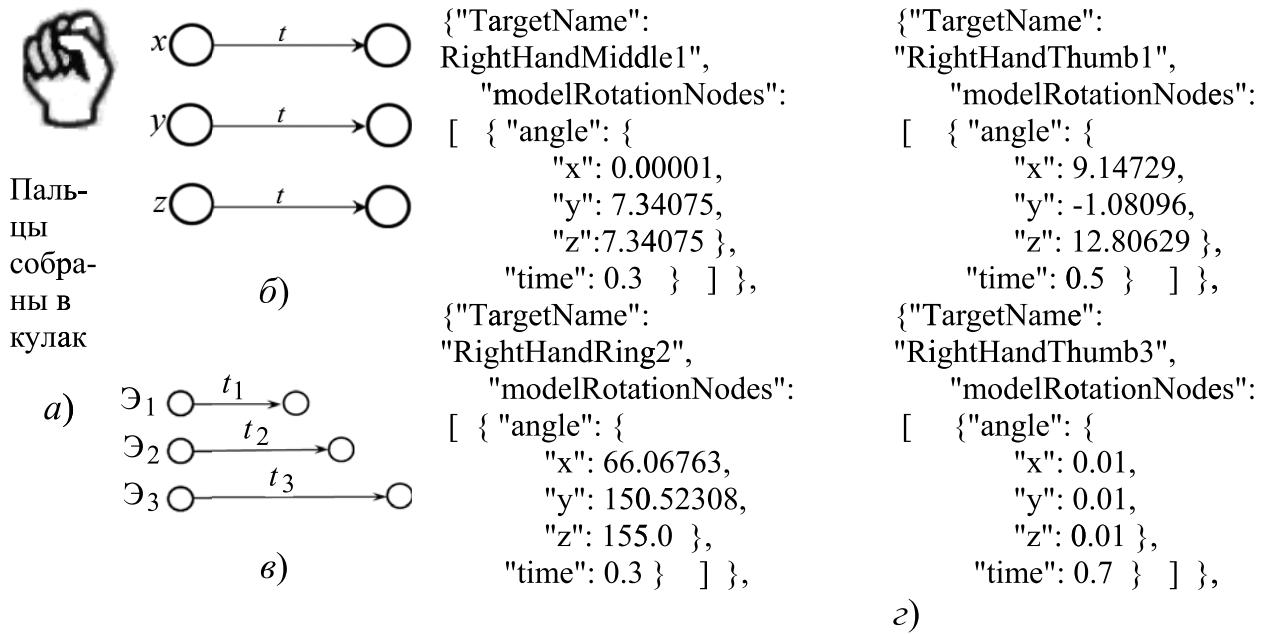


Рис. 4. Пример реализации положения пальцев руки:

а — знак конфигурации пальцев; б — диаграмма одновременного перемещения по координатам; в — диаграмма движения элементов с разной скоростью, где кружок означает абсолютное значение; г — фрагмент анимационного файла; t_i — время выполнения i -го действия; Θ_i — i -й подвижный элемент

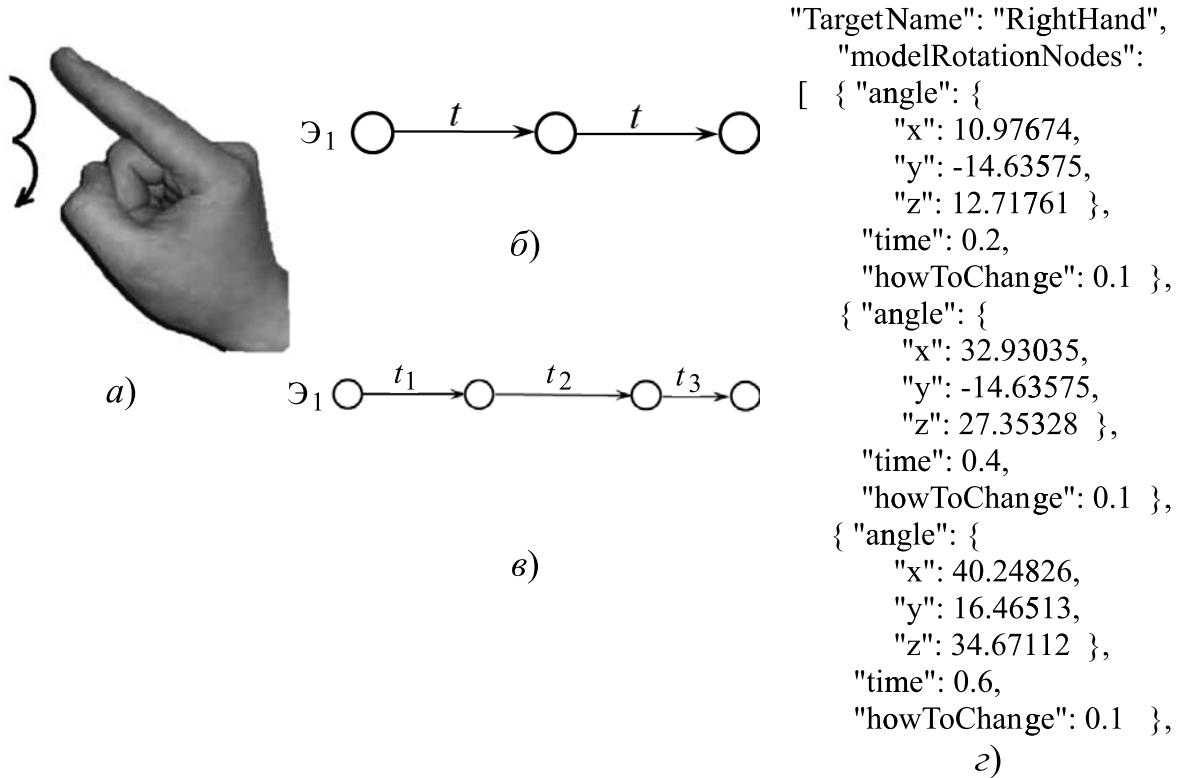


Рис. 5. Пример сложного движения:

а — демонстрация буквы З; б — диаграмма равномерного движения; в — диаграмма неравномерного передвижения; г — фрагмент анимационного файла; t_i — время выполнения i -го действия; Θ_i — i -й подвижный элемент

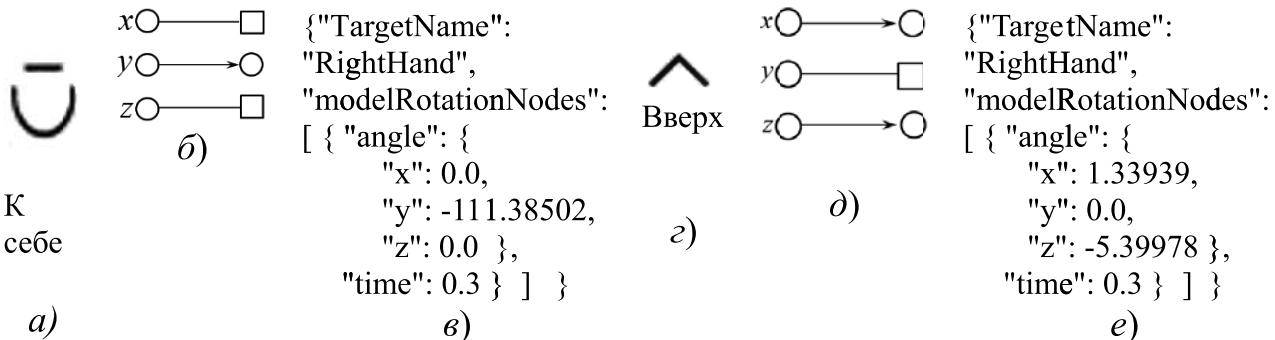


Рис. 6. Пример управления движением по отдельным координатам:

a, г — пиктограммы нотации; б, д — диаграммы выполнения; в, е — фрагменты анимационных файлов; кружок — абсолютное новое значение; квадрат — предыдущее абсолютное значение

предплечье и плечо. Однако при составлении предложений, словосочетаний и слов из дактилем неиспользуемые элементы необходимо вернуть в исходное состояние. В анимационных файлах для корректного перехода от одной демонстрации к другой требуется прописывать "фиктивное" движение. Например, после демонстрации дактилем Л необходимо показать дактилем А, сустав ключица (плечо) должен вернуться в исходное состояние, где он находится для большинства букв.

Таким образом, в те слова, после которых необходим возврат в исходное (начальное) состояние, не обязательно прописывать фиктивные движения, а в те слова, которые могут входить в словосочетания, требуется вносить все фиктивные движения суставов.

Для реализации знаков нотации "Направление и характер движения исполняемого жеста" предусмотрен относительный принцип изменения координат. Предыдущее положение подвижного элемента берет-

ся за начальное, а в анимационном файле движения новые координаты записываются как приращения. Учет данной особенности осуществляется введением управляющего слова в описании перемещения подвижного элемента *"howToChange"*: 1. Например, при реализации движения одновременно с переконфигурацией пальцев анимационные файлы могут выполняться параллельно с использованием относительных и абсолютных координат (рис. 7).

Возможно использование в одном анимационном файле как относительного движения, так и абсолютного. Это позволяет некоторые подвижные элементы устанавливать в конкретное положение (наиболее часто — в исходное), что актуально для создания словосочетаний.

В одном файле для каждого подвижного элемента создается свой поток движения. Это свидетельствует о том, что выполнение действий в нескольких последовательных файлах может накладываться и создавать эффект непрерывности демонстрации жестов.

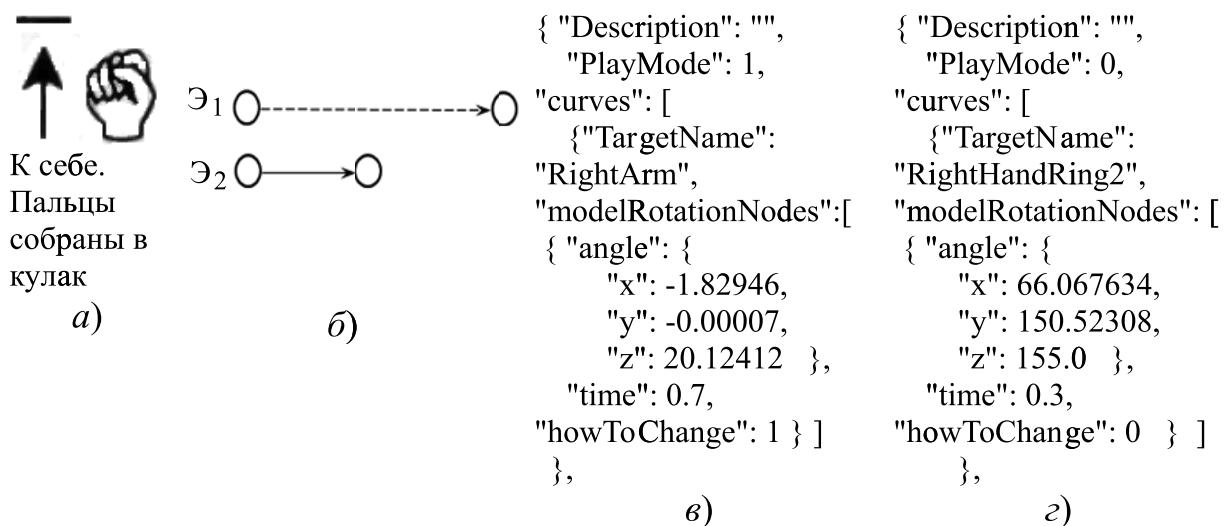


Рис. 7. Пример относительного движения для слова "взять":

а — фрагмент нотации Димскис; б — диаграмма выполнения, где штриховая линия — относительное вычисление координат; в, г — фрагменты анимационных файлов; \mathcal{E}_i — i -й подвижный элемент

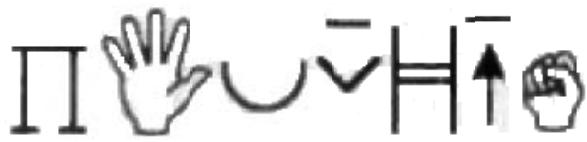


Рис. 8. Нотация Димскис слова "взять"

Если необходимо, чтобы очередной анимационный файл начинался только по окончании выполнения предыдущих действий, что соответствует фиксации положения руки, то используется управляющее слово в заголовке файла — "PlayMode": 1. Так, в примере на рис. 5, в, если установить "PlayMode": 0, то выполнение демонстрации слова "взять" будет происходить непрерывно, без фиксации первого положения руки (рис. 8).

Для организации паузы определенной длительности (типовая длительность — 1 с) используется служебный нотационный символ КС (конец синхронизации). Данный символ необходим при реализации слов, которые в соответствии с нотацией Димскис представляются как последовательность конфигурации пальцев. Например, слова: апрель, май, июнь, июль (рис. 9). Это связано с тем, что нотация Димскис ориентирована на демонстрацию человеком и не достаточно учитывает особенности компьютерной реализации.

Для демонстрации слов и дактилем в словаре предусмотрен служебный символ "возврат в исходное состояние", которым заканчиваются все анимации.

Некоторым знакам нотации соответствует несколько анимационных файлов. Это обусловлено тем обстоятельством, что необходимое действие можно выполнить по-разному. Например, символ нотации "Место исполнения жеста: с касанием правой щеки" можно выполнить с отведением локтя или нет. Поэтому в каждом конкретном случае предлагается выбрать наиболее приемлемое решение.

При составлении нотационной записи следует ориентироваться на наглядность представления слова, так как многие сурдопереводчики в зависимости от обстановки и пристрастий демонстрируют слова в разных местах. Кроме того, для некоторых слов существуют разные варианты жестикуляции, что отражается и у Димскис, не говоря уже о разных диалектах и школах.

Для некоторых букв, например, Л, М, П, Т, следует ввести соответствующие им нотационные сим-

- 1) Правая рука;
- 2) ладонь раскрыта;
- 3) вверх;
- 4) от себя;
- 5) на уровне груди перед собой;
- 6) к себе;
- 7) пальцы собраны в кулак

Рис. 8. Нотация Димскис слова "взять"

волы, что облегчает понимание и улучшает наглядность. При этом можно создать анимационный файл, который реализует более естественное движение рук.

Реализация программного комплекса

Для реализации описанной концепции разработан программный комплекс, который имеет два режима: "Разработчик" и "Пользователь" (рис. 10).

В режиме "Разработчик" технологический инструментарий включает все необходимые модули для создания и демонстрации жестов, для привязки разработанных клипов к нотации Димскис, для создания полноценного мультимедийного справочника по РЖЯ и разработки уроков по обучению РЖЯ.

Основным модулем для разработки анимационных файлов (клипов) является "Редактор клипов". При запуске программы загружается файл описания сцены с аватаром, находящимся в исходном положении (рис. 11, а, см. третью сторону обложки). В редакторе доступны для отображения все созданные анимационные файлы. Разработчик имеет возможность создавать сценарии, используя всю иерархию библиотеки от простейших анимаций до составных. При этом возможно увеличение аватара для точного установления координат отдельных элементов (рис. 11, б, см. третью сторону обложки), что требуется при создании формы руки. В данном режиме осуществляется регулирование скорости выполнения клипа, его просмотр и редактирование.

В редакторе клипов можно загрузить ранее созданный файл, создать новый и сохранить созданный под любым именем. Данные действия реализуются с помощью элементов верхнего меню. Основное меню предусматривает шесть основных действий по созданию клипа. К загруженному файлу можно добавить новые данные, которые созданы в режиме редактирования. При создании новых движений аватара можно загрузить последовательно несколько ранее созданных файлов, проиграть их (установить

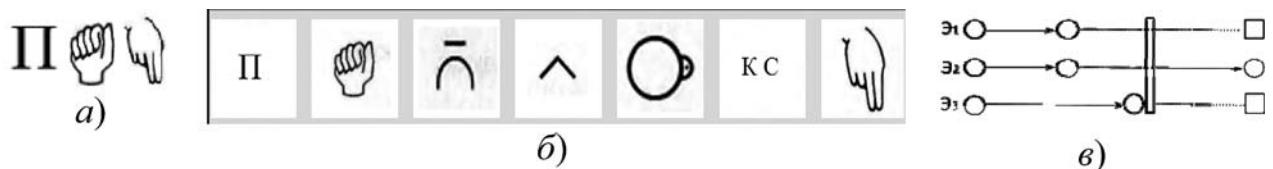


Рис. 9. Нотационные записи и диаграмма слова "апрель":

а — нотация Димскис; б — нотация для компьютерной реализации; в — диаграмма реализации; полоска — пауза;
Э_i — i-й подвижный элемент

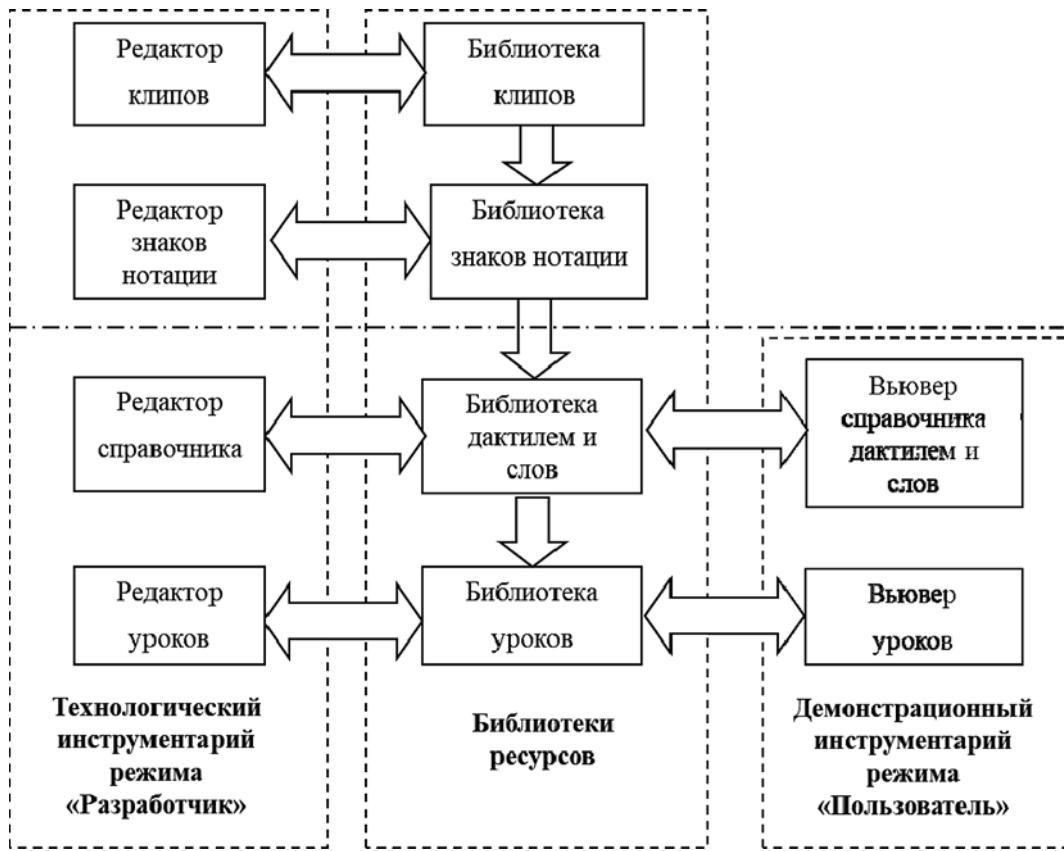


Рис. 10. Иерархическая структура программного комплекса

руки аватара в нужное положение) и, очистив содержимое файла, начать создавать новые перемещения руки. Данный режим используется при реализации анимаций из группы "Направление и характер движения исполняемого жеста". При редактировании движений предусмотрен откат на шаг назад и возврат в начальное состояние. Возможно внесение предварительной задержки начала выполнения движения и добавление интервала времени на выполнение последовательных движений. При создании анимационного файла необходимо указать способ расчета передвижения: в абсолютных (*Absolute*) или относительных координатах (*Relative*). Кроме того, следует указать, ожидать завершения действий в предыдущем клипе (*Sync*) или нет (*Normal*). В окне редактора отображается создаваемый анимационный файл. В правой части экрана редактора отображается аватар, который можно приблизить или отдалить с помощью манипулятора мыши, включив режим "+", а затем зафиксировать выбранный масштаб.

Для управления движениями анимационного персонажа предусмотрено меню, расположенное под аватаром. Элементы меню: воспроизведение, пауза, шаг назад, изменение времени воспроизведения, возврат в нулевое состояние, активация суставов, возврат в начальное состояние. Активация суставов выделяет у аватара все управляемые подвижные элементы. При выделении необходимого в данный

момент сустава на экран выводятся его координаты. Изменение координат с помощью ползункового регулятора перемещает выбранный подвижный элемент. Разработанный набор возможностей создания клипа и управления аватаром позволяет создавать все необходимые перемещения подвижных элементов для естественного отображения движения рук анимационного персонажа. Созданный анимационный файл записывается в библиотеку клипов.

"Редактор знаков нотации" позволяет связать созданный клип с символом нотации Димскис, создать словесное описание выбранного анимированного символа и указать его расположение в библиотеке знаков нотации (рис. 12, а, см. третью сторону обложки). Для удобства работы с большими группами символов ("Место исполнения жеста (54 символа)" и "Направление и характер движения исполняемого жеста (105 символов)") они разбиты на несколько контекстно-зависимых групп.

На основе полученных анимированных знаков нотации в "Редакторе справочника" проводится разработка мультимедийного словаря РЖЯ за счет сборки составных клипов. Сначала выбирается символ нотации, соответствующий руке (одновременно двум), которой выполняется жест — П, Л, ЛП. В зависимости от этого выбора выводятся контекстно-доступные символы нотации. Создается сценарий в соответствии с последовательностью нотации Димскис,

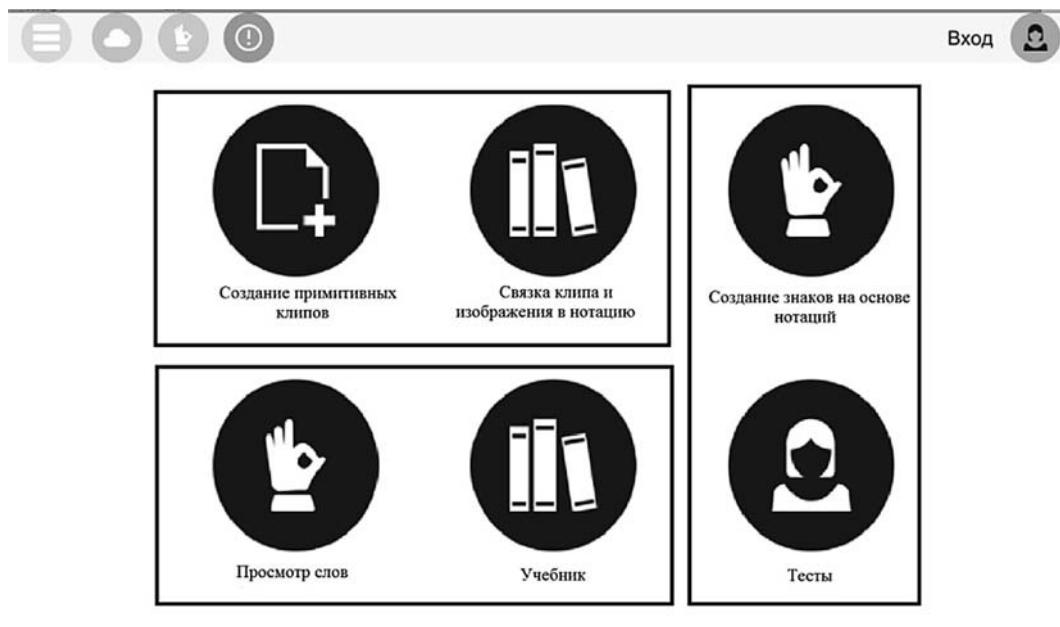


Рис. 14. Внешний вид программного комплекса

который можно оперативно проверить. По окончании разработки дактилемы или слова осуществляется описание этого жеста и запись в справочник (рис. 12, б, см. третью сторону обложки).

На основе библиотеки дактилем и слов (рис. 13, см. третью сторону обложки) создаются обучающие курсы в "Редакторе уроков". Для разработки сценариев обучения РЖЯ может использоваться стандартный текстовый редактор или редактор, встроенный в программный комплекс, который позволяет оперативно проверить и настроить урок. В отличие от справочника, где после демонстрации аватар возвращается в исходное положение, в уроках можно составлять сложные словосочетания, а для возврата в начальное состояние необходимо применять служебный символ.

Словарь РЖЯ составляет около 3000 слов, поэтому редакторы справочника и уроков в режиме "Разработчик" могут использоваться сторонними специалистами в области жестового языка. Для этой цели программный комплекс доступен в сети Интернет. Такой подход позволяет привлечь заинтересованных лиц к наполнению словаря и уроков. Также для изучения РЖЯ в режиме "Пользователь" возможна загрузка на персональные компьютеры и мобильные устройства модулей просмотра библиотеки дактилем и слов, а также созданных уроков. На рис. 14 показаны доступные модули программного комплекса.

Первые два модуля "Создание примитивного клипа" и "Связка клипа и изображения в нотацию" доступны в режиме "Разработчик" только на локальном компьютере. Два следующих модуля — "Создание знаков на основе нотаций" и "Тесты" (создание уроков) — доступны сторонним разработчикам после авторизации (регистрации) на сайте. Модули "Просмотр слов" и "Учебник" можно скачать на пользовательское устройство или использовать через сеть Интернет.

Заключение

Разработанный программный комплекс является достаточно наглядным и простым в использовании, что позволяет привлечь знакомое с РЖЯ заинтересованное сообщество к пополнению справочника через сеть Интернет и к составлению обучающих уроков. Кроме того, комплекс легко пополняется новыми символами, что дает возможность совершенствовать как саму систему, так и нотации Димскис. Он ориентирован для использования как в сети Интернет, так и на мобильных устройствах, и на персональных компьютерах.

Список литературы

1. Государственная программа Российской Федерации "Доступная среда" на 2011–2020 годы. URL: <http://www.rosmiintrud.ru/docs/government/170> (дата обращения 14.05.2018).
2. Гриф М. Г., Лукоянычев А. В. 3D-анимация русского жестового языка на основе нотации Димскис // Программная инженерия. 2017. Т. 8, № 7. С. 310–318.
3. Карпов А. А., Кагиров И. А. Формализация лексико-на системы компьютерного синтеза языка жестов // Труды СПИИРАН. 2011. Вып. 1 (16). С. 123–140.
4. Мануева Ю. С., Гриф М. Г., Козлов А. Н. Построение системы компьютерного сурдоперевода русского языка // Труды СПИИРАН. 2014. Вып. 6 (37). С. 170–187.
5. Димскис Л. С. Изучаем жестовый язык: учебное пособие. М.: Академия, 2002. 128 с.
6. Smith M., Queiroz C. Unity5.x Cookbook. Packt Publishing, 2015. 570 p.
7. Zucconi A. An Introduction to Procedural Animations. URL: <https://www.alanzucconi.com/2017/04/17/procedural-animations/> (дата обращения 14.05.2018).
8. Гриф М. Г., Лукоянычев А. В. Мультимедийный словарь русского жестового языка на основе нотации Димскис // Вестник АГТУ, Сер. Управление, вычислительная техника и информатика. 2017. № 1. С. 105–114.

The Program Complex for Training Russian Sign Language Based on Unity3D

M. G. Grif, grifmg@mail.ru, **A. V. Lukoyanychev**, dizzystyle@yandex.com, Novosibirsk State technical University, Novosibirsk, 630073, Russian Federation

Corresponding author:

Grif Michael G., Doctor of Department "Automated Control System", Novosibirsk State Technical University, Novosibirsk, 630073, Russian Federation
E-mail: dizzystyle@yandex.com

Received on June 26, 2018

Accepted on July 16, 2018

The article presents the results of the development of a software package for teaching sign language. Computer systems for training sign languages are built on the basis of notation records. The perspective grounds of Dimskis notation use for the computer implementation of Russian sign language are presented. The peculiarities of the notation record use for the control of a computer animated character that demonstrates dactilemas and words of sign language are considered. The graphics package Unity3D is used to implement the 3D model of the animated character (avatar). Dimskis notation is not completely computer-oriented, so the notation records are supplemented with required signs for the full descriptions of the word. For the convenience of working with Unity, the software complex has been developed that allows one to translate an animation clip created in Unity3D into a script text format and vice versa — the text format of a description in the animation movement. This approach allows us to increase the visibility of animation files and simplify the management of the avatar. The service symbols and words that allow you to organize the synchronization and relativity of avatar movements were introduced to the developed script language. The features of the realization of various animation ways for the demonstration of words and word combinations are considered. To realize this concept the software package with two modes: "Developer" and "User" has been developed. The article describes the hierarchical structure and functional possibilities of the software complex for the creating of sign language guide and lessons, and the prospects for its further use.

Keywords: Russian Sign Language, a multimedia dictionary, notation Dimskis, Unity3D, animated character, avatar

For citation:

Grif M. G., Lukoyanychev A. V. The Program Complex for Training Russian Sign Language Based on Unity3D, *Programmnaya Ingeneria*, 2018, vol. 9, no. 8, pp. 375–384.

DOI: 10.17587/prin.9.375-384

References

1. **Gosudarstvennaya programma Rossijskoj Federacii "Dostupnaya sreda"** na 2011–2020 gody, available at: <http://www.rosmintrud.ru/docs/government/170> (date of access 14.05.2018) (in Russian).
2. **Grif M. G., Lukoyanychev A. V.** 3D-animacija russkogo zhestovogo yazuka na osnove notacii Dimskis (3D-animation of Russian sign language on the basis of Dimskis notation), *Programmnaya Ingeneria*, 2017, vol. 8, no. 7, pp. 310–318 (in Russian).
3. **Karpov A. A., Kagirov I. A.** Formalizaciya leksikona sistemy kompyuternogo sinteza yazuka zhestov (The formalization of the lexicon of computer synthesis of sign language system), *Trudy SPIIRAN*, 2011. iss. 1 (16), pp. 123–140 (in Russian).
4. **Manueva J. S., Grif M. G., Kozlov A. N.** Postroenie sistemy kompyuternogo surdoperevoda russkogo yazyka (Computer Sign Language Interpretation System Development of Russian Language), *Trudy SPIIRAN*, 2014, no. 6 (37), pp. 170–187 (in Russian).
5. **Dimskis L. S.** *Izuchenie zhestovyy yazyk: uchebnoe posobie* (The study of sign language: study guide), Academy, 2002, 128 p. (in Russian).
6. **Smith M., Queiroz C.** *Unity5.x Cookbook*, Packt Publishing, 2015, 570 p.
7. **Zucconi A.** *An Introduction to Procedural Animations*, available at: <https://www.alanzucconi.com/2017/04/17/procedural-animations/> (date of access 14.05.2018).
8. **Grif M. G., Lukoyanychev A. V.** Multimediynyi slovar russkogo zhestovogo yazuka na osnove notacii Dimskis (Multimedia Dictionary of Russian Sign Language based on Notation Dimskis), *Vestnik AGTU, Ser. Management, Computer Science and Informatics*, 2017, no. 1, pp. 105–114 (in Russian).

ООО "Издательство "Новые технологии". 107076, Москва, Стромынский пер., 4
Технический редактор Е. М. Патрушева. Корректор Е. В. Комиссарова

Сдано в набор 07.08.2018 г. Подписано в печать 24.09.2018 г. Формат 60×88 1/8. Заказ PI818
Цена свободная.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

Рисунки к статье М. Г. Грифа, А. В. Лукоянчева
«ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ОБУЧЕНИЯ
РУССКОМУ ЖЕСТОВОМУ ЯЗЫКУ НА ОСНОВЕ Unity3D»

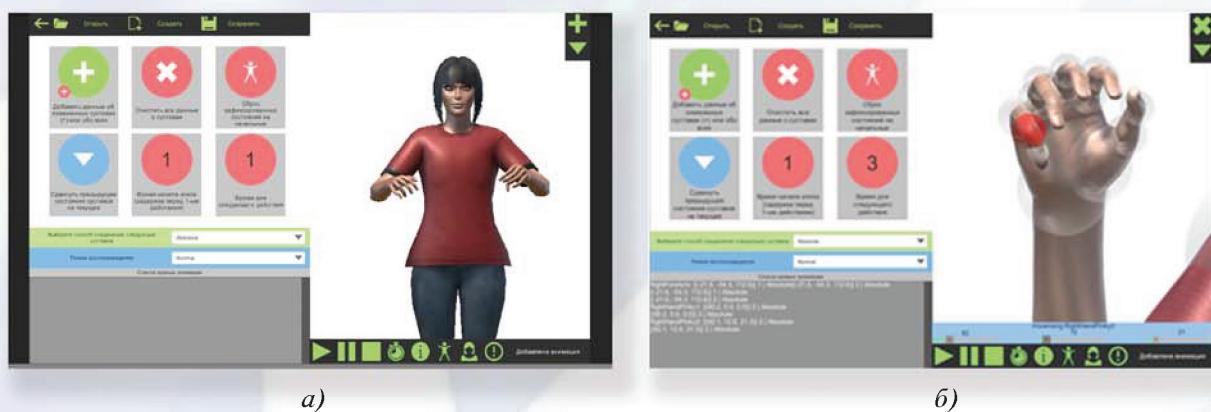


Рис. 11. Внешний вид программы (а) и пример редактирования конфигурации руки (б)



Рис. 12. Внешний вид редакторов знаков нотации (а) и справочника (б)

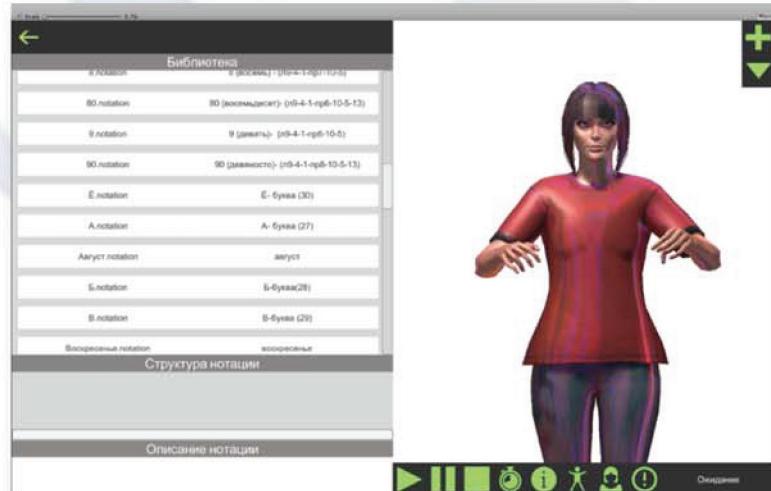
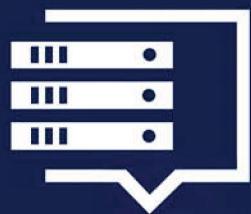


Рис. 13. Внешний вид выовера справочника дактилем и слов



PROSTOR

КОНФЕРЕНЦИЯ ПО ТЕХНОЛОГИЯМ ХРАНЕНИЯ ДАННЫХ

1 ноября 2018

ЧТО ТАКОЕ PROSTOR?

Это специализированное отраслевое событие, которое посвящено анализу тенденций и изменений на рынке СХД, оценке конкурирующих технологий и решению прикладных задач.

Ежегодная конференция PROSTOR объединяет системных интеграторов и производителей аппаратных решений для обсуждения вопросов хранения данных.



Москва, ст. м. Кропотkinsкая
Deworkacy Красный Октябрь

PROSTOR.RAIDX.RU

Участие **бесплатное** по предварительной регистрации.

Внимание: количество мест ограничено!

+7 812 622 16 80

request@raidx.ru