ISSN 2220-3397

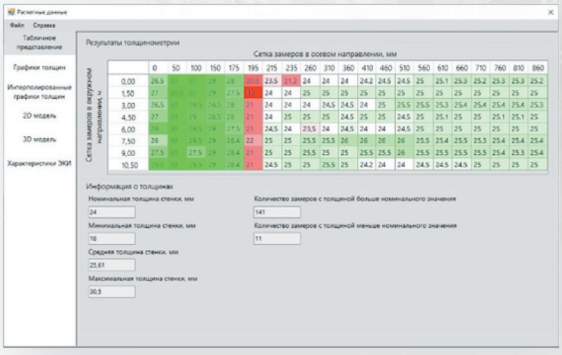
ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Программная инженерия

ПрИн
5
2020
Том 11

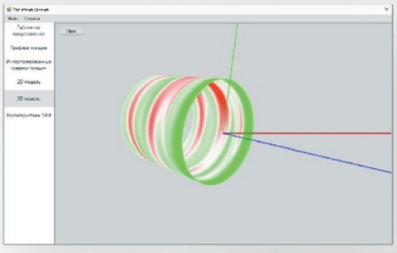
Рисунок к статье **О. М. Гулиной, Н. Л. Сальникова, В. П. Семишкина, М. Н. Типикиной**

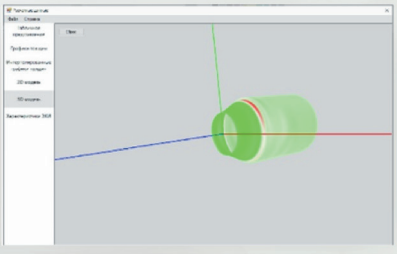
«РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММ ДЛЯ УПРАВЛЕНИЯ РЕСУРСОМ МЕХАНИЧЕСКИХ ЭЛЕМЕНТОВ АЭС В УСЛОВИЯХ ЭРОЗИОННО-КОРРОЗИОННОГО ИЗНОСА»

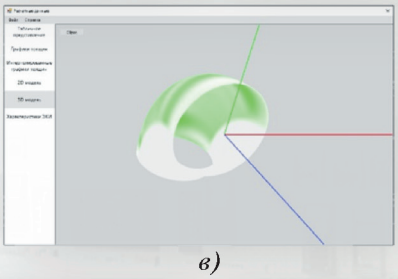


а)

б)

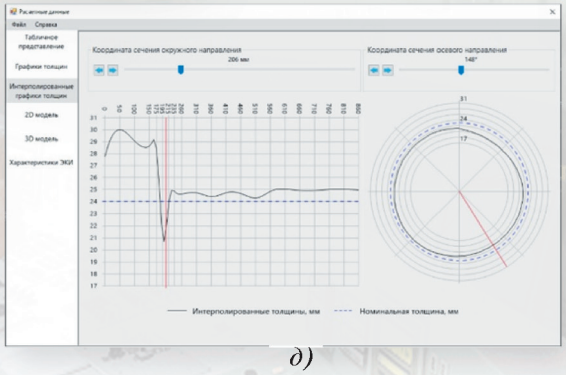




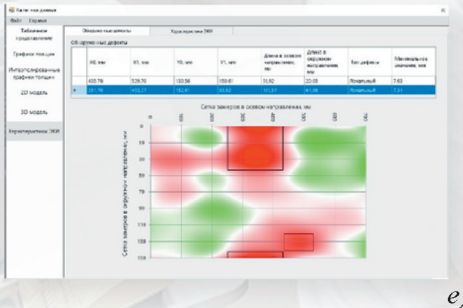


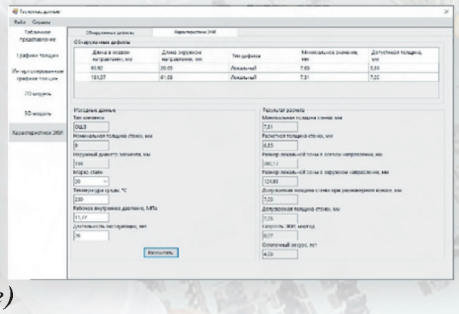
в)

г)



д)





е)

Рис. 6. Результат реализации расширяемой системы

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с сентября 2010 г. DOI 10.17587/issn.2220-3397 ISSN 2220-3397

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

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 (индекс по Объединенному каталогу "Пресса России" — **22765**) или непосредственно в редакции.
Тел.: (499) 269-53-97. Факс: (499) 269-55-10.
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.

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 11
N 5
2020

Published since September 2010 **DOI 10.17587/issn.2220-3397** **ISSN 2220-3397**

CONTENTS

Information about the jornal is available online at:
http://novtex.ru/prin/eng e-mail:prin@novtex.ru

УДК 004.7.056.5 DOI: 10.17587/prin.11.259-269

В. А. Галатенко, д-р физ.-мат. наук, зав. сектором, galat@niisi.ras.ru,
К. А. Костюхин, канд. физ.-мат. наук, ст. науч. сотр., kost@niisi.ras.ru,
Федеральное государственное учреждение "Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук" (ФГУ ФНЦ НИИСИ РАН), Москва

Информационная безопасность интернета вещей: обзор основных положений[1]

Технология интернета вещей развивается быстрыми темпами. Это относится и к промышленному, и к бытовому интернету. Счет "вещей" идет на миллиарды, сформировалось очень много областей их применения. В то же время состояние информационной безопасности интернета вещей нельзя назвать удовлетворительным, защитные меры явно уступают технологиям их коммерческого применения. Это особенно опасно потому что интернет вещей охватывает два мира — цифровой и физический, и нарушения безопасности могут нанести не только информационный, но и физический урон.

Статья представляет собой обзор основных положений информационной безопасности интернета вещей. Сделана попытка рассмотреть как программно-технический, так и законодательный уровни безопасности интернета вещей. Этим она отличается от других публикаций аналогичной направленности. Только целостный, комплексный подход способен повысить реальную информационную безопасность.

Изложены основные понятия, описана эталонная модель интернета вещей, уделено внимание особенностям интернета вещей, существенным с точки зрения безопасности, приведены типичные угрозы для интернета вещей.

Детально рассмотрены законодательный уровень информационной безопасности, рекомендации по безопасности для интернета вещей, описание использования от производителя и установка программных коррекций.

Предполагается, что читатель знаком с основами информационной безопасности. При необходимости можно обратиться к учебному курсу [*1*].

Ключевые слова: *интернет вещей, промышленный интернет вещей, информационная безопасность, киберфизические системы, кибербезопасность, сенсор, актуатор*

Основные понятия

Интернет вещей (*Internet of Things, IoT*) [2] — глобальная инфраструктура для информационного общества, которая обеспечивает возможность предоставления более сложных услуг путем соединения друг с другом (физических и виртуальных) вещей на основе существующих и развивающихся функционально совместимых информационно-коммуникационных технологий.[2]

Имеются и другие определения этого термина, например: *интернет вещей* [3] — инфраструктура взаимосвязанных объектов, людей, систем и информационных ресурсов вместе с интеллектуальными сервисами, позволяющая обрабатывать информацию о физическом и виртуальном мирах и соответствующим образом реагировать на возникающие ситуации.

Авторство термина "интернет вещей" приписывает себе Кевин Эстон (*Kevin Ashton*). Якобы так называлась презентация, сделанная им для компании Procter & Gamble (P&G) в 1999 г. [4].

Вещь (*Thing*) — применительно к интернету вещей означает предмет физического мира (физические вещи) или информационного мира (виртуальные вещи), который может быть идентифицирован и интегрирован в сети связи.

Устройство (*Device*) — применительно к интернету вещей означает элемент оборудования, который обладает обязательными возможностями связи и дополнительными возможностями измерения, срабатывания, а также ввода, хранения и обработки данных.

[1] Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (проведение фундаментальных научных исследований 47 ГП) по теме № 0065-2019-0002 "Исследование и реализация программной платформы для перспективных многоядерных процессоров" (рег. № АААА-А19-119012290074-2).

[2] Благодаря задействованию возможностей идентификации, сбора, обработки и передачи данных в интернете вещей обеспечивается наиболее эффективное использование вещей для предоставления услуг для всех типов приложений при одновременном выполнении требований безопасности и неприкосновенности частной жизни. В широком смысле интернет вещей можно воспринимать как концепцию, имеющую технологические и социальные последствия.

Датчик, сенсор (*Sensor*) [5]: электронное устройство, которое измеряет физическое состояние или химический состав и доставляет электронный сигнал, соответствующий наблюдаемой характеристике.

Исполнительный механизм, актуатор (*Actuator*) [6]: устройство, которое инициирует физическое действие после возбуждения входным сигналом.

Масштабирование применительно к интернету вещей [7]:

- масштабирование вверх — масштабирование интернет-технологий до десятков миллиардов недорогих узлов;
- масштабирование вниз — масштабирование характеристик каждого из этих узлов и сетей, построенных из них, с целью сделать масштабирование вверх экономически и физически целесообразным.

Необходимость масштабирования вниз характеристик узлов ведет к появлению ограниченных узлов.

Ограниченный узел — узел, некоторые характеристики которого, считающиеся само собой разумеющимися для интернет-узлов на данный момент, оказываются недостижимыми, чаще всего в силу ценовых и/или физических ограничений на размер, массу, энергопотребление и т. п.

Если не иметь в виду сетевую связность, можно говорить об *ограниченных устройствах*. Использование ограниченных узлов в сетях зачастую ведет к ограничениям на сами сети, однако сети могут быть ограниченными и по другим причинам, поэтому следующие два термина различаются.

Ограниченная сеть — сеть, некоторые характеристики которой, считающиеся само собой разумеющимися для привычных канальных уровней в интернете на данный момент, оказываются недостижимыми.

Сеть из ограниченных узлов — сеть, на характеристики которой повлияло наличие значительного числа ограниченных узлов. Сеть из ограниченных узлов всегда является ограниченной сетью.

Проблемная сеть — сеть, имеющая серьезные проблемы с тем, что современные приложения вправе ожидать от сквозной IP-модели: например, сквозная IP-связность может не предоставляться вообще или предоставляться с перерывами и т. п. Все проблемные сети считаются ограниченными, но не все ограниченные сети обязательно являются проблемными.

Эталонная модель интернета вещей

Эталонная модель интернета вещей, предложенная международным союзом электросвязи (МСЭ) [2], включает в себя четыре горизонтальных уровня, а также два вертикальных среза, где сосредоточены возможности управления и возможности обеспечения безопасности.

Четырьмя горизонтальными уровнями являются:

- уровень приложения;
- уровень поддержки услуг и поддержки приложений;
- уровень сети;
- уровень устройства.

Роль уровня приложения очевидна.

Уровень поддержки услуг и поддержки приложений включает в себя две группы функциональных возможностей (далее — возможностей):

- общие возможности поддержки — типовые возможности, которые могут использоваться различными приложениями, такие как обработка или хранение данных;
- специализированные возможности поддержки — конкретные возможности, которые предназначены для удовлетворения требований разнообразных приложений.

Уровень сети состоит из следующих двух типов возможностей:

- возможности организации сетей — функции управления сетевыми соединениями;
- возможности транспортировки — предоставление соединений для транспортировки информации.

Возможности уровня устройства подразделяются на возможности устройства и возможности шлюза.

Возможности устройства включают в себя:

- ◇ прямое взаимодействие с сетью (без использования возможностей шлюза);
- ◇ непрямое взаимодействие с сетью (с помощью возможностей шлюза);
- ◇ организацию специальных сетей (устройства могут иметь возможность строить сети произвольным образом);
- ◇ спящий режим и пробуждение (могут поддерживаться механизмы "сна" и "пробуждения" для экономии энергии).

Возможности шлюза включают, в том числе:

- ○ поддержку нескольких интерфейсов с использованием различных проводных и беспроводных технологий;
- ○ преобразование протокола.

Возможности управления в интернете вещей охватывают управление неисправностями, управление конфигурацией, управление учетом, управление показателями работы и управление безопасностью.

Важнейшие общие возможности управления включают:

- управление устройствами, например, дистанционные активацию и деактивацию устройств, диагностику, обновление прошивки и/или программного обеспечения, управление рабочим состоянием устройств;
- управление топологией локальной сети;
- управление трафиком и перегрузками.

Существует два вида возможностей обеспечения безопасности: общие и специализированные. Общие возможности обеспечения безопасности не зависят от приложений и включают:

- ◇ на уровне приложения — авторизацию, аутентификацию, защиту конфиденциальности и целостности данных, защиту приватности, аудит безопасности, антивирусы;

- ◇ на уровне сети — авторизацию, аутентификацию, защиту конфиденциальности и целостности данных;
- ◇ на уровне устройства — аутентификацию, авторизацию, проверку целостности устройства, управление доступом, защиту конфиденциальности и целостности данных.

Специализированные возможности обеспечения безопасности тесно связаны с требованиями приложений, например с требованиями безопасности мобильных платежей.

Предложено большое число других эталонных моделей интернета вещей. Составить представление о них можно, например по обзорам [8—10].

Особенности интернета вещей, существенные с точки зрения безопасности

Важнейшей особенностью интернета вещей является масштаб. Здесь важно как масштабирование вверх — по числу устройств, так и масштабирование вниз — возможности устройств интернета вещей ограничены. Большое число скомпрометированных устройств чревато сильными атаками, ограниченность возможностей затрудняет реализацию сервисов безопасности.

Обычно в качестве целей безопасности фигурируют конфиденциальность, целостность и доступность. Из этой триады для интернета вещей (особенно промышленного) на первое место следует поставить доступность, на второе — целостность.

Кроме того, следует учитывать долгий срок службы устройств интернета вещей — десять и более лет. Это значит, что нужно изначально предусмотреть механизм обновления их встроенного, системного и прикладного программного обеспечения.

Если устройство промышленного интернета вещей управляет непрерывным производством, его нельзя останавливать и/или перезагружать. Обновление программного обеспечения, отражение и ликвидация последствий атак должны осуществляться "на ходу", что делает обязательным контролируемое выполнение, самолечение.

В процессе длительной непрерывной эксплуатации характеристики сервисов, предоставляемых устройствами промышленного интернета вещей, не должны деградировать.

К устройствам промышленного интернета вещей предъявляются требования "жесткого" реального времени. Нарушение этих требований — угроза безопасности.

Устройства интернета вещей взаимодействуют с внешним физическим миром: получают данные от датчиков (сенсоров) и выдают управляющие воздействия на исполнительные устройства (актуаторы). Должна быть обеспечена целостность этих двух потоков данных во времени, а все аномалии должны оперативно выявляться.

Типичные угрозы для интернета вещей

Предполагается высокая мотивированность атакующих, их высокая квалификация в сфере IT и информационной безопасности, а также обеспеченность ресурсами.

На уровне сенсоров и актуаторов возможно их физическое повреждение, удаление/вставка/замена. Вредоносные сенсоры и актуаторы могут интерферировать с легальными. Это делает необходимым аутентификацию сенсоров и актуаторов, выявление их аномального покомпонентного и совокупного поведения.

На сетевом уровне возможны все традиционные атаки. К их числу относятся перечисленные далее.

- Пассивное прослушивание сети (атака на конфиденциальность).
- Активное прослушивание сети (вмешательство в сетевые потоки данных — атаки на целостность): изменение отдельных пакетов, их переупорядочение, воспроизведение ранее переданных пакетов, вставка, удаление пакетов. К этой группе принадлежит "впрыскивание" поддельных данных сенсоров и поддельных управляющих сигналов для актуаторов.
- Изменение структуры сети, вставка/удаление узлов, изменение правил маршрутизации, атаки на протокол разрешения адресов (ARP).
- Атаки на управляющий протокол ICMP, например, отправка ping-пакетов с широковещательным обратным адресом и т. п.
- Атаки на доступность, например, SYN-наводнение.
- Отправка некорректных пакетов. Это особенно опасная атака, так как для ее отражения требуется безупречная реализация сетевых сервисов в промышленных контроллерах. В середине 2019 г. в системе VxWorks было выявлено 11 уязвимостей, 6 из которых — критичные, допускающие, например, удаленное выполнение произвольного кода внешними злоумышленниками. Под угрозой оказались примерно два миллиарда устройств.
- Stuxnet-подобные атаки. Для их реализации атакующий должен получить полный контроль над всеми коммуникационными каналами между промышленными контроллерами, сенсорами и актуаторами.

На уровне собственно промышленных контроллеров основными атаками являются перечисленные далее.

- o Выполнение кода, нужного злоумышленнику. Для достижения цели может применяться "впрыскивание" кода или использование уже имеющегося кода (так называемое программирование, ориентированное на возвраты). Первой фазой атаки обычно является эксплуатация уязвимости "переполнение буфера".
- o Нарушение целостности и/или конфиденциальности данных путем переполнения буфера.
- o Обход механизмов аутентификации и/или авторизации для несанкционированного доступа и эскалации привилегий.

- Атаки на доступность путем агрессивного потребления ресурсов, например, процессора, памяти, средств синхронизации и т. п.
- Атаки на доступность путем создания тупиков.
- Использование отладочных интерфейсов (аппаратных и/или программных).

Для устройств промышленного интернета вещей актуальны следующие угрозы:

- несанкционированный доступ к локальным, сетевым и отладочным интерфейсам;
- нарушение целостности аппаратной и программной конфигурации;
- компрометация криптографического материала;
- небезопасное начальное состояние;
- неумелое переконфигурирование пользователем;
- переполнение буфера и выполнение данных как программ;
- отсутствие возможности или чрезмерная сложность устранения выявленных уязвимостей;
- отсутствие возможности разобраться в ситуации после зависания.

Законодательный уровень информационной безопасности

Интернет вещей нуждается в законодательном регулировании и государственной поддержке. Одним из первых шагов в этом направлении стал представленный в сенат США 1 августа 2017 г. законопроект S.1691 [11], устанавливающий минимальные эксплуатационные стандарты кибербезопасности для устройств, имеющих выход в интернет и закупаемых федеральными агентствами США. Перечислим основные положения.

- Подобные устройства не должны иметь фиксированные или жестко зашитые верительные данные безопасности (пароли, криптографические ключи и т. п.). У пользователей должна быть возможность установить эти данные по своему усмотрению.
- Аппаратное и программное обеспечение не должно иметь известных уязвимостей.
- Должна предоставляться возможность аутентифицированных обновлений.
- В устройствах должны использоваться актуальные версии коммуникационных и криптографических протоколов и технологий.

Аналогичный по направленности закон, вступивший в силу 1 января 2020 г., принят в штате Калифорния [12]. Закон обязывает производителей устройств с сетевой связностью снабжать устройства надлежащими защитными средствами (в соответствии с характером устройства и обрабатываемой им информацией), чтобы предотвратить несанкционированный доступ, уничтожение, использование, модификацию и раскрытие информации.

Более конкретным является требование уникальности предустановленного пароля для каждого устройства и необходимость установки нового пароля перед первым использованием.

Евросоюз также озабочен кибербезопасностью информационно-коммуникационных технологий. Основой соответствующих мер служит закон о кибербезопасности [13]. В законе констатируется, что цифровизация и сетевая связность стали всепроникающими, что с развитием интернета вещей число потребительских устройств, имеющих интернет-доступ, быстро растет, однако их кибербезопасность нельзя считать удовлетворительной, производители не закладывают безопасность в проекты устройств. Такое положение дел недопустимо, производители во всех странах Евросоюза должны следовать стандартам кибербезопасности.

Закон предусматривает формирование современной системы сертификации по требованиям кибербезопасности. Считается, что от этого выиграют как пользователи, которые будут иметь представление о степени защищенности приобретаемых ими устройств, так и производители, получающие конкурентные преимущества.

Рекомендации по безопасности для интернета вещей

Европейское агентство по сетевой и информационной безопасности (*European Union Agency for Network and Information Security*, ENISA) выпустило базовые рекомендации по безопасности для интернета вещей [14]. Рекомендации подразделяются на три категории:

- политика безопасности;
- организационные меры;
- технические меры.

Ключевой элемент политики безопасности — целенаправленное обеспечение защищенности и приватности, начиная со стадии проектирования, соблюдение архитектурных и технологических принципов обеспечения безопасности. Предусмотрено управление активами, идентификация и оценка рисков и угроз.

На организационном уровне должна быть предусмотрена поддержка всего жизненного цикла изделий, оговорены условия прекращения поддержки, должны применяться проверенные решения, заранее вырабатываться процедуры реагирования на инциденты и выявления уязвимостей, программы информирования и обучения пользователей, регламентироваться отношения со сторонними организациями.

Наиболее обширной является категория технических мер. Она включает в себя следующие подкатегории:

- аппаратную защиту (аппаратный корень доверия, специализированные сопроцессоры безопасности и т. д.);
- управление доверием и целостностью (доверенная загрузка, криптографическое подписывание кода, возможность возврата к гарантированно безопасному состоянию и т. п.);

- сильную подразумеваемую защищенность и приватность (все применимые защитные средства должны быть по умолчанию включены, все ненужные функциональные возможности должны быть выключены, подразумеваемые пароли должны быть индивидуальными и сильными);
- защиту персональных данных (собирать минимум данных, пользователи должны иметь возможность узнать, какие данные собираются);
- обеспечение надежной, безопасной работы (предусматривать возможность аварий и автономной работы, использовать средства самодиагностики и самолечения);
- автоматические безопасные обновления программного обеспечения (обеспечение обратной совместимости, сохранение пользовательских настроек);
- аутентификацию (обязательная смена подразумеваемых входных имен и паролей при инициализации, сильные пароли, двух- или многофакторная аутентификация, защита от переборных атак, невозможность использования атакующим механизмов восстановления или переустановки пароля, шифрование верительных данных безопасности);
- авторизацию (реализация принципа минимизации привилегий, изоляция привилегированного кода);
- управление доступом, физическую защиту (контроль конфиденциальности и целостности, выявление и реагирование на несанкционированное изменение аппаратуры, шифрование носителей данных, отсутствие лишних аппаратных интерфейсов);
- криптографию (надлежащее и эффективное использование, возможность применения легковесной криптографии, безопасное управление ключами);
- защищенные и доверенные коммуникации (обеспечение доступности, целостности и конфиденциальности, использование актуальных протоколов, верификация всех соединений, следование ограничительной политике безопасности — все, что не разрешено, запрещено, ограничение темпа передачи для противодействия автоматизированным атакам);
- защищенные интерфейсы и сетевые сервисы (сегментация рисков, изоляция скомпрометированных компонентов, индивидуальные секретные ключи для устройств, балансировка нагрузки, сквозное шифрование пользовательских сеансов);
- безопасную обработку входных и выходных данных (проверка входных и фильтрация выходных данных);
- протоколирование событий безопасности;
- мониторинг и аудит (верификация поведения устройств и эффективности регуляторов безопасности).

В документе [14] проведен анализ имеющихся проблем в кибербезопасности интернета вещей. К их числу относятся:

- отсутствие целостного, единого подхода к кибербезопасности и законотворчеству в этой области;
- недостаток информированности и квалификации;
- небезопасное проектирование и разработка (архитектурные просчеты, известные уязвимости);
- недостаточная взаимная совместимость различных устройств, платформ и каркасов;
- недостаточное экономическое стимулирование обеспечения безопасности;
- ненадлежащее управление жизненным циклом изделий.

Для решения перечисленных проблем предлагаются следующие высокоуровневые рекомендации:

- поддержка инициатив и законотворчества в области безопасности интернета вещей;
- информирование о необходимости обеспечить безопасность интернета вещей (обучение пользователей, повышение квалификации разработчиков);
- подготовка руководств по управлению жизненным циклом аппаратного и программного обеспечения интернета вещей (изначальная ориентация на безопасность, защищенность подразумеваемых конфигураций);
- достижение взаимной совместимости устройств и платформ разных производителей (поддержка открытых стандартов и каркасов безопасности);
- экономическое и административное стимулирование защиты интернета вещей;
- формирование безопасного управления жизненным циклом продуктов и услуг интернета вещей;
- достижение взаимопонимания и согласия между всеми заинтересованными сторонами.

В концентрированном виде рекомендации по обеспечению безопасности и защищенности интернета вещей приведены в своде правил [15], разработанном по инициативе правительства Великобритании в рамках программы "безопасность по проекту" (*Secure by Design*) [16].

Рекомендованы следующие 13 правил.

1. Не должно быть подразумеваемых паролей (все пароли устройств должны быть уникальными и не допускать переустановку в единое фабричное подразумеваемое значение).

2. Должна быть реализована политика раскрытия уязвимостей (все поставщики интернет-устройств и услуг должны предоставлять общедоступную точку контакта для докладов об уязвимостях, уязвимости должны оперативно ликвидироваться; например, в Великобритании доклады об уязвимостях можно направлять по адресу https://www.ncsc.gov.uk/contact).

3. Программное обеспечение интернет-устройств должно поддерживаться в актуальном состоянии (изменения должны легко реализовываться, быть безопасными, своевременными и не нарушать нормального функционирования устройств; причины изменений должны быть понятны потребителям; ограниченные устройства могут заменяться целиком).

4. Данные, критичные с точки зрения безопасности, должны храниться защищенным образом, верительные данные безопасности (пароли, криптографические ключи), жестко зашитые в программы, не допускаются, поскольку их несложно раскрыть.

5. Коммуникации должны быть защищенными, сеансы удаленного администрирования должны шифроваться, настоятельно рекомендуется использование открытых, апробированных интернет-стандартов.

6. Должен соблюдаться принцип минимизации привилегий, все неиспользуемое должно быть удалено или деактивировано, это уменьшает возможности для атак.

7. Целостность программного обеспечения должна контролироваться, должны использоваться механизмы доверенной загрузки, при выявлении нарушения целостности должен возбуждаться сигнал тревоги, сетевая связность должна минимизироваться и должен выполняться возврат к заведомо безопасному состоянию.

8. Должно соблюдаться законодательство о защите персональных данных.

9. Системы должны быть устойчивы к отказам (к потере сетевой связности и электропитания); по возможности сервисы должны оставаться работоспособными в локальном режиме и аккуратно восстанавливаться при возобновлении электропитания.

10. Следует осуществлять мониторинг данных телеметрии для выявления аномального поведения.

11. Потребители должны иметь ясные инструкции по удалению персональных данных, а сама процедура должна быть несложной (удаление может понадобиться, например, перед выведением устройства из эксплуатации).

12. Установка и сопровождение устройств должны быть несложными, соблюдающими требования информационной безопасности, с дружественным интерфейсом.

13. Необходимо проверять входные данные, поступающие от пользователя, по сети, через прикладной программный интерфейс и т. п. В частности, должны контролироваться типы и значения.

Обсуждается введение меток безопасности для потребительских устройств интернета вещей [17]. Такие метки позволили бы потребителям принимать более взвешенные решения при покупке устройств. Иными словами, предлагается дисциплина добровольной сертификации по требованиям безопасности, которая давала бы конкурентные преимущества прошедшим ее компаниям.

Активно работает в области безопасности интернета вещей национальный институт стандартов и технологий (NIST) США. Доклад [18] носит вводный характер, в нем поясняется специфика безопасности интернета вещей:

- взаимодействие с физическим миром;
- различия в управлении по сравнению с традиционными компьютерами;
- отличия в наборе имеющихся защитных средств.

Высокоуровневыми целями безопасности для устройств интернета вещей являются:

- защита устройств (они не должны становиться плацдармами для атак);
- защита данных (обеспечение доступности, целостности, конфиденциальности и приватности).

Для достижения целей предлагается проводить анализ рисков и корректировать политику безопасности с учетом результатов этого анализа. Среди категорий защитных мер выделены:

- управление активами;
- управление уязвимостями;
- управление доступом (физическим и логическим);
- защита данных (при хранении, обработке и передаче);
- активный аудит (выявление нарушений безопасности и реагирование на них).

Отправной точкой для производителей устройств, желающих обеспечить приемлемый уровень защищенности, может служить доклад [19]. В нем определен базовый набор защитных средств для интернета вещей. Кроме того, описан подход к обеспечению информационной безопасности интернета вещей на основе анализа рисков. Это важно для расширения базового набора защитных средств с учетом специфики конкретных устройств и их использования. Наконец, производители устройств получают возможность ознакомиться с передовым опытом в этой области.

Устройство называется минимально защитимым, если оно снабжено техническими средствами (аппаратными и программными), позволяющими пользователям реализовать регуляторы безопасности, необходимые для нейтрализации типичных рисков. Ответственным за безопасность остается пользователь, но встроенные технические средства делают обеспечение безопасности более простым и эффективным.

При проектировании устройств желательно представлять себе будущих пользователей и то, как, где и в каком окружении они будут применять устройства. Корпоративным пользователям желательно иметь развитые средства конфигурирования, в то время как индивидуальным пользователям излишняя конфигурируемость может скорее помешать. Если пропускная способность сети невелика, то установка больших программных коррекций может быть проблематичной. Важны характер обрабатываемых данных и то, является ли устройство общедоступным (как, например, торговый автомат) или расположено в контролируемой зоне.

В базовый набор защитных средств входят:

- идентификация устройств (логическая и физическая), являющаяся необходимым условием управления активами, уязвимостями и доступом;

- конфигурирование программного обеспечения устройств авторизованными субъектами, в частности, восстановление безопасной подразумеваемой конфигурации;
- защита данных, в частности, криптографическими средствами;
- управление логическим доступом к интерфейсам (локальным и сетевым), в частности, возможность отключать неиспользуемые интерфейсы;
- установка программных коррекций авторизованными субъектами с помощью безопасных и конфигурируемых механизмов;
- протоколирование событий безопасности с предоставлением доступа только авторизованным субъектам.

На безопасность устройства влияют проектные решения. К числу рекомендуемых решений принадлежат следующие принципы проектирования:

◇ обеспечение запаса ресурсов, аппаратных и программных (достаточных, в частности, для поддержки криптографии);

◇ предусмотрительность, поддержка будущей модернизации (например, увеличения длины криптографических ключей);

◇ аппаратная поддержка защитных средств (например, аппаратный корень доверия, аппаратно защищенное хранилище криптографических ключей);

◇ концептуальная экономия (например, отсутствие или возможность отключения лишних интерфейсов);

◇ отказ от защитных средств, затрудняющих эксплуатацию (пример — блокировка входа после нескольких неудачных попыток аутентификации, что может вызвать отказ в обслуживании); у пользователя должна быть возможность конфигурирования или отключения подобных возможностей.

Устройство может полагаться на защищенность своего окружения. Например, в физически защищенных помещениях можно не заботиться о блокировании локальных интерфейсов. Можно полагаться на шлюзы как защиту при взаимодействии с внешними сетями.

Информирование пользователей — важный аспект информационной безопасности. Производители должны предоставлять подобную информацию в наглядной, доступной форме, с учетом специфики предполагаемых пользователей. Пользователи должны знать:

— защитные возможности устройств, от каких рисков они предохраняют, как их лучше использовать;

— как устройство устанавливать, конфигурировать, обслуживать и выводить из использования;

— перечень типов данных, которые устройство может собирать, а также список тех, кто имеет потенциальный доступ к данным;

— наличие программных обновлений, порядок их установки, возможные изменения конфигурации и т. п.

Весьма полезным является проведенный Национальным институтом стандартов и технологий США выборочный анализ информационной безопасности потребительских устройств для умного дома, таких как умные лампы, розетки, камеры, телевизоры и т. д. [20]. Картина получилась неоднородной. Многие устройства защищены на современном уровне, используют сильную криптографию и актуальные версии коммуникационных протоколов. Но есть и слабо защищенные изделия, не препятствующие несанкционированному доступу. Для производителей устройств сформулированы следующие рекомендации:

- при начальном конфигурировании устройств от пользователя должна требоваться установка уникального сильного пароля;
- коммуникационные протоколы и криптография должны быть актуальными; в частности, следует использовать прикрепление сертификатов и открытых ключей [21] для защиты от атак типа "нелегальный посредник";
- ликвидация выявленных уязвимостей должна проводиться оперативно, а программные коррекции должны распространяться и устанавливаться безопасным образом, с возможностью откатки при возникновении проблем;
- доступ к неиспользуемым физическим и логическим портам должен быть заблокирован;
- на устройствах, расположенных вне контролируемой зоны, кнопки перезагрузки быть не должно;
- защитные средства должны иметь интерфейс, дружественный для нетехнических пользователей.

Очень ценным является обзор международных стандартов в области безопасности интернета вещей [22].

Две главные проблемы — наличие стандартов и их реализация. Благополучной можно считать ситуацию в области сетевой и физической безопасности: набор стандартов полон и они реализованы. Во всех остальных областях реализация стандартов отстает — для доверенной аппаратуры и системной инженерии они не реализованы. Что касается полноты набора стандартов, то к относительно благополучным можно отнести управление доступом и криптографию: отстает только стандартизация медицинских устройств и (для криптографии) промышленный интернет вещей. Для системной инженерии безопасности стандартов слишком мало. В целом стандартизацию и реализацию можно охарактеризовать как фрагментарные, с недостаточным темпом обновления.

Описание использования от производителя

Специализированные, ограниченные устройства, образующие интернет вещей, предназначены для выполнения определенных функций. Прочие использования можно считать ненадлежащими и блокировать их. Из ограниченности способов использования вытекает ограниченность коммуни-

кационного профиля устройства. Производитель может специфицировать этот профиль, а средства сетевого управления — запретить прочую сетевую активность. Это как защищает само устройство от внешних атак, так и препятствует участию скомпрометированных устройств в атаках на другие узлы сети. В этом состоит основная идея спецификации описания использования от производителя (*Manufacturer Usage Description*, MUD) [23]. Спецификацией предусмотрено три основных архитектурных элемента:

- универсальный локатор ресурсов (MUD-URL), позволяющий добраться до описания;
- собственно описание (MUD-файл, хранящийся на файловом сервере);
- средства системы сетевого управления для выборки и интерпретации описания (MUD-менеджер).

MUD-URL должен использовать протокол https. Задаваться он может, например, в сертификате X.509 или в рамках протокола динамического конфигурирования хоста (DHCP [24]).

С MUD-файлом ассоциирован файл подписи, служащий для контроля целостности. В MUD-файле могут задаваться конкретные хосты для облачных сервисов и классы устройств (например, устройства того же производителя) для взаимодействия в рамках локальной сети. Обычно разрешается доступ к локальным сервисам DNS/NTP.

MUD-менеджер интерпретирует содержимое MUD-файла и добавляет маршрутизаторам и/или коммутаторам правила фильтрации сетевых потоков данных. Таким образом, сетевой администратор избавлен от ручной работы, что является необходимым условием масштабируемости.

Национальный центр передового опыта в кибербезопасности (*National Cybersecurity Center of Excellence*, NCCoE) Национального института стандартов и технологий США реализовал проект, использующий MUD для защиты устройств интернета вещей в рамках малого бизнеса и домашних сетей (краткое изложение см. в работе [25]).

Одна из возможных конфигураций включает, помимо собственно MUD-элементов, серверы сигнализации угроз и программных коррекций.

Рассмотрены четыре варианта поддержки MUD. Одну из них составляют Cisco MUD manager (программное обеспечение с открытыми исходными текстами), коммутатор Cisco Catalyst 3850-S, сертификаты от DigiCert и некоторые другие элементы. Cisco MUD manager поддержан реализацией (также с открытыми исходными текстами) сервера аутентификации, авторизации и учета FreeRADIUS. MUD-файлы создаются с помощью инструмента MUD file maker (https://www.mudmaker.org/). Правила для разрешенных коммуникаций задаются в удобочитаемом виде, пример подобного правила приведен на листинге.

```
   internet communication class — доступ
к облачным сервисам и другим интернет-хостам:
      host: www.osmud.org
      protocol: TCP
      direction-initiated: from IoT device
      source port: any
      destination port: 443
```

Листинг. Пример правила для создания MUD-файла

Описание использования от производителя, вообще говоря, не обязано ограничиваться коммуникационными аспектами. Оно реализует очень глубокую идею, которая в работе [26] называется наложением добровольных ограничений. Подобные ограничения, отражающие специфику устройства и/или семантику приложения, а также особенности реализации, позволяют универсальным образом выявлять несанкционированную активность и пресекать ее.

Установка программных коррекций

Установка программных коррекций — обязательная составная часть жизненного цикла систем. Для устройств интернета вещей она особенно важна ввиду длительного срока службы, но вместе с тем она и особенно трудна в силу ограниченности устройств и содержащих их сетей, большого числа устройств, их расположения в удаленных и труднодоступных местах. Требуется не только устранять уязвимости, но и менять конфигурационные параметры, добавлять функциональность.

Процесс установки программных коррекций должен удовлетворять следующим требованиям [27]:

- принимаемый образ аутентифицирован, его целостность защищена;
- конфиденциальность передаваемого образа может быть защищена, чтобы затруднить получение злоумышленниками открытого бинарного файла.

Основным понятием в данном контексте является манифест, содержащий метаданные о программном образе. Манифест защищен от несанкционированных изменений и содержит информацию для аутентификации источника. Структура манифеста и составляющие его элементы описаны в спецификации [28]. Бинарное представление манифеста определено в документе [29].

Манифест оптимизирован для потребления ограниченными устройствами. Он не является обычным описательным документом и не описывает, чем является программная коррекция, вместо этого специфицируя, что получатель должен делать.

Программный образ может состоять из нескольких частей, если в состав устройства входят несколько микроконтроллеров. Образ также может представлять собой сжатый архив, содержащий программный код, конфигурационные параметры и т. д. По соображениям эффективности в образ могут входить только изменяемые фрагменты. Начальный загрузчик должен решить, загружать ли старый или получить или проверить новый программный образ.

Манифест может содержать требования, влияющие на процесс установки программных коррекций, например, устанавливать только коррекции от определенного производителя, предназначенные для определенной модели, не устанавливать старых коррекций и т. п.

Программный образ создается автором и выгружается на сервер программных коррекций. В процессе распределения коррекций по устройствам участвует потребитель. Оператор устройства отслеживает процесс, контролируя статус.

Заключение

Интернет вещей развивается быстро, поэтому естественно, что он переживает типичные болезни роста — фрагментарность и неравномерность развития. Идейная база для обеспечения безопасности в основном сформирована (но продолжает формироваться), вопрос в том, насколько быстро произойдет гармонизация подходов, а передовые идеи будут восприняты производителями устройств. Многие государственные и негосударственные органы активно продвигают средства безопасности, информируют и обучают производителей и потребителей.

Список литературы

1. **Галатенко В. А.** Основы информационной безопасности. Под редакцией академика РАН В. Б. Бетелина. — М.: ИНТУИТ.РУ, 2003. — 280 с.
2. **Рекомендация** МСЭ-Т Y.4000/Y.2060 (06/2012), Обзор интернета вещей. URL: http://handle.itu.int/11.1002/1000/11559 (дата обращения 27.07.2020).
3. **Information** technology — Internet of Things (IoT) — Definition and Vocabulary. URL: https://www.iso.org/standard/69470.html (дата обращения 27.07.2020).
4. **Ashton K.** That 'Internet of Things' Thing // RFID Journal, June 2009. URL: http://www.rfidjournal.com/articles/view?4986 (дата обращения 27.07.2020).
5. **Рекомендация** МСЭ-Т Y.4105/Y.2221 (01/2010), Требования к обеспечению приложений и услуг повсеместно распространенной сенсорной сети (USN) в среде СПП. URL: http://handle.itu.int/11.1002/1000/10235 (дата обращения 27.07.2020).
6. **Рекомендация** МСЭ-Т Y.4109/Y.2061 (06/2012), Требования к поддержке машинно-ориентированных приложений связи в среде сетей последующих поколений. URL: http://handle.itu.int/11.1002/1000/11560 (дата обращения 27.07.2020).
7. **Bormann C., Ersue M., Keranen A.** Terminology for Constrained-Node Networks // Internet Engineering Task Force (IETF), Request for Comments: 7228, May 2014. URL: https://tools.ietf.org/html/rfc7228 (дата обращения 27.07.2020).
8. **Di Martino B., Rak M., Ficco M.** et al. Internet of things reference architectures, security and interoperability: A survey // Internet of Things. — 2018. — Vol. 1—2. — P. 99—112.
9. **Ande R., Adebisi B., Hammoudeh M., Saleem J.** Internet of Things: Evolution and Technologies from a Security Perspective // Sustainable Cities and Society. — March 2020. — Vol. 54. URL: https://www.sciencedirect.com/science/article/pii/S2210670719303725 (дата обращения 27.07.2020).
10 **Aly M., Khomh F., Haoues M., Quintero A., Yacout S.** Enforcing security in Internet of Things frameworks: A Systematic Literature Review // Internet of Things. — June 2019. — Vol. 6. URL: https://www.sciencedirect.com/science/article/pii/S2542660518300805 (дата обращения 27.07.2020).
11. **S.1691.** Internet of Things (IoT) Cybersecurity Improvement Act of 2017. URL: https://www.congress.gov/bill/115th-congress/senate-bill/1691/text (дата обращения 27.07.2020).
12. **State** of California (2017-2018). SB-327 Information privacy: connected devices, Senate Bill No. 327, Chapter 886. URL: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327 (дата обращения 27.07.2020).
13. **Regulation** (EU) 2019/881 of the European Parliament and of the Council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) No 526/2013 (Cybersecurity Act). URL: https://eur-lex.europa.eu/eli/reg/2019/881/oj (дата обращения 27.07.2020).
14. **Baseline** Security Recommendations for IoT in the context of Critical Information Infrastructures // ENISA, November 2017. URL: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot (дата обращения 27.07.2020).
15. **Code** of Practice for Consumer IoT Security. Department for Digital, Culture, Media and Sport (DCMS), in conjunction with the National Cyber Security Centre (NCSC), October 2018. URL: https://www.gov.uk/government/publications/code-of-practice-for-consumer-iot-security (дата обращения 27.07.2020).
16. **Secure** by Design: Improving the cyber security of consumer Internet of Things. Report. Department for Digital, Culture, Media and Sport (DCMS), March 2018. URL: https://www.gov.uk/government/publications/secure-by-design-report (дата обращения 27.07.2020).
17. **Mandating** security requirements for consumer Internet of Things (IoT) products. Consultation Stage Impact Assessment. Department for Digital, Culture, Media and Sport (DCMS), May 2019. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/798722/Secure_by_Design_Consultation_Stage_Regulatory_Impact_Assessment.pdf (дата обращения 27.07.2020).
18. **Boeckl K., Fagan M., Fisher W.** et al. Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks // NISTIR 8228, June 2019. URL: https://www.nist.gov/publications/considerations-managing-internet-things-iot-cybersecurity-and-privacy-risks (дата обращения 27.07.2020).
19. **Fagan M., Megas K. N., Scarfone K., Smith M.** Functional Cybersecurity Activities for IoT Device Manufacturers // NISTIR 8259, May 2020. URL: https://csrc.nist.gov/publications/detail/nistir/8259/final (дата обращения 27.07.2020).
20. **Fagan M., Yang M., Tan A., Randolph L., Scarfone K.** Security Review of Consumer Home Internet of Things (IoT) Products // Draft NISTIR 8267, October 2019. URL: https://csrc.nist.gov/publications/detail/nistir/8267/draft (дата обращения 27.07.2020).
21. **Walton J., Steven J., Manico J., Wall K., Iramar R.** Certificate and Public Key Pinning // Open Web Application Security Project (OWASP), 2018. URL: https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning (дата обращения 27.07.2020).
22. **Interagency** Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT) // NISTIR 8200, November 2018. URL: https://csrc.nist.gov/publications/detail/nistir/8200/final (дата обращения 27.07.2020).
23. **Lear E., Droms R., Romascanu D.** Manufacturer Usage Description Specification // Internet Engineering Task Force (IETF), Request for Comments: 8520, March 2019. URL: https://tools.ietf.org/html/rfc8520 (дата обращения 27.07.2020).
24. **Mrugalski T., Siodelski M., Volz B.** et al. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) // Internet Engineering Task Force (IETF), Request for Comments: 8415, November 2018. URL: https://tools.ietf.org/html/rfc8415 (дата обращения 27.07.2020).
25. **Dodson D., Polk T., Souppaya M., Barker W. C., Grayeli P., Symington S.** Securing Small-Business and Home Internet of Things (IoT) Devices. Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD) // NIST SPECIAL PUBLICATION 1800-15A, Volume A: Executive Summary, November 2019. URL: https://www.nccoe.nist.gov/projects/building-blocks/mitigating-iot-based-ddos (дата обращения 27.07.2020).
26. **Галатенко А. В., Галатенко В. А.** О постановке задачи разграничения доступа в распределенной объектной среде // Вопросы кибернетики. Информационная безопасность. Операционные системы реального времени. Базы данных. — М.: НИИСИ РАН, 1999. — С. 3—13.
27. **Moran B., Tschofenig H., Brown D., Meriac M.** A Firmware Update Architecture for Internet of Things // Internet Engineering Task Force (IETF) Draft, January 2018. URL: https://tools.ietf.org/id/draft-moran-suit-architecture-01.html (дата обращения 27.07.2020).
28. **Moran B., Tschofenig H., Birkholz H.** An Information Model for Firmware Updates in IoT Devices // Internet Engineering Task Force (IETF) Draft, June 2020. URL: https://tools.ietf.org/html/draft-ietf-suit-information-model-07 (дата обращения 27.07.2020).
29. **Moran B., Tschofenig H., Birkholz H.** A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest // Internet Engineering Task Force (IETF) Draft, July 2020. URL: https://tools.ietf.org/html/draft-ietf-suit-manifest-09 (дата обращения 27.07.2020).

IoT Information Security: Fundamental Statements Review

V. A. Galatenko, galat@niisi.ras.ru, **K. A. Kostyukhin,** kost@niisi.ras.ru, Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences", Moscow, 117218, Russian Federation

Corresponding author:

Kostyukhin Konstantin A., Senior Researcher, Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences", Moscow, 117218, Russian Federation
E-mail: kost@niisi.ras.ru

Received on July 27, 2020
Accepted on August 04, 2020

Internet of things technology is developing at an exceptionally fast pace. This applies to both industrial and consumer Internet. The "things" account for billions, and many areas of application have been formed. At the same time, the state of information security of the Internet of things is not satisfactory, and protective measures are clearly inferior to Commerce. This is especially dangerous because the Internet of things spans two worlds: digital and physical, and security breaches can cause both informational and physical damage.

The Internet of things is developing rapidly, so it is natural that it experiences typical growth diseases-fragmentation and uneven development. The base for ensuring security is mostly formed (but continues to be formed), the question is how quickly there will be a harmonization of approaches, and advanced ideas will be accepted by device manufacturers. Many state and non-state agencies actively promote security tools, inform and train manufacturers and consumers.

The article is an overview of the main provisions of information security of the Internet of things. An attempt is made to consider software and technical and legislative levels of Internet of things security. This makes it different from other publications of a similar nature. Only a holistic, integrated approach can improve real information security.

Authors outline basic concepts and describe a reference model of Internet of things, draw attention to the peculiarities of the Internet of things that are important from the security point of view, enumerate typical threats for Internet of things.

The legislative level of information security, security recommendations for the Internet of things, manufacturer's description of usage, and installation of software corrections are considered in detail.

Keywords: *Internet of Things, IoT, Industrial Internet of Things, IIoT, cyber-physical systems, information security, cyber security, sensor, actuator*

For citation:

Galatenko V. A., Kostyukhin K. A. IoT Information Security: Fundamental Statements Review, *Programmnaya Ingeneria*, 2020, vol. 11, no. 5, pp. 259—269

DOI: 10.17587/prin.11.259-269

References

1. **Galatenko V. A.** *Information security fundamentals,* Moscow, INTUIT.RU, 2003, 280 p. (in Russian).

2. **Recommendation** ITU-T Y.4000/Y.2060 (06/2012), Overview of the Internet of things, available at: http://handle.itu.int/11.1002/1000/11559 (accessed 27.07.2020).

3. **Information** technology — Internet of Things (IoT) — Definition and Vocabulary, available at: https://www.iso.org/standard/69470.html (accessed 27.07.2020).

4. **Ashton K.** That 'Internet of Things' Thing, *RFID Journal*, June 2009, available at: http://www.rfidjournal.com/articles/view?4986 (accessed 27.07.2020).

5. **Recommendation** ITU-T Y.4105/Y.2221 (01/2010), Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment, available at: http://handle.itu.int/11.1002/1000/10235 (accessed 27.07.2020).

6. **Recommendation** ITU-T Y.4109/Y.2061 (06/2012), Requirements for the support of machine-oriented communication applications in the next generation network environment, available at: http://handle.itu.int/11.1002/1000/11560 (accessed 27.07.2020).

7. **Bormann C., Ersue M., Keranen A.** Terminology for Constrained-Node Networks, *Internet Engineering Task Force (IETF)*, Request for Comments: 7228, May 2014, available at: https://tools.ietf.org/html/rfc7228 (accessed 27.07.2020).

8. **Di Martino B., Rak M., Ficco M., Esposito A., Maisto S. A., Nacchia S.** Internet of things reference architectures, security and interoperability: A survey, *Internet of Things,* 2018, vol. 1—2, pp. 99—112.

9. **Ande R., Adebisi B., Hammoudeh M., Saleem J.** Internet of Things: Evolution and Technologies from a Security Perspective, *Sustainable Cities and Society,* March 2020, vol. 54, available at: https://www.sciencedirect.com/science/article/pii/S2210670719303725 (accessed 27.07.2020).

10. **Aly M., Khomh F., Haoues M., Quintero A., Yacout S.** Enforcing security in Internet of Things frameworks: A Systematic Literature Review, *Internet of Things,* June 2019, vol. 6, available at: https://www.sciencedirect.com/science/article/pii/S2542660518300805 (accessed 27.07.2020).

11. **S.1691**. Internet of Things (IoT) Cybersecurity Improvement Act of 2017, available at: https://www.congress.gov/bill/115th-congress/senate-bill/1691/text (accessed 27.07.2020).

12. **State** of California (2017-2018). SB-327 Information privacy: connected devices, Senate Bill No. 327, Chapter 886, available at: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327 (accessed 27.07.2020).

13. **Regulation** (EU) 2019/881 of the European Parliament and of the council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) No. 526/2013 (Cybersecurity Act), available at: https://eur-lex.europa.eu/eli/reg/2019/881/oj (accessed 27.07.2020).

14. **Baseline** Security Recommendations for IoT in the context of Critical Information Infrastructures, *ENISA,* November 2017, available at: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot (accessed 27.07.2020).

15. **Code** of Practice for Consumer IoT Security, Department for Digital, Culture, Media and Sport (DCMS), in conjunction with the National Cyber Security Centre (NCSC), October 2018, available at: https://www.gov.uk/government/publications/code-of-practice-for-consumer-iot-security (accessed 27.07.2020).

16. **Secure** by Design: Improving the cyber security of consumer Internet of Things. Report, Department for Digital, Culture, Media and Sport (DCMS), March 2018, available at: https://www.gov.uk/government/publications/secure-by-design-report (accessed 27.07.2020).

17. **Mandating** security requirements for consumer Internet of Things (IoT) products. Consultation Stage Impact Assessment, Department for Digital, Culture, Media and Sport (DCMS), May 2019, available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/798722/Secure_by_Design_Consultation_Stage_Regulatory_Impact_Assessment.pdf (accessed 27.07.2020).

18. **Boeckl K., Fagan M., Fisher W., Lefkovitz N., Megas K. N., Nadeau E., O'Rourke D. G., Piccarreta B., Scarfone K.** Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks, *NISTIR 8228*, June 2019, available at: https://www.nist.gov/publications/considerations-managing-internet-things-iot-cybersecurity-and-privacy-risks (accessed 27.07.2020).

19. **Fagan M., Megas K. N., Scarfone K., Smith M.** Functional Cybersecurity Activities for IoT Device Manufacturers, *NISTIR 8259*, May 2020, available at: https://csrc.nist.gov/publications/detail/nistir/8259/final (accessed 27.07.2020).

20. **Fagan M., Yang M., Tan A., Randolph L., Scarfone K.** Security Review of Consumer Home Internet of Things (IoT) Products, *Draft NISTIR 8267*, October 2019, available at: https://csrc.nist.gov/publications/detail/nistir/8267/draft (accessed 27.07.2020).

21. **Walton J., Steven J., Manico J., Wall K., Iramar R.** Certificate and Public Key Pinning, Open Web Application Security Project (OWASP), 2018, available at: https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning (accessed 27.07.2020).

22. **Interagency** Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT), *NISTIR 8200*, November 2018, available at: https://csrc.nist.gov/publications/detail/nistir/8200/final (accessed 27.07.2020).

23. **Lear E., Droms R., Romascanu D.** Manufacturer Usage Description Specification, *Internet Engineering Task Force (IETF),* Request for Comments 8520, March 2019, available at: https://tools.ietf.org/html/rfc8520 (accessed 27.07.2020).

24. **Mrugalski T., Siodelski M., Volz B., Yourtchenko A., Richardson M., Jiang S., Lemon T., Winters T.** Dynamic Host Configuration Protocol for IPv6 (DHCPv6, *Internet Engineering Task Force (IETF),* Request for Comments 8415, November 2018, available at: https://tools.ietf.org/html/rfc8415 (accessed 27.07.2020).

25. **Dodson D., Polk T., Souppaya M., Barker W. C., Grayeli P., Symington S.** Securing Small-Business and Home Internet of Things (IoT) Devices. Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD), *NIST SPECIAL PUBLICATION 1800-15A,* Volume A: Executive Summary, November 2019, available at: https://www.nccoe.nist.gov/projects/building-blocks/mitigating-iot-based-ddos (accessed 27.07.2020).

26. **Galatenko A. V., Galatenko V. A.** About the problem of access control in a distributed object environment, *Voprosy kibernetiki. Informacionnaya bezopasnost'. Operacionnye sistemy real'nogo vremeni. Bazy dannyh,* Moscow, FGU FNC NIISI RAN, 1999, pp. 3—13 (in Russian).

27. **Moran B., Tschofenig H., Brown D., Meriac M.** A Firmware Update Architecture for Internet of Things, *Internet Engineering Task Force (IETF) Draft*, January 2018, available at: https://tools.ietf.org/id/draft-moran-suit-architecture-01.html (accessed 27.07.2020).

28. **Moran B., Tschofenig H., Birkholz H.** An Information Model for Firmware Updates in IoT Devices, *Internet Engineering Task Force (IETF) Draft*, June 2020, available at: https://tools.ietf.org/html/draft-ietf-suit-information-model-07 (accessed 27.07.2020).

29. **Moran B., Tschofenig H., Birkholz H.** A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest, *Internet Engineering Task Force (IETF) Draft*, July 2020, available at: https://tools.ietf.org/html/draft-ietf-suit-manifest-09 (accessed 27.07.2020).

ИНФОРМАЦИЯ

УДК 004.2 DOI: 10.17587/prin.11.270-276

В. В. Корнеев, д-р техн. наук, проф., гл. науч. сотр.,
И. Е. Тарасов, д-р техн. наук, консультант,
ФГУП "Научно-исследовательский институт "Квант", Москва, info@rdi-kvant.ru

Архитектура СБИС с конфигурируемым конвейером

Целью статьи является обоснование выбора архитектуры СБИС, ориентированной на технологические нормы 28 нм и менее, для классов задач, пригодных к решению на массово-параллельных вычислителях с конфигурируемыми конвейерными узлами. Предлагаемые функциональные возможности вычислительных узлов СБИС позволяют применять их в различных областях техники, что потенциально увеличивает потребность в выпуске таких СБИС. В качестве общего подхода к их проектированию и построению предлагается архитектура с последовательной обработкой потока данных цепочкой конфигурируемых вычислительных узлов.

Ключевые слова: *СБИС, архитектура, цифровая обработка сигналов*

Введение

Под технологическим сдвигом в микроэлектронике понимают необходимость корректировки подходов к проектированию СБИС и систем на их основе вследствие изменения технологических процессов. К таким заметным сдвигам можно отнести переход к синхронному стилю описания цифровых схем, произошедший при достижении технологической нормы 90 нм, а также дальнейшее развитие этого направления в виде перехода к "локально синхронным, глобально асинхронным" системам, начиная с норм 28 нм [1—4]. Такие переходы обусловлены отсутствием возможности полного переноса архитектурных и схемотехнических подходов при создании СБИС на базе новых технологических процессов.

Требования и ограничения технологических процессов могут находиться в некотором противоречии с требованиями используемых алгоритмов и программ для решения прикладных задач. Необходимо найти архитектуру, удовлетворяющую всем условиям реализуемости на базе рассматриваемого технологического процесса и эффективную на выбранных классах задач, быть может, при изменении алгоритмов их решения.

Однако сложно говорить об однозначной количественной оценке эффективности тех или иных вычислительных архитектур применительно к классам решаемых задач и к требованиям современных технологий микроэлектроники. Характеристики вычислительного устройства определяются множеством факторов, между которыми сложно установить строгие математические соотношения. Кроме того, в силу специфики разработки СБИС, точные значения ряда характеристик определяются уже на поздних этапах разработки. По этой причине судить об оценке эффективности следует скорее с привлечением аппарата анализа технических рисков.

Высокая стоимость подготовки производства и сильная зависимость характеристик СБИС от возможностей технологических процессов делают востребованным анализ возможных архитектур на основе их технологической реализуемости и эффективности. Впоследствии актуальные задачи могут быть скорректированы или переосмыслены в соответствии с ограничениями технологических процессов. В свое время появление векторно-конвейерного суперкомпьютера CRAY-1 вызвало разработку соответствующих алгоритмов, при исполнении которых достигалась производительность, близкая к пиковой.

Современные тенденции в разработке высокопроизводительных СБИС

Современные технологии микроэлектроники вносят свои ограничения в создаваемые архитектуры [5]. Эти ограничения обусловлены:

- усиливающимся по мере роста степени интеграции энергопотреблением интегральных схем, в значительной мере определяемым суммарной длиной проводников, доставляющих команды и данные к функциональным устройствам;
- ограничениями площади локальных кластеров на кристалле СБИС, состоящих из функциональных устройств и источников команд и операндов для них, использующих синхронное тактовое дерево, а также переход к архитектурам класса GALS (*Globally Asynchronous, Locally Synchronous*), в которых реализована глобальная асинхронность при передаче между синхронными вычислительными устройствами разных кластеров;
- проблемой "темного кремния", заключающейся в схемотехнической возможности формирования схемы, которая при ее реализации на кристалле СБИС приведет к превышению предельно допустимой рассеиваемой мощности, что делает необходимым целый комплекс мер по обеспечению снижения потребляемой мощности как на технологическом, так и на архитектурном уровне.

Вследствие ограничений на время распространения тактового сигнала синхронная схема вычислительного устройства может занимать площадь и содержать количество транзисторов не более, чем напе-

ред определенное для используемых технологических норм и тактовой частоты. Более того, ограничения по потребляемому току в статике и динамике, а также выделяемой и отводимой теплоте, приводят к тому, что в каждом такте может изменяться состояние только части из общего числа транзисторов. По изложенным выше причинам рациональным решением является введение в синхронные вычислительные устройства блоков памяти, создающих эффект большого процента транзисторов, не переключающихся на каждом такте. Транзисторы, оставшиеся после реализации блоков памяти, используются в функциональных блоках вычислительного устройства. Эти обстоятельства наряду с рядом других [5] делают практически реализуемыми архитектуры, которые состоят из совокупности вычислительных устройств, имеющих локальные блоки памяти.

В качестве иллюстрации можно привести изменения в рациональных архитектурах вычислителей на СБИС для работы с глубокими нейросетями, реализованных по разным технологическим процессам 65 [6] и 16 нм [7, 8]. В работе [6] предложена архитектура FlexFlow, в которой в вычислениях на кристалле постоянно задействовано 80 % оборудования на основе использования всех видов параллелизма глубоких сверточных нейросетей.

Однако для технологии 16 нм выбрана другая архитектура, а именно — сеть вычислителей с локальными блоками памяти. На большом кристалле может разместиться много (тысячи) таких вычислителей. Так, на кристалле Cerebras [7] — *Cerebras Wafer Scale Engine* (WSE), площадь которого 46 225 мм^2 со сторонами 21,5 см, размещено 400 000 вычислителей с суммарной памятью 18 Гбайт. Эта память распределена по вычислителям одноуровневой локальной памяти (около 48 Кбайт на один вычислитель) с суммарной пропускной способностью 9 Пбайт/с и коммуникационной средой Swarm с топологией 2D с суммарной пропускной способностью 100 Пбит/с.

Кристалл Graphcore [8] содержит сеть из 1216 вычислителей с локальной памятью 256 Кбайт каждый. Для объединения в единую сеть совокупности кристаллов Graphcore каждый из них имеет 10 каналов с двунаправленной пропускной способностью 64 Гбайт/с. В работе [8] приведен результат измерения пропускной способности однонаправленной передачи между вычислителями разных кристаллов Graphcore на уровне 27 Гбайт/с, что несколько меньше, чем 32 Гбайт/с между соседними вычислителями на одном кристалле. Путем объединения соответствующего числа кристаллов Graphcore может быть получена вычислительная система с требуемой для реализации обучения и получения выводов нейросети производительностью.

Таким образом, на уровне всего кристалла реализуется сетевая архитектура из вычислителей с локальной памятью, что позволяет преодолеть ограничения по энергетике и теплоотводу, в отличие от архитектуры, используемой при технологии 65 нм и максимизирующей загрузку исполнительных устройств.

Организация вычислений в кристаллах, состоящих из вычислителей с локальной памятью, требует решения двух задач: загрузки программ и данных в локальную память; собственно разработки параллельных программ для выполнения требуемых вычислений и управления вычислениями, включая синхронизацию и обмены данными между вычислителями. Если решения о загрузке модулей локальной памяти принимаются в управляющем компьютере, то вычислители с локальной памятью объединяются сетевой средой, как это сделано в вычислительных системах на базе Graphcore [8]. Для кластера из некоторого числа близко расположенных на кристалле вычислителей можно использовать управляющий процессор, обладающий собственной буферной памятью для управления вычислениями этого кластера. Следующий уровень иерархической структуры кристалла образуется путем создания кластеров верхнего уровня под управлением собственных управляющих процессоров. Процесс объединения может продолжаться до получения одного кластера с главным управляющим процессором всего кристалла, имеющим собственную накристальную локальную память, доступ к большой внекристальной памяти и внешним устройствам. Такую иерархическую кластерную архитектуру имеет кристалл PEZY [9].

Далее будут рассмотрены вопросы архитектуры вычислителя с реализацией распределенной арифметики, которая позволяет повысить производительность и энергоэффективность исполнения алгоритмов, допускающих конвейеризацию вычислений. Распределенная арифметика подразумевает реализацию сложных операций путем соединения нескольких функциональных узлов, которые при этом допускают конвейеризацию.

Перспективные области применения конфигурируемых конвейерных вычислителей

Для конвейерных структур обработки данных характерно отсутствие сложных связей между узлами по схеме "каждый с каждым". Это существенно снижает сложность реализации топологии СБИС и связанные с ней технические риски ухудшения характеристик микросхемы, выявляемые на завершающих этапах ее разработки. Исходя из соображений, изложенных выше, следует рассмотреть классы алгоритмов, эффективно реализуемых на базе конвейерных вычислителей, и при этом имеющих перспективы массового применения.

Среди таких классов алгоритмов можно, в первую очередь, упомянуть цифровую обработку сигналов (ЦОС), для которой характерно массовое использование операций "умножение с накоплением" — УН (MAC, *Multiply and Accumulate*). Во многих публикациях отмечается привлекательность конвейеризованных структур для реализации цифровых фильтров. Для задач, основанных на цифровой фильтрации, число операций УН/с (MAC/s) является основным показателем производительности вычислений. Предпочтительным свойством алгоритмов ЦОС для конвейерной реализации можно считать допустимость для них латентности и отсутствие обратных связей по данным, которые могли бы усложнить логику управления конвейеризованной цепочкой вычислителей.

Дополнительно можно отметить возможность генерирования коэффициентов фильтров силами самой СБИС. Такое решение является альтернативой вариантам, при которых коэффициенты хранятся в памяти СБИС или подаются через внешние интерфейсы. В первом случае потребность в памяти может оказаться чрезмерной при реализации фильтров высоких порядков или умножения на гармонические ряды с наложенными оконными функциями. Во втором случае суммарная пропускная способность внешних интерфейсов при повышении производительности СБИС окажется недостаточной при превышении определенного порога производительности, следовательно, архитектура оказывается плохо масштабируемой.

Разработка вычислительного узла для задач распределенной арифметики

Для рассмотренных видов операций, основанных на распределенной арифметике, необходимо реализовать их в виде, соответствующем требованиям цифровой схемотехники в части ограничения площади синхронных регионов и уменьшения длины критического пути [10].

Операция "умножение с накоплением" для распределенной арифметики представляется в виде последовательности умножений на каждый разряд множителя. Для однобитного умножения операция сводится к сложению аккумулятора с множимым, сдвинутым на n разрядов влево (что соответствует умножению на целую степень двойки), или сложению с нулем, если соответствующий разряд множителя равен нулю. Очевидно, что сложение с нулем выполнять нецелесообразно, поэтому при умножении на постоянный коэффициент можно ограничиться умножением только на те разряды, которые установлены в 1 (единицу).

Для потоковой генерации коэффициентов гармонического ряда удобным оказывается алгоритм CORDIC (*Coordinate Rotating Digital Computer*, т. е. цифровой вычислитель для вращения координат) [11]. Этот алгоритм основан на том, что при вращении некоторого вектора отслеживаются как угол поворота вектора, так и связанные с ним координаты конца этого вектора x, y, которые при единичной длине вектора представляют собой косинус и синус от угла поворота:

$$x' = \cos\varphi\,(x - y\,\mathrm{tg}\varphi);$$
$$y' = \cos\varphi\,(y + x\,\mathrm{tg}\varphi).$$

Если поворот будет проводиться только на такие углы, для которых $\mathrm{tg}\,\varphi$ равен целой степени двойки (т. е. 1, 1/2, 1/4, 1/8 и т. д.), то умножение $x\,\mathrm{tg}\,\varphi$ равносильно сдвигу x на 0, 1, 2, 3... позиции вправо. Аналогично выполняется вычисление $y\,\mathrm{tg}\,\varphi$. Таким образом, умножение на синус и косинус в алгоритме оказалось заменено на умножение на 1 (благодаря вынесению за скобки $\cos x$) и умножение на степени двойки, которое может быть выполнено сдвигом.

Имея набор углов поворота φ, тангенсы которых равны целой степени двойки, можно последовательными поворотами на положительные и отрицательные углы добиться того, чтобы суммарный угол поворота стал равен некоторому значению. Имея таблицу углов, и проинициализировав $x = 1$, $y = 0$, можно повернуть угол φ максимально близко к искомому углу φ_0, заданному в качестве аргумента. Если после очередного поворота $\varphi > \varphi_0$, то на следующей итерации алгоритма поворот происходит на отрицательный угол. Таким образом, методом последовательного приближения φ устремляется к φ_0, а его координаты x, y естественным образом представляют косинус и синус угла φ.

Структура вычислителя для реализации алгоритма CORDIC несколько отличается от схем умножения с накоплением, поскольку соответствует последовательным преобразованиям данных. Тем не менее можно указать на интерес к проектам СБИС, ориентированным на аппаратную поддержку этого алгоритма [12]. Кроме того, современная тенденция к использованию программно-определяемого радио также может использовать генерацию ядер свертки в реальном времени, например, для OFDM [13].

Структурные схемы узлов последовательной распределенной арифметики для выполнения операции "шаг умножения" и "шаг алгоритма CORDIC" показаны на рис. 1, 2 соответственно. Анализ структурных схем показывает, что возможна реализация обобщенной схемы, содержащей сумматоры, узлы барабанного сдвига (*barrel shifter*) и конфигурируемые коммутаторы, которая обеспечивала бы динамическую настройку узлов СБИС для выполнения одной из операций. Вариант такой обобщенной схемы показан на рис. 3 (часть узлов удалена для упрощения рисунка). Путем соответствующей установки управляющих сигналов внутренних мультиплексоров схема на рис. 3 может выполнять функции схемы на рис. 1 (шаг умножения с накоплением) или схемы на рис. 2 (шаг алгоритма CORDIC).

Показанные узлы могут быть организованы в виде цепочки, для которой путем анализа целевых алгоритмов и возможностей топологических библиотек требуется уточнить:

1) разрядности регистров-аккумуляторов;

2) количество вычислительных узлов в одной стадии конвейера;

3) размер памяти таблиц;

4) размер памяти для хранения данных о конфигурации.

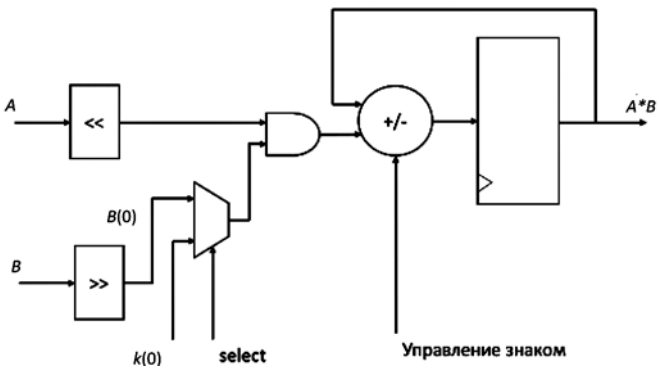


Рис. 1. Структурная схема узла последовательной распределенной арифметики для выполнения операции "шаг умножения"

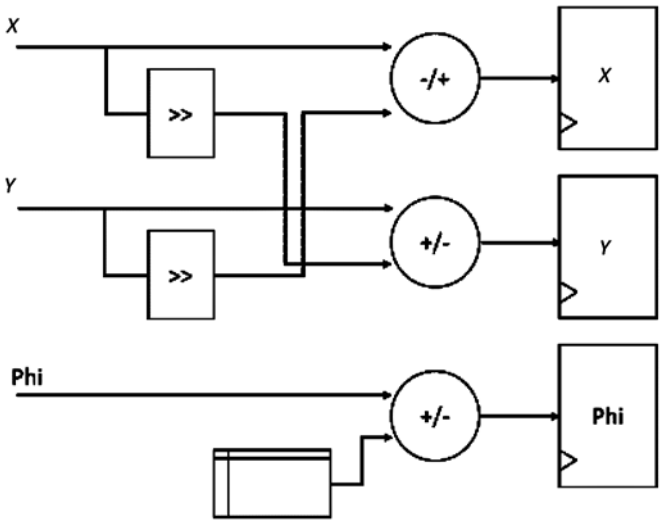


Рис. 2. Структурная схема узла последовательной распределенной арифметики для выполнения операции "шаг алгоритма CORDIC"

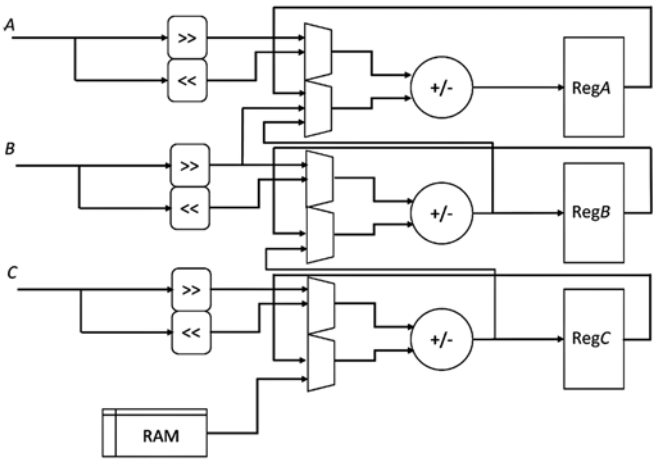


Рис. 3. Вариант структурной схемы узла, обобщающего операции "шаг умножения с накоплением" и "шаг алгоритма CORDIC" путем конфигурирования внутренних мультиплексоров

Для сокращения удельного веса памяти узлы могут быть объединены по архитектуре SIMD, т. е. с организацией параллельных каналов данных, обрабатываемых узлами, имеющими идентичные настройки. Такой подход позволит использовать одну и ту же память для конфигурирования нескольких каналов обработки данных.

Архитектура СБИС с конфигурируемым конвейером для высокопроизводительной распределенной цифровой обработки сигналов

Структурная схема СБИС с конфигурируемым конвейером на базе узлов последовательной распределенной арифметики показана на рис. 4. В этой СБИС имеется: N каналов для ввода сигналов; M каналов цифровой обработки на базе узлов последовательного умножения; K управляющих процессоров кластеров, на которых выполняется в том числе и постобработка.

Сценарии совместной работы каналов обработки данных могут быть следующими:

1) реализация фильтров с конечной импульсной характеристикой (КИХ-фильтров) с постоянными коэффициентами;

2) вычисление корреляционной функции между каналами;

3) генерация гармонических сигналов с помощью одного из каналов и вычисление корреляции между входным и сгенерированным сигналами.

Соответствующее конфигурирование узлов и межканальных коммутаторов выполняется процессорами постобработки. На рис. 5 показан вариант реализации межканального коммутатора для смежных каналов обработки данных. Видно, что выход одного из каналов подан на входной коммутатор второго канала, что позволяет использовать его в качестве входных данных, увеличивая тем самым длину цепочки узлов, выполняющую обработку. На рис. 5 также видно, что при соответствующем топологическом управлении размещением узлов выход цепочки может располагаться в непосредственной близости от входных коммутаторов. Такое решение позволит исключить возможные сложности с чрезмерной длительностью задержек распространения сигналов.

На рис. 6 показаны варианты совместного использования каналов. Вариант на рис. 6, *а* подразумевает независимую работу каналов, когда каждый из входных коммутаторов выбирает соответствующий независимый вход данных. Вариант на рис. 6, *б* демонстрирует совместную работу каналов, когда канал 2 логически объединен с каналом 1, образуя цифровой фильтр более высокого порядка.

Может оказаться, что при большом числе каналов, объединенных кластером обработки данных, сложность коммутатора окажется чрезмерной, что неоправданно увеличит задержки распространения сигналов. Однако в данном случае можно воспользоваться тем фактом, что в системах цифровой обработки сигналов допустима латентность, которая вносится конвейерными структурами. При этом раз-

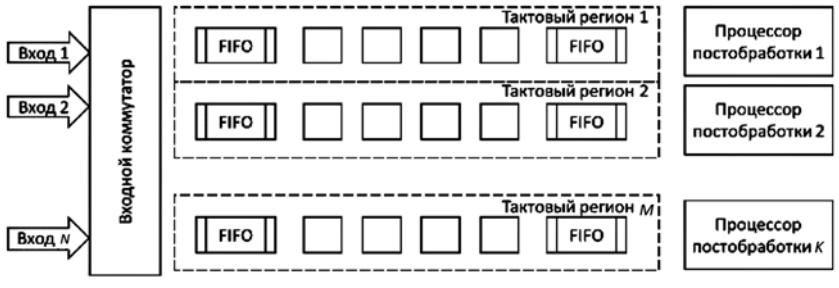


Рис. 4. Структурная схема СБИС с конфигурируемым конвейером

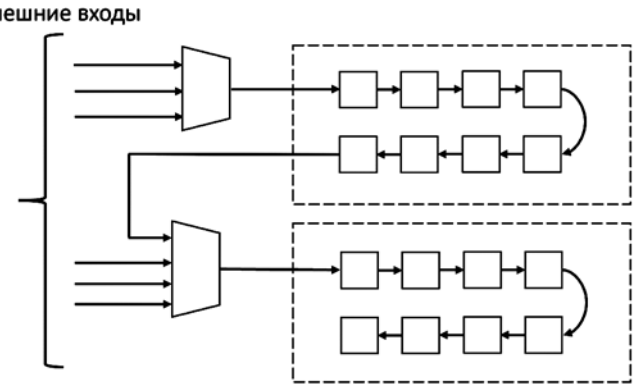


Рис. 5. Межканальные коммутаторы для смежных каналов обработки данных

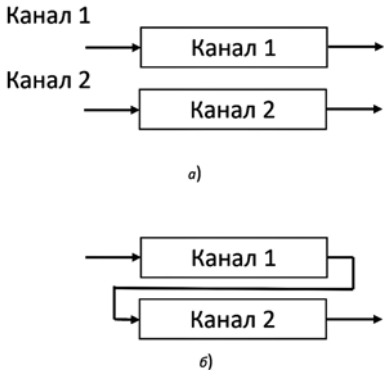


Рис. 6. Варианты использования смежных каналов:
а — многоканальная фильтрация; *б* — объединение каналов для образования фильтра более высокого порядка

деление каналов на тактовые регионы с передачей данных через FIFO обеспечивает последовательную потоковую обработку для любого практически реализуемого показателя латентности. Поэтому коммутаторы, обеспечивающие передачу данных между каналами, могут быть выполнены по конвейеризованной схеме с внедрением дополнительных уровней регистров, уменьшающих задержку распространения сигналов до приемлемых значений. На практике можно рассматривать кластеры, содержащие 8...32 канала обработки данных, что дает верхнюю оценку сложности коммутаторов как 8-в-1...32-в-1. Если учитывать возможную конвейеризацию данных внутри таких коммутаторов, их технические характеристики становятся приемлемыми. Действительно, прямая реализация мультиплексоров подразумевает последовательное соединение не более $\log2(N)$ вентилей, т. е. 3—5 вентилей для мультиплексоров 8-в-1...32-в-1. Такая схема внесет несущественную задержку по сравнению с вычислительными устройствами. Кроме того, сам мультиплексор может быть конвейеризован.

Дополнительно имеется возможность на основе данных предварительного моделирования ограничить функциональные возможности межканальных коммутаторов, обеспечив в минимальном варианте всего две функции: повторный запуск полученных данных на том же канале (например, для увеличения порядка КИХ-фильтра) и запуск данных соседнего канала (для вычисления взаимной корреляции, например, при работе соседнего канала в качестве генератора коэффициентов гармонического ряда).

Приведем ряд архитектурных особенностей, соответствующих современным тенденциям в области цифровой микроэлектроники, которые позволяют упростить реализацию рассмотренной СБИС и снизить риски получения ухудшенных технических характеристик вследствие факторов, выявляемых на этапе топологического проектирования.

1. Распределенная память для управления встроенными мультиплексорами обеспечивает топологическое разрежение компонентов, что является ответом на проблему "темного кремния". При необходимости уменьшения удельного тепловыделения размер памяти может быть увеличен, что снизит значение переключательной активности СБИС на единицу площади.

2. Каналы обработки данных являются независимыми, содержат только конвейерные схемы, которые не формируют длинные связи при объединении. Это соответствует архитектурному подходу GALS и предполагает хорошо локализуемые тактовые сети. Таким образом, увеличение площади СБИС не образует дополнительные трассировочные линии с увеличенной задержкой.

Оценка характеристик СБИС с конфигурируемым конвейером

Оценка проекта СБИС была выполнена на базе двух платформ, а именно — ПЛИС с архитектурой FPGA и тестовых библиотек с нормами 28 нм. Поскольку ячейки FPGA имеют другое соотношение основных ресурсов по сравнению со СБИС, синтез проекта в этом базисе преследовал цель проверки концепции.

В результате синтеза базовой ячейки, выполняющей два шага "умножение с накоплением" для 32-разрядных данных, объем составил 360...400 логических ячеек FPGA Xilinx серии 7 (с 6-входовыми логическими генераторами). Можно отметить, что соотношение логических генераторов к аппаратным умножителям для этой серии FPGA составляет приблизительно 250. Следовательно, компонент, выполняющий два шага умножения, менее эффективно использует ресурсы FPGA по сравнению с аппаратным умножителем операндов из 18 и 25 бит (для FPGA Xilinx серии 7). Таким образом, можно констатировать, что распределенная арифметика не является оптимальным решением для FPGA, выступая в лучшем случае в качестве дополнения для аппаратных компонентов "умножение с накоплением" (DSP48 для FPGA Xilinx).

В то же время оценка аналогичного компонента в технологическом базисе 28 нм HPC + показала, что

при тактовой частоте 800...1200 МГц аналогичный узел занимает 1200...1500 мкм^2, что с учетом размещения необходимой памяти позволяет прогнозировать суммарную площадь такого узла 5000 мкм^2. Это означает размещение на кристалле СБИС свыше 200 узлов распределенной арифметики на 1 мм^2.

Рассмотренная архитектура в целом соответствует основным тенденциям в области проектирования СБИС. Среди возможных проблемных вопросов, выявляемых на этапе топологического проектирования (т. е. на поздних стадиях разработки, делающих проблематичными существенные архитектурные изменения), выше были отмечены высокое локальное потребление энергии и трудность трассировки тактовых сетей по большой площади кристалла. Дополнительно можно обозначить сложности общего характера, которые возникают при трассировке сигналов при наличии высокой связности отдельных модулей проекта.

В архитектуре СБИС эти вопросы разрешаются следующим образом. Прежде всего, логическое разделение СБИС на отдельные модули позволяет соответствующим образом локализовать тактовые регионы. Далее, распределенный характер расположения модулей памяти обеспечит прореживание компонентов с повышенным тепловыделением, приводя усредненные показатели к приемлемому уровню.

Использование распределенной арифметики позволяет применить при реализации КИХ-фильтра прием, основанный на выполнении только тех шагов умножения, для которых установлены ненулевые разряды в соответствующих позициях коэффициентов фильтра. С учетом специфики импульсных характеристик многих фильтров, имеющих большой динамический диапазон, после исследования на ряде примеров было установлено, что для реализации фильтра достаточно выполнить около 30 % шагов. Поэтому производительность 200 узлов, выполняющих по два шага, на частоте 1 ГГц эквивалентна приблизительно 80 ГУН/с/мм^2 (80 GMAC/s/mm^2) для 16-разрядных коэффициентов.

Можно указать, что производительность КИХ-фильтров для FPGA Xilinx UltraScale+ (16 nm FinFET) достигает значения 5 ТУН/с (5 TMAC/s) [14]. Это свидетельствует о реальной возможности разработки СБИС, ориентированной на потоковые операции умножения с накоплением. Такая архитектура может являться альтернативой или дополнением высокопроизводительным FPGA.

Заключение

Анализ, проведенный в статье, показывает возможность создания проблемно-ориентированной СБИС, по крайней мере для одного семейства задач, использующих схожие по структуре вычислительные узлы. Представление архитектуры СБИС открывает возможность исследования эффективности реализации других семейств задач и уточнения архитектурных параметров для их реализации. Такой подход позволяет проектировать СБИС, объединяющие смежные области применения, что актуально с учетом возрастающей стоимости подготовки производства и выпуска тестовых полупроводниковых пластин при переходе к технологическим процессам с меньшими нормами.

Список литературы

1. **Oliveira D. L., Faria L. A., Lussari E.** FPGA implementation of robust asynchronous wrappers for Globally-Asynchronous systems (GALS)//2012 VIII Southern Conference on Programmable Logic, Bento Goncalves, 2012. — P. 1—6. DOI: 10.1109/SPL.2012.6211772.

2. **Farouk H. A., El-Hadidi M. T.** Implementing Globally Asynchronous Locally Synchronous processor pipeline on commercial synchronous FPGAs // 2010 17th International Conference on Telecommunications, Doha, 2010. — P. 989-994. DOI: 10.1109/ICTEL.2010.5478856.

3. **Oliveira D. L., Lussari E.** Synthesis of robust controllers for GALS_FPGA from multi-burst graph specification // 2011 VII Southern Conference on Programmable Logic (SPL), Cordoba, 2011. — P. 123—129. DOI: 10.1109/SPL.2011.5782636.

4. **UltraScale** Architecture Clocking Resources User Guide UG572 (v1.9) October 31, 2019. URL: https://www.xilinx.com/support/documentation/user_guides/ug572-ultrascale-clocking.pdf.

5. **Корнеев В. В., Тарасов И. Е.** Особенности архитектуры массово-параллельных проблемно-ориентированных СБИС. Москва//Программная инженерия. — 2019. — Том 10, № 4. — С. 160—166.

6. **Wenyan Lu, Guihai Yan, Jiajun Li, Shijun Gong, Yinhe Han, Xiaowei Li.** FlexFlow: A Flexible Dataflow Accelerator Architecture for Convolutional Neural Networks// In Proceedings of the 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA.2017). 2017. — P. 553—564.

7. **Moore S. K.** 6 Things to Know About the Biggest Chip Ever Built// IEEE Spectrum. 21 Aug 2019. URL: https://spectrum.ieee.org/tech-talk/semiconductors/processors/4-things-to-know-about-the-biggest-chip-ever-built

8. **Jia Z., Tillman B., Maggioni M., Scarpazza D. P.** Dissecting the Graphcore IPU Architecture via Microbenchmarking. Technical Report. December 7, 2019. arXiv:1912.03413v1 [cs.DC] 7 Dec 2019.

9. **Torii S., Ishikawa H.** ZettaScaler: Liquid immersion cooling Manycore based Supercomputer // ISC 2017. — JUNE 18—22, 2017. Frankfurt am Main, Germany. URL: https://is-candar.org/candar17/keynotes

10. **Kanduri A. Rahmani A., Liljeberg P.** et al. A Perspective on Dark Silicon//The Dark Side of Silicon. Springer, Cham. 2017. — P. 3—20. DOI: 10.1007/978-3-319-31596-6_1.

11. **Volder J. E.** The CORDIC Trigonometric Computing Technique// IRE Transactions on Electronic Computers. — Sept. 1959. — Vol. EC-8, No. 3. — P. 330—334. DOI: 10.1109/TEC.1959.5222693.

12. **Hu Y. H.** CORDIC-based VLSI architectures for digital signal processing// IEEE Signal Processing Magazine. — July 1992. — Vol. 9, No. 3. — P. 16— 35. DOI: 10.1109/79.143467.

13. **Angarita F., Canet M. J., Sansaloni T.** et al. Efficient Mapping of CORDIC Algorithm for OFDM-Based WLAN //J Sign Process Syst Sign Image Video Technol. — 2008. — Vol. 52. — Article 181. URL: https://doi.org/10.1007/s11265-007-0146-6

14. **Mehta N.** UltraScale Architecture: Highest Device Utilization, Performance, and Scalability. URL: https://www.xilinx.com/support/documentation/white_papers/wp455-utilization.pdf

VLSI Architecture with a Configurable Pipeline

V. V. Korneev, korv@rdi-kvant.ru, **I. E. Tarasov,** ilya_e_tarasov@mail.ru,
Research and Development Institute "Kvant",
Moscow, 125438, Russian Federation

Corresponding author:

Korneev Victor V., Principal Researcher, Research and Development Institute "Kvant", Moscow, 125438, Russian Federation
E-mail: korv@rdi-kvant.ru

Received on July 12, 2020
Accepted on July 30, 2020

The analysis carried out in the article shows the possibility of creating a problem-oriented VLSI, fabricated according to the technological standards of 28 nm or less, for at least one family of digital signal processing problems using similar computing nodes in structure. The use of distributed arithmetic allows one to apply a technique based on performing only those multiplication steps for which non-zero digits are set in the corresponding positions of the filter coefficients. Therefore, the performance of 200 nodes executing 2 steps at 1 GHz is equivalent to approximately 80 GMAC/s / mm^2 for 16-bit coefficients. The VLSI architecture view opens up the possibility to study the effectiveness of implementing other families of tasks and refine the architectural parameters for their implementation. The proposed functionality of VLSI computing nodes allows them to be used in various fields of technology, which potentially increases the need for the release of such VLSI.

Keywords: *VLSI, architecture, digital signal processing*

For citation:

Korneev V. V., Tarasov I. E. VLSI Architecture with a Configurable Pipeline, *Programmnaya Ingeneria,* 2020, vol. 11, no. 5, pp. 270—276

DOI: 10.17587/prin.11.270-276

References

1. **Oliveira D. L., Faria L. A., Lussari E.** FPGA implementation of robust asynchronous wrappers for Globally-Asynchronous systems (GALS), *2012 VIII Southern Conference on Programmable Logic, Bento Goncalves*, 2012, pp. 1—6, DOI: 10.1109/SPL.2012.6211772.

2. **Farouk H. A., El-Hadidi M. T.** Implementing Globally Asynchronous Locally Synchronous processor pipeline on commercial synchronous FPGAs, *2010 17th International Conference on Telecommunications*, Doha, 2010, pp. 989—994, DOI: 10.1109/ICTEL.2010.5478856.

3. **Oliveira D. L., Lussari E.** Synthesis of robust controllers for GALS_FPGA from multi-burst graph specification, *2011 VII Southern Conference on Programmable Logic (SPL)*, Cordoba, 2011, pp. 123—129, DOI: 10.1109/SPL.2011.5782636.

4. **UltraScale** Architecture Clocking Resources User Guide UG572 (v1.9) October 31, 2019, available at: https://www.xilinx.com/support/documentation/user_guides/ug572-ultrascale-clocking.pdf

5. **Korneev V. V., Tarasov I. E.** Peculiar Properties of Architecture of Parallel Application Specific Integral Circuit, *Programmnaya Ingeneria,* 2019, vol. 10, no. 4, pp. 160—166 (in Russian).

6. **Wenyan Lu, Guihai Yan, Jiajun Li, Shijun Gong, Yinhe Han, Xiaowei Li.** FlexFlow: A Flexible Dataow Accelerator Architecture for Convolutional Neural Networks, *In Proceedings of the 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA.2017),* 2017, pp. 553—564.

7. **Moore S. K.** 6 Things to Know About the Biggest Chip Ever Built, *IEEE Spectrum,* 21 Aug. 2019, available at: https://spectrum.ieee.org/tech-talk/semiconductors/processors/4-things-to-know-about-the-biggest-chip-ever-built

8. **Jia Z., Tillman B., Maggioni M., Scarpazza D. P.** Dissecting the Graphcore IPU Architecture via Microbenchmarking. Technical Report. December 7, 2019. arXiv:1912.03413v1 [cs.DC] 7 Dec. 2019.

9. **Torii S., Ishikawa H.** ZettaScaler: Liquid immersion cooling Manycore based Supercomputer, *ISC 2017.* June 18—22, 2017 Frankfurt Am Main, Germany, available at: https://is-candar.org/candar17/keynotes

10. **Kanduri A., Rahmani A. M., Liljeberg P., Hemani A., Jantsch A., Tenhunen H.** A Perspective on Dark Silicon, *The Dark Side of Silicon.* Springer, Cham. 2017. P. 3—20. DOI: 10.1007/978-3-319-31596-6_1.

11. **Volder J. E.** The CORDIC Trigonometric Computing Technique, *IRE Transactions on Electronic Computers*, Sept. 1959, vol. EC-8, no. 3, pp. 330—334, DOI: 10.1109/TEC.1959.5222693.

12. **Hu Y. H.** CORDIC-based VLSI architectures for digital signal processing, *IEEE Signal Processing Magazine*, July 1992, vol. 9, no. 3, pp. 16—35, DOI: 10.1109/79.143467.

13. **Angarita F., Canet M. J., Sansaloni T.** et al. Efficient Mapping of CORDIC Algorithm for OFDM-Based WLAN, *J Sign Process Syst Sign Image Video Technol*, 2008, vol. 52, article 181, available at: https://doi.org/10.1007/s11265-007-0146-6.

14. **Mehta N.** UltraScale Architecture: Highest Device Utilization, Performance, and Scalability, available at: https://www.xilinx.com/support/documentation/white_papers/wp455-utilization.pdf

УДК 519.724.2 DOI: 10.17587/prin.11.277-284

И. Б. Казаков, аспирант, i_b_kazakov@mail.ru, Московский государственный университет им. М. В. Ломоносова

Передача информации в каналах, задаваемых структурами частичного стирания. Часть 1

Представлены результаты исследований проблемы построения скрытых каналов передачи данных, в которых информация передается движением по игровому полю. Введен источник помех, порожденных попаданием игрока-передатчика в зоны невидимости (например, вследствие сетевых сбоев или аспектов игровой механики), и рассмотрена задача надежной передачи информации через зашумленный канал. Формальная модель, отражающая содержательную постановку, названа структурой частичного стирания. В рамках модели доказан критерий эквивалентности пары траекторий с точки зрения приемника информации. Представлено обсуждение методов борьбы с организацией скрытых каналов рассматриваемого типа.

Ключевые слова: *скрытые каналы, блуждания по плоскости, блуждания в $\mathbb{Z}^k$, структура частичного стирания*

Введение

В статье представлены результаты цикла исследований автора, посвященных теории скрытых каналов. Напомним основные определения и термины этого направления. Скрытым каналом называется коммуникационный канал, передающий информацию методом, который изначально не был для этого предназначен. Это означает, что существуют злоумышленники, передающие информацию таким образом, что сам факт передачи не может быть зафиксирован сторонними наблюдателями. Исторически первой работой, где было введено само понятие скрытого канала, считается работа [1]. Современный краткий обзор можно найти в работах [2, 3], также следует упомянуть работы отечественных исследователей [4, 5].

Статья состоит из двух частей. В части 1 речь идет о скрытом канале блужданий по плоскости. Блуждания по плоскости изучают в связи с задачей построения скрытых каналов через многопользовательские online-игры. В рассматриваемом классе игр имеется плоскость, на которой расположены игровые сущности. Сервер хранит местоположения этих сущностей на плоскости, т. е. их координаты. Предполагается также, что у каждого игрока, т. е. подключенного к серверу клиента, имеется сущность, поведением которой он может управлять. Как правило, данная сущность называется игровым персонажем. Один из клиентов передает серверу команды о перемещении управляемого им персонажа по плоскости, а другие клиенты могут получать от сервера данные о местоположении (и, следовательно, также об изменениях местоположения) данного игрового персонажа.

Задача построения скрытых каналов через online-шутеры ранее исследовалась, например, в работе [6], где описано как информация передавалась посредством внесения малых отклонений в исходную траекторию игрока. Под online-играми могут пониматься не только "шутеры от первого лица", но также и игры с произвольной механикой. Упомянем, что построению скрытого канала в отмеченном общем случае посвящена, например, работа [7].

В настоящей работе канал передачи данных через блуждания по плоскости построен следующим образом. Вся игровая плоскость разбита на многоугольники. Игрок, который хочет передать скрытое сообщение, должен находиться в области видимости игрока, которому он хочет передать данное сообщение. Следуя сложившейся традиции, будем называть передающего субъекта Алисой, а принимающего — Бобом.

Актом передачи информации считается переход в смежный многоугольник. Таким образом, на каждом тактовом акте могут быть переданы ровно столько значений, сколько смежных многоугольников имеется у того многоугольника, в котором игрок находится в настоящий момент. Такой подход уже представлялся в работе автора [8]. В этой работе был рассмотрен вопрос, связанный с тем, что движение по игровой плоскости не может быть абсолютно свободным: в реальных online-играх на игровом поле всегда имеются многочисленные препятствия, т. е. области, внутри которых игроки не могут находиться.

Однако в большинстве известных автору работ (например, [6, 7]), а также в работе [8] предположено, что каждый клиент получает от сервера информацию обо всех последовательных состояниях игры. Применительно к блужданиям по плоскости это означает получение сведений о местоположении всех других (точнее, находящихся в зоне видимости) игроков на плоскости во все моменты времени, которое принимается состоящим из последовательных тактов.

Сформулированное выше предположение в реальных условиях не соблюдается в силу переменной задержки между отправкой данных сервером и их получением клиентом, а также потому что сервер не обязан отправлять их на каждом такте. Таким образом, часть информации теряется, и клиент получает сведения о местоположениях прочих игроков лишь в некоторые моменты времени. Как следствие, возникает задача построения скрытого канала именно в условиях потери и/или зашумления информации. Данная статья продолжает цикл работ автора [9, 10], посвященных решению этой задачи.

В целях лучшего понимания структуры материала статьи, анонсируем полученные результаты. Прежде всего, введено новое понятие — "структура частичного стирания", отражающее упомянутые выше потери информации. Блуждания по плоскости сведены к блужданиям в множестве векторов $\mathbb{Z}^k$. Данное сведение позволяет решить задачу поиска моментов пересечения траекторий. Сформулированы и доказаны соответствующие утверждения.

Представим структуру изложения материала. Понятию "структура частичного стирания" посвящен разд. 1. В разд. 2 представлено сведение к блужданиям в $\mathbb{Z}^k$, а также алгоритм проверки тождества концевых вершин траекторий. Разд. 3 посвящен обсуждению методов борьбы с организацией рассматриваемого в настоящей работе скрытого канала.

1. Основные определения

Данный раздел посвящен понятию "структура частичного стирания". В подразд. 1.1 представлены соображения, относящиеся к траекториям движения Алисы, а также моментам времени, в которые Боб наблюдает ее местоположение. Далее, в подразд. 1.2 проведем соответствующее абстрагирование, отождествив траектории с символами некоего алфавита A, а подмножества моментов времени наблюдения — с определенными на нем разбиениями, тем самым определив структуру частичного стирания на траекториях.

1.1. Траектории движения Алисы

Прежде всего, дадим пояснения, более точно описывающие принятую в настоящей работе модель движения Алисы.

Во-первых, игровое время принимается дискретным, т. е. состоящим из мельчайших тактов. Во-вторых, Алиса всегда движется с постоянной скоростью, причем может менять каждый такт направления своего движения. Принимается также, что Алиса может двигаться лишь в конечном числе направлений, отстоящих друг от друга на равные углы. Отсюда очевидным образом следует, что множество точек плоскости, в которых она может побывать, счетно. На этом множестве точек определено отношение смежности, означающее, из какой точки в какую Алиса может перейти за один такт. Таким образом, движение по плоскости сводится к движению по графу. Далее будем называть его графом местоположений Алисы и обозначать как G_{loc}.

Если Алиса за один такт может перейти из точки v_1 в точку v_2, то естественно предположить, что она также может совершить данное движение в обратном направлении. Поэтому если Алиса может сделать шаг в некотором направлении, то она может сделать его и в противоположном. Следовательно, число направлений, по которым может двигаться Алиса, или, что то же самое, степени вершин G_{loc}, всегда четно.

В-третьих, зафиксировано N — общее число моментов, на которых Боб рассматривает движение Алисы. Зафиксируем также некую начальную вершину графа. Таким образом, за алфавит A пересылаемых символов можно принять множество возможных траекторий Алисы, т. е. путей длины N в графе G_{loc}. Траекторию, взятую в качестве пути графа, можно рассматривать как функцию $a : \{1, 2, 3...N\} \to G_{loc}$ таковую, что для всех $t = 1, 2, ..., N-1$, $a(t)$ и $a(t+1)$ являются смежными вершинами.

Однако Боб видит местоположение Алисы не во все моменты времени из $\{1, 2, ..., N\}$, а только в некоторые из них. Множество таковых моментов обозначается как $T \subset \{1, 2, 3, ..., N\}$.

Каждому T соответствует разбиение, определенное на множестве траекторий A. Будем констатировать, что траектории a_1, a_2 эквивалентны по множеству моментов T, если в каждый из моментов T данные траектории совпадают. Зададим соответствующее определение.

Определение 1.1. Справедливо отношение $a_1 \sim^T a_2$, если $\forall t \in T,\ a_1(t) = a_2(t)$.

Для каждого возможного множества моментов наблюдения T определена вероятность p_T того, что Боб зарегистрирует местонахождение Алисы именно в данные моменты. Поскольку множество T задает некое разбиение, то далее будем считать, что p_T приписана этому разбиению. Вероятности p_T зависят от свойств величины межкадровых интервалов, т. е. промежутков времени между моментами, в которые сервер высылает Бобу текущее местоположение Алисы. В настоящей работе вопросы, связанные с моделями межкадровых интервалов, рассматриваться не будут, поскольку они не относятся непосредственно к изначально заявленному предмету изучения, так как для всех дальнейших рассуждений о структурах частичного стирания конкретные значения вероятностей безразличны. Как следствие, читатель может считать их взятыми произвольно. Отдельно отметим, что если для какого-то T выполнено $p_T = 0$, то данное разбиение T возможно исключить из рассмотрения.

Таким образом, имеются: множество траекторий, далее называемое первичным алфавитом A, множество определенных на данном алфавите разбиений $\mathfrak{T}$, а также приписанные данным разбиениям веса-вероятности p_T.

1.2. Структура частичного стирания

С учетом представленных выше определений, можно абстрагироваться от самих траекторий и представить формальное определение структуры частичного стирания.

Определение 1.2. Структурой частичного стирания называется тройка $\left(A, \mathfrak{T}, \{p_T\}_{T\in\mathfrak{T}}\right)$, где A — это некий алфавит; $\mathfrak{T}$ — множество, состоящее из разбиений (или, что то же самое, отношений эквивалентности) алфавита A; $\{p_T\}_{T\in\mathfrak{T}}$ — веса-вероятности,

приписанные указанным разбиениям. Общая сумма данных весов, поскольку они также названы вероятностями, принимается равной 1.

Поясним далее смысл приведенной выше конструкции. Периодически, через определенный интервал времени Алиса отправляет по каналу передачи некий символ a из алфавита A. Однако Боб не получает самого отправленного символа a, а получает лишь часть информации о том, какой символ алфавита A был отправлен ему Алисой.

Опишем, что именно означает факт получения Бобом частичной информации о символе $a \in A$. Прежде всего отметим, что ответ, полученный Бобом, определяется вероятностным, а не детерминированным образом. Каждому разбиению из структуры частичного стирания соответствует событие "было выбрано именно данное разбиение". Событие, соответствующее разбиению T, далее будем обозначать как X_T. Класс эквивалентности по разбиению T, которому принадлежит некое $a \in A$, обозначается как $\pi_T(a)$. Данные события — несовместны, а их вероятности равны приписанным весам соответствующих разбиений. Таким образом, происходит одно и только одно из таких событий.

Пусть произошло событие X_T. Тогда полагаем, что Боб получает об отправленном Алисой символе $a \in A$ следующую информацию: во-первых, какое именно событие X_T произошло, а во-вторых, какому классу эквивалентности $\pi_T(a)$ принадлежал символ a. Таким образом, полученные Бобом сведения можно представить как пару $(\pi_T(a), T)$.

В дальнейшем множество пар вида $(\pi_T(a), T)$ для всех символов $a \in A$ и всех разбиений T, взятых из структуры частичного стирания, будет обозначаться как алфавит Боба B.

Между символами алфавита A и символами алфавита B имеется естественным образом определяемое отношение, выражающее тот факт, что символ $b \in B$ может быть получен посредством утраты части информации о символе $a \in A$. Определим данное отношение формальным образом.

Определение 1.3. Справедливо отношение $a \mapsto b$, $a \in A$, $b \in B$, если $b = (\pi_T(a'), T)$ и $a \in \pi_T(a')$.

Задача состоит в том, чтобы в описанных выше условиях организовать передачу информации от Алисы к Бобу, причем способ передачи по возможности должен быть наиболее эффективным.

Данное отношение естественным образом обобщается и на слова из множеств A^*, B^*.

Определение 1.4. Справедливо отношение $\alpha \mapsto \beta$, $\alpha \in A^*$, $\beta \in B^*$, если $|\alpha| = |\beta| = k$ и $\forall i = 1...k$, $\alpha(i) \mapsto \beta(i)$.

Утверждение 1.1. Для любого $\alpha \in A^*$ существует $\beta \in B^*$ такое, что $\alpha \mapsto \beta$.

Действительно, выберем в структуре частичного стирания произвольные разбиения $T_1, T_2, T_3, ..., T_k$ и для всех $i = 1...k$ положим $b_i = (\pi_{T_i}(\alpha(i)), T_i)$. Тогда $\alpha(1)...\alpha(k) = \alpha \mapsto \beta = b_1...b_k$.

2. Вложение траекторий в $\mathbb{Z}^k$

В данном разделе изучается ранее представленный граф возможных местоположений G_{loc}. Покажем, что в силу изоморфизма движение по этому графу сводится к движению по пространству $\mathbb{Z}^k$. Во вводном подразд. 2.1 представим задачу проверки тождества концов траекторий, решению которой будут посвящены подразд. 2.2—2.5.

2.1. Постановка задачи

В силу характера отмеченного выше абстрагирования необходимо уметь для двух произвольных траекторий $a_1(t)$, $a_2(t)$ определять моменты их пересечения. Поясним, что под "моментами пересечения" имеются в виду такие t, для которых $a_1(t) = a_2(t)$. Обозначим число возможных направлений движения как j, т. е. степень каждой из вершин графа G_{loc} равна j.

Напомним принятые допущения. Под траекторией понимается слово в алфавите $E_j = \{0, 1, ..., j-1\}$, где j — четное число. Каждая буква данного слова задает выбор направления движения Алисы: если очередная буква есть число $n \in E_j$, то направление движения образует с вещественной осью угол, равный $\frac{2\pi n}{j}$.

Задача нахождения моментов пересечения очевидным образом сводится к задаче проверки тождества концов траекторий. Действительно, для каждого $i \in \{1, 2, ..., N\}$ выделим подтраектории-префиксы длины i исходных траекторий, а затем проверим, не совпадают ли концы выделенных подтраекторий.

Рассмотрим в качестве примера частный случай — $j = 4$. Четыре направления — это хождение по клеточкам. Указанная выше задача в данном случае решается тривиально. Для однозначной идентификации конечной точки пути достаточно вычислить два числа: "горизонтальное смещение" и "вертикальное смещение". Под горизонтальным смещением подразумевается число "ходов вправо" минус число "ходов влево". Вертикальное — аналогично, это разность между числом ходов "вверх" и "вниз".

Требуется указать аналогичный алгоритмически проверяемый критерий однозначной идентификации местоположения для произвольно взятого четного числа направлений j.

Для более точного понимания материала опишем структуру дальнейшего его изложения в настоящем разделе. Прежде всего, в подразд. 2.2 введем некоторые вспомогательные понятия. После чего представим в подразд. 2.3 требуемый алгоритм и докажем его корректность в подразд. 2.4. В подразд. 2.5 будет представлена проводная процедура, проводящая расчет всех траекторий длины N.

2.2. Вспомогательные понятия

Представим вспомогательные понятия, а также необходимые для дальнейшего изложения утверждения. Некоторые из утверждений являются известными из курса алгебры фактами, их доказательства могут быть найдены, например, в работе [11].

Прежде всего, определим так называемый "круговой многочлен" и отметим его свойства. Полагаем: $z_0 = \cos\left(\frac{2\pi}{j}\right) + i\sin\left(\frac{2\pi}{j}\right)$ — примитивный корень; φ — функция Эйлера; $k = \varphi(j)$.

Определение 2.1. Круговым многочленом будем называть многочлен Φ_j, определяемый посредством тождества $\Phi_j(z) = \prod_{1 \leqslant l \leqslant j,\,(l,j)=1} \left(z - z_0^l\right)$.

Факт (свойства кругового многочлена).

1. $\Phi_j \in \mathbb{Z}[x]$.
2. Φ_j неприводим над $\mathbb{Z}$.
3. Старший коэффициент Φ_j равен 1.
4. $\Phi_j(z_0) = 0$.
5. $\deg(\Phi_j) = \varphi(j)$.
6. Многочлен $z^{\frac{j}{2}}+1$ делится на Φ_j.

Имеется также утверждение, связывающее между собой многочлены над кольцом $\mathbb{Z}$ с многочленами над полем $\mathbb{Q}$ и называемое леммой Гаусса.

Факт (лемма Гаусса).

1. Пусть $p = p_1 p_2$ для $p,\ p_1,\ p_2 \in \mathbb{Z}[x]$, а также некое простое число r делит все коэффициенты многочлена p. Тогда r также делит все коэффициенты или многочлена p_1, или многочлена p_2.

2. Если некий многочлен Φ неприводим в $\mathbb{Z}[x]$, то он неприводим и в $\mathbb{Q}[x]$.

Доказательство леммы Гаусса, а также целочисленности коэффициентов кругового многочлена и его неприводимости над кольцом целых чисел есть в работе [11]. Свойства 3—5 данного многочлена очевидны непосредственно из определения. Докажем свойство 6.

Утверждение 2.1. Многочлен $z^{\frac{j}{2}}+1$ делится на Φ_j.

Доказательство.

1. Так как Φ_j не имеет кратных корней, то для доказательства требуемого свойства достаточно показать, что каждый корень Φ_j является также и корнем многочлена $z^{\frac{j}{2}}+1$.

2. Действительно, пусть z_0^l — некий корень кругового многочлена, т. е. НОД(l, j) = 1. Так как j четно, то l — нечетно.

3. Тогда $z_0^{\frac{jl}{2}} = \left(z_0^{\frac{j}{2}}\right)^l = (-1)^l = -1$.

Что и требовалось доказать.

Определим также базовые векторы $\mathbf{v}_0, \mathbf{v}_1,\ ...,\ \mathbf{v}_{j-1} \in \mathbb{Z}^k$, соответствующие остаткам от деления одночленов $1,\ z,\ ...,\ z^{j-1}$ на многочлен Φ_j. Под указанным соответствием подразумевается биекция, соотносящая многочлен $a_0 + a_1 z + a_2 z^2 + ... + a_{k-1} z^{k-1} \in \mathbb{Z}[x]$ с вектором $(a_0,\ ...,\ a_{k-1}) \in \mathbb{Z}^k$.

Относительно множества базовых векторов сразу следует отметить следующее его свойство.

Утверждение 2.2. Пусть некий вектор $\mathbf{w} \in \{\mathbf{v}_0,\ ...,\ \mathbf{v}_{j-1}\}$. Тогда также $-\mathbf{w} \in \{\mathbf{v}_0,\ ...,\ \mathbf{v}_{j-1}\}$.

Доказательство.

1. Достаточно доказать, что $\mathbf{v}_l = -\mathbf{v}_{l+\frac{j}{2}}$ для всех $l = 0 ... \frac{j}{2} - 1$.

2. Требуемое очевидным образом следует из того, что многочлен $z^l + z^{l+\frac{j}{2}} = z^l\left(z^{\frac{j}{2}}+1\right)$ делится на Φ_j. В свою очередь, это выполнено, так как на Φ_j делится многочлен $z^{\frac{j}{2}}+1$.

Что и требовалось доказать.

Базовые векторы $\mathbf{v}_0,\ ...,\ \mathbf{v}_{j-1}$ определяют на $\mathbb{Z}^k$ отношение смежности, позволяющее рассматривать данное множество как граф.

Определение 2.2. Два вектора $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{Z}^k$ полагаются смежными, тогда и только тогда, когда $\mathbf{w}_1 - \mathbf{w}_2 \in \{\mathbf{v}_0, \mathbf{v}_1,\ ...,\ \mathbf{v}_{j-1}\}$.

Замечание. В силу предыдущего утверждения данное отношение симметрично, т. е. граф является неориентированным.

2.3. Проверка тождества концов траекторий

Алгоритм принимает на вход пару траекторий, т. е. слов из множества E_j^*. Конечным результатом работы является ответ вида "да/нет" на вопрос "совпадают ли концы данных траекторий".

Алгоритм

Шаг 1. (предвычисления) Рассчитать (или посмотреть в справочной литературе, например, в работе [11]) круговой многочлен Φ_j. Также следует провести подсчет остатков от деления одночленов $1,\ z,\ ...,\ z^{j-1}$ на Φ_j, тем самым рассчитав вектора $\mathbf{v}_0,\ ...,\ \mathbf{v}_{j-1}$. Данные векторы следует записать в соответствующий массив.

Шаг 2. (проверка) Пусть дана траектория $i_1 i_2 ... i_{N'} \in E_j^*$. Тогда следует рассчитать местоположение ее конца $\mathbf{v}_{end} = \mathbf{v}_{i_1} + ... + \mathbf{v}_{i_{N'}}$, просуммировав в цикле соответствующие базовые векторы $\mathbf{v}_{i_1}\ ...,\ \mathbf{v}_{i_{N'}}$. Указанный расчет следует провести для обеих траекторий, принятых на вход, тем самым рассчитать их концы $\mathbf{v}_{end}^1,\ \mathbf{v}_{end}^2$.

Если оказалось, что $\mathbf{v}_{end}^1 = \mathbf{v}_{end}^2$, то следует вернуть ответ "да". В противном случае вернуть ответ "нет".

Теорема 2.1. Представленный алгоритм корректен, т. е. он возвращает ответ "да" тогда и только тогда, когда совпадают концы траекторий, подаваемых ему на вход.

Замечание. При $j = 4$ получаем $\Phi_4 = z^2 + 1$. Соответствующие остатки $1,\ z,\ -1,\ -z$, т. е. $\mathbf{v}_1 = (1, 0)$, $\mathbf{v}_2 = (0, 1)$, $\mathbf{v}_3 = (-1, 0)$, $\mathbf{v}_4 = (0, -1)$. Таким образом, при $j = 4$ критерий, использованный в данном алгоритме, в точности повторяет критерий в терминах смещений, ранее представленный в разд. 1.

Замечание. Очевидно, что сложность алгоритма пропорциональна сумме длин входных траекторий.

2.4. Доказательство корректности

Докажем сформулированную выше теорему 2.1. Прежде всего, установим координаты возможных местоположений Алисы в подразделе 2.4.1. Подразд. 2.4.2 посвящен решению вспомогательной задачи отыскания всех целочисленных многочленов, которые имеют корень $z_0 = \cos\left(\frac{2\pi}{j}\right) + i\sin\left(\frac{2\pi}{j}\right)$. В подразд. 2.4.2 представлено окончание доказательства теоремы 2.1, использующее ранее полученные результаты. В подразд. 2.4.3 представлен изоморфизм графов G_{loc} и $\mathbb{Z}^k$. Существование такого изоморфизма является важным дополнительным результатом исследования.

2.4.1. Комплексные координаты

Отметим, что так как Алиса движется по плоскости, то в каждый момент времени ее местоположение может быть задано парой действительных чисел (x, y) или одним комплексным числом $z = x + iy$. Укажем комплексные координаты точек, в которые может прийти Алиса, т. е. комплексные координаты вершин графа G_{loc}.

Выскажем некоторые соображения, необходимые для формализованного описания поставленной задачи. Вершине, из которой Алиса начинает свое движение, приписывается координата $z = 0$. Считается, что каждый такт Алиса передвигается на вектор длины 1, т. е. к текущему положению Алисы, выражаемому координатой z, прибавляется некое комплексное число Δz, $|\Delta z| = 1$. Примем также, что одно из возможных направлений хода идет вдоль вещественной оси.

Сразу становится очевидным, что возможными Δz являются в точности корни уравнения $z^j = 1$. То есть $\Delta z \in \{1, z_0, z_0^2, \ldots, z_0^{j-1}\}$. Таким образом, координаты вершин графа G_{loc} определяются как произвольные суммы корней уравнения $z^j = 1$, причем вершины с координатами w_1 и w_2 смежны тогда и только тогда, когда $w_1 - w_2 \in \{1, z_0, z_0^2, \ldots, z_0^{j-1}\}$.

В соответствии с принятой в подразд. 2.1 интерпретацией траекторий, траектории $i_1 \ldots i_N \in E_j^*$ соответствует путь, последовательно проходящий через вершины с координатами $0, z_0^{i_1}, z_0^{i_1} + z_0^{i_2}, \ldots, z_0^{i_1} + \ldots + z_0^{i_N}$. Иначе выражаясь, комплексное число w, являющееся координатой некой вершины, является результатом подстановки корня z_0 в целочисленный многочлен: $w = p(z_0)$. Обратно, если подставить z_0 как аргумент в произвольный целочисленный многочлен, то получится комплексная координата какой-нибудь вершины. Как следствие, переходу в смежную вершину соответствует прибавление некоторого одночлена z^j.

Однако одной и той же вершине могут соответствовать более одного многочлена: $w = p_1(z_0) = p_2(z_0)$. Далее установим в явном виде для каждой вершины класс соответствующих ей целочисленных многочленов. Прежде всего заметим, что если $w = p_1(z_0) = p_2(z_0)$, то $(p_1 - p_2)(z_0) = 0$, т. е. z_0 — корень $p_1 - p_2$. Верно также и обратное, если z_0 — корень $p_1 - p_2$, то $p_1(z_0) = p_2(z_0)$.

2.4.2. Вспомогательные утверждения

Попытаемся отыскать все многочлены из $\mathbb{Z}[x]$, имеющие корень z_0.

Утверждение 2.3. Пусть $p_1, p_2 \in \mathbb{Q}[x]$ — многочлены с рациональными коэффициентами, имеющие общий комплексный корень z', т. е. $p_1(z') = p_2(z') = 0$. Тогда данные многочлены не могут быть взаимно-простыми, т. е. они имеют некий общий делитель, не являющийся константным многочленом.

Доказательство.

1. Предположим обратное. Тогда согласно расширенному алгоритму Евклида существуют многочлены $\alpha, \beta \in \mathbb{Q}[x]$ такие, что $\alpha p_1 + t\alpha p_2 = 1$.

2. Так как данное соотношение многочленов присутствует в $\mathbb{Q}[x]$, то оно присутствует также и в $\mathbb{C}[x]$.

3. Подставляя общий корень z', получаем $0 = \alpha(z')p_1(z') + \beta(z')p_2(z') = 1$. Приходим к противоречию.

Что и требовалось доказать.

В дальнейшем понадобится тот факт, что если целочисленный многочлен делится на другой целочисленный многочлен со старшим коэффициентом 1, то частное также является целочисленным многочленом. Докажем его.

Утверждение 2.4. Пусть для многочленов над полем рациональных чисел выполнено $p = p_1 p_2$, причем коэффициенты многочленов p, p_1 целочисленны, а старший коэффициент p_1 равен 1. Тогда коэффициенты p_2 также целочисленны.

Доказательство.

1. $p_2 \in \mathbb{Q}[x]$. Пусть a — минимальное натуральное число такое, что $ap_2 \in \mathbb{Z}[x]$. Примем следующие обозначения: $p_2' = ap_2$, $p' = ap$.

2. Домножим исходное условие на a. Получим: $p' = ap = p_1(ap_2) = p_1 p_2'$, $p_1, p_2' \in \mathbb{Z}[x]$.

3. Если $a = 1$, то все уже доказано. Поэтому считаем $a \neq 1$. Пусть теперь $r \in \mathbb{Z}$ — некий простой делитель a.

4. Тогда все коэффициенты p' делятся на r и, следовательно, согласно лемме Гаусса на r должны делиться или все коэффициенты p_1, или все коэффициенты p_2'.

5. Однако старший коэффициент многочлена p_1 не может делиться на r, поскольку равен 1. Следовательно, на r делятся все коэффициенты p_2'.

6. Как следствие, $\frac{a}{r} p_2 \in \mathbb{Z}[x]$, что противоречит выбору a.

Что и требовалось доказать.

Лемма 2.1. Пусть $p \in \mathbb{Z}[x]$, $p(z_0) = 0$. Тогда p делится на Φ_j в $\mathbb{Z}[x]$. Обратно: если p делится на Φ_j, то $p(z_0) = 0$.

Доказательство.

1. У многочленов p, Φ_j имеется общий корень z_0. Следовательно, если их рассматривать как многочлены над полем рациональных чисел, у них имеется общий делитель $\phi \in \mathbb{Q}[x]$, $\deg(\phi) > 0$.

2. Многочлен Φ_j неприводим над кольцом $\mathbb{Z}$. По лемме Гаусса данный многочлен неприводим и над полем $\mathbb{Q}$.

3. Так как ϕ — делитель Φ_j, Φ_j — неприводим и $\deg(\phi) > 0$, то $\Phi_j = \alpha\phi$, где $\alpha \in \mathbb{Q}$, $\alpha \neq 0$.

4. Также возможно записать: $p = \phi p'$, $p' \in \mathbb{Q}[x]$. Следовательно, $p = (\alpha^{-1} p')\Phi_j$.

5. $p, \Phi_j \in \mathbb{Z}[x]$, а старший коэффициент Φ_j равен 1. Согласно предыдущему утверждению из этого следует, что также $\alpha^{-1} p' \in \mathbb{Z}[x]$. Это и означает, что p делится на Φ_j в $\mathbb{Z}[x]$.

6. $\Phi_j(z_0) = 0$. Это означает, что выполнение утверждения леммы в обратную сторону очевидно.

Что и требовалось доказать.

Вывод. Два многочлена $p_1, p_2 \in \mathbb{Z}[x]$ соответствуют одной и той же вершине V_{loc} тогда и только тогда, когда их разность делится на круговой многочлен Φ_j.

2.4.3. Окончание доказательства

Утверждение 2.5. Представленный в подразд. 2.3 алгоритм корректен.

Доказательство.

1. Полагаем, что на вход алгоритма подаются две траектории $i_1^1 i_2^1 \ldots i_{N_1}^1$, $i_1^2 i_2^2 \ldots i_{N_2}^2 \in E_j^*$.

2. В соответствии с выводами подразд. 2.4.1, комплексные координаты концов данных траекторий $w_1 = z_0^{i_1^1} + \ldots + z_0^{i_{N_1}^1} = p_1(z_0)$ и $w_2 = z_0^{i_1^2} + \ldots + z_0^{i_{N_2}^2} = p_2(z_0)$.

3. Сославшись на лемму 2.1 установим, что $w_1 = w_2$ тогда и только тогда, когда многочлен $p' = p_1 - p_2$ делится на Φ_j.

4. Перепишем условие в виде $z^{i_1^1} + \ldots + z^{i_{N_1}^1} \equiv z^{i_1^2} + \ldots + z^{i_{N_2}^2} \pmod{\Phi_j}$. Оно, в свою очередь, равносильно $\left(z^{i_1^1} \bmod \Phi_j\right) + \ldots + \left(z^{i_{N_1}^1} \bmod \Phi_j\right) \equiv \left(z^{i_1^2} \bmod \Phi_j\right) + \ldots + \left(z^{i_{N_2}^2} \bmod \Phi_j\right)$.

5. Согласно отождествлению, описанному в подразд. 2.2, переходим от многочленов вида $z^l \bmod \Phi_j$ к векторам $\mathbf{v}_l$, получаем: $\mathbf{v}_{i_1^1} + \ldots + \mathbf{v}_{i_{N_1}^1} = \mathbf{v}_{i_1^2} + \ldots + \mathbf{v}_{i_{N_2}^2}$.

6. Таким образом, концевые вершины траекторий совпадают тогда и только тогда, когда выполнено $\mathbf{v}_{i_1^1} + \ldots + \mathbf{v}_{i_{N_1}^1} = \mathbf{v}_{i_1^2} + \ldots + \mathbf{v}_{i_{N_2}^2}$, а именно данное условие и проверяет алгоритм.

Таким образом, теорема 2.1 доказана.

2.4.4. Изоморфное вложение

Помимо изложенного выше представим также дополнительный результат. Прежде всего заметим, что вершины графа G_{loc} могут быть отождествлены с остатками от деления на Φ_j, т. е. поставлены данным остаткам во взаимно-однозначное соответствие: вершине с комплексной координатой $p(z_0)$ способствует многочлен $p \bmod \Phi_j$.

Остатки от деления на многочлен Φ_j степени k ничто иное как многочлены степени менее k. Эти многочлены, в свою очередь, отождествляются с векторами из $\mathbb{Z}^k$ описанным в разд. 1 образом.

Отдельно отметим, что данное отождествление является изоморфизмом абелевых групп, т. е. сохраняет аддитивные соотношения. Таким образом, имеется взаимно-однозначное соответствие между вершинами G_{loc} и векторами из пространства $\mathbb{Z}^k$, на котором было определено отношение смежности.

Обозначим композицию двух последовательных упомянутых отождествлений как $\psi : V_{loc} \to \mathbb{Z}^k$. Заметим, что как композиция двух биекций, ψ также является биекцией.

Теорема 2.2. Функция ψ является изоморфизмом графов G_{loc} и $\mathbb{Z}^k$.

Доказательство.

I. Сохранение смежности при прямом отображении.

1. Пусть даны две смежные вершины графа G_{loc} с координатами $w_1 = p_1(z_0)$, $w_2 = p_2(z_0)$, $p_1, p_2 \in \mathbb{Z}[x]$. В силу леммы 2.1 возможно, заменяя при необходимости p_1 на $p_1 \bmod \Phi_j$ и p_2 на $p_2 \bmod \Phi_j$, считать, что $\deg(p_1), \deg(p_2) < k$.

2. Смежность данных вершин означает, что $w_1 - w_2 = z_0^l$ для некоторого l, $0 \leqslant l \leqslant j-1$. Следовательно, $p_1(z_0) - p_2(z_0) = z_0^l$. Положим $p' = p_1 - p_2 - z^l$, таким образом, $p'(z_0) = 0$.

3. Согласно лемме 2.1 это означает, что $\left(p_1 - p_2 - z^l\right) \times \bmod \Phi_j = 0$. Откуда следует $p_1 - p_2 = z^l \bmod \Phi_j$.

4. Многочленам p_1, p_2 соответствуют некие векторы $\mathbf{v}'$, $\mathbf{v}'' \in \mathbb{Z}^k$, а многочлену $z^l \bmod \Phi_j$ — базовый вектор $\mathbf{v}_l$.

5. Как следствие, выполнено $\mathbf{v}' - \mathbf{v}'' = \mathbf{v}_l$, что и означает смежность $\mathbf{v}'$, $\mathbf{v}''$ в $\mathbb{Z}^k$.

II. Сохранение смежности при обратном отображении.

1. Обратно, пусть теперь $\mathbf{v}'$, $\mathbf{v}'' \in \mathbb{Z}^k$ — смежные векторы, т. е. для некоторого l, $0 \leqslant l \leqslant j-1$ выполнено $\mathbf{v}' - \mathbf{v}'' = \mathbf{v}_l$.

2. Полагаем, что векторам $\mathbf{v}'$, $\mathbf{v}''$ соответствуют многочлены p_1, p_2 степени менее k. Таким образом, $p_1 - p_2 = z^l \bmod \Phi_j$.

3. Вновь применяя лемму 2.1 (но уже в обратную сторону), получаем $p_1(z_0) - p_2(z_0) = z_0^l$.

4. Откуда следует смежность вершин G_{loc} с координатами $p_1(z_0)$ и $p_2(z_0)$.

Что и требовалось доказать.

Доказанная теорема позволяет абстрагироваться от блужданий по плоскости, заменив их блужданиями по пространству $\mathbb{Z}^k$. При этом траектория $i_1 \ldots i_N \in E_j^*$ может быть реинтерпретирована как последовательно проходящая через вершины 0, $\mathbf{v}_{i_1}$, $\mathbf{v}_{i_1} + \mathbf{v}_{i_2}$, ..., $\mathbf{v}_{i_1} + \ldots + \mathbf{v}_{i_N}$. Обоснуем данное отождествление.

Утверждение 2.6. Пусть вершина $w \in V_{loc}$ имеет комплексную координату $z_0^{i_1} + \ldots + z_0^{i_N}$. Тогда $\psi(w) = \mathbf{v}_{i_1} + \ldots + \mathbf{v}_{i_N}$.

Доказательство.

1. Условие утверждения означает также, что вершине w соответствует многочлен $p = z^{i_1} + \ldots + z^{i_N}$. Положим также $p' = p \bmod \Phi_j$.

2. Тогда можно записать: $p' = \left(z^{i_1} \bmod \Phi_j\right) + \ldots + \left(z^{i_N} \bmod \Phi_j\right)$. Каждому многочлену вида $z^l \bmod \Phi_j$ поставлен в соответствие базовый вектор $\mathbf{v}_l$. Данное соответствие сохраняет сложение. Следовательно, многочлену p' соответствует вектор $\mathbf{v}_{i_1} + \ldots + \mathbf{v}_{i_N}$.

3. Отсюда следует $\psi(w) = \mathbf{v}_{i_1} + \ldots + \mathbf{v}_{i_N}$.

Что и требовалось доказать.

С учетом представленного рассуждения утверждение теоремы 2.1 становится тавтологией. Тем самым, представлен альтернативный способ ее доказательства посредством вывода из более сильной теоремы 3.2.

2.5. Производная процедура

Представим теперь производную процедуру, сохраняющую в памяти информацию о моментах пересечения всех N^j траекторий длины N.

Алгоритм

Шаг 1 (предвычисления). Рассчитать круговой многочлен Φ_j, а также остатки от деления на него одночленов $1, z, \ldots, z^{j-1}$. Записать в массиве соответствующие векторы $\mathbf{v}_0, \ldots, \mathbf{v}_{j-1}$.

Шаг 2 (инициализация). Подготовить множество траекторий. Результатом работы данного шага является массив размера N^j, элементами которого являются все последовательности чисел от 0 до $j-1$ длины N, т. е. слова из E_j^N.

Шаг 3 (расчет местоположений). Возьмем очередной элемент полученного на предыдущем

шаге массива. Он представляет собой кортеж $(i_1, ..., i_N)$. Рассчитаем последовательные местоположения блужданий в $\mathbb{Z}^k$, т. е. векторные суммы $\mathbf{v}_{i_1}$, $\mathbf{v}_{i_1}+\mathbf{v}_{i_2}$,..., $\mathbf{v}_{i_1}+...+\mathbf{v}_{i_N}$. Таковой расчет следует осуществить для каждой траектории, т. е. N^j раз. Результатом работы данного шага является двумерный массив, имеющий N столбцов и N^j строк. Элементы данного массива — векторы из $\mathbb{Z}^k$.

Шаг 4. (абстрагирование). Для каждого столбца получившегося массива заменить векторы из $\mathbb{Z}^k$ на числа из $\mathbb{Z}$ таким образом, чтобы тождественные векторы заменялись на тождественные числа, а различные векторы — на различные числа.

В результате получаем двумерный целочисленный массив, т. е. таблицу, также имеющую N столбцов и N^j строк. Представленная процедура обсчитывает каждую из N^j траекторий за пропорциональное длине время $O(N)$. Следовательно, он имеет сложность $O(N^{j+1})$. Конечным результатом являются массив траекторий и итоговая таблица.

Предположим теперь, что требуется по двум траекториям определить моменты их пересечения. Для того чтобы это осуществить, сначала необходимо по массиву траекторий определить индексы соответствующих элементов, а далее найти совпадающие столбцы у соответствующих строк итоговой таблицы.

Таким образом, представлено полное решение задачи о моментах пересечения траекторий, что позволяет рассчитать соответствующую структуру частичного стирания.

3. О методах борьбы

В настоящей работе изучены блуждания по плоскости, посредством которых организуется скрытый канал. Однако в теории скрытых каналов рассматривается и решается не только задача организации канала злоумышленниками, но и задача защиты систем от подобного рода злоумышленников. Имеется в виду задача обнаружения факта наличия скрытого канала, а также задача предотвращения его построения.

Далее приведем некоторые соображения о мерах противодействия организации скрытого канала блужданий по плоскости. Прежде всего отметим, что использованный способ построения предполагает смену передающим игроком направления движения практически на каждом такте. Этот игрок будет также двигаться способом, напоминающим хаотическое "броуновское движение". Таковое движение является беспорядочным блужданием и не похоже на свойственное обыкновенным игрокам целенаправленное движение в определенном направлении.

С учетом отмеченного обстоятельства, если на стороне сервера, размещающего игровое поле и местоположения игроков, требуется определить, занимается ли кто-нибудь из них организацией скрытого канала, то следует измерить частоты смены направления движения соответственно для всех присутствующих игроков. Если некто имеет аномально высокую частоту, то этого игрока, очевидно, следует начать подозревать в организации скрытого канала. Вторым по значимости признаком являются околонулевые суммарные смещения за определенный промежуток времени.

Непосредственно очевидным способом предотвращения организации является запрет на слишком частую смену направлений. То есть сделать изменение направление движения возможным не на каждом такте, а раз в несколько тактов. Очевидно, что в соответствующее число раз уменьшится пропускная способность скрытого канала. Однако если разрешать смену направления слишком редко, то возникнут неудобства уже у обычных игроков. Следовательно, "замедлять" желательно только "подозреваемых".

Существенным недостатком предложенного метода противодействия является то, что он не предотвращает скрытую передачу информации, а только замедляет ее в соответствующее число раз. Поэтому изложим другую идею: выполнять команды подозреваемого игрока "принципиально неправильно". Поясним далее, что имеется в виду. Напомним, что управление движением игрока осуществляется посредством передачи неких команд от клиента к серверу.

Если сервер начал подозревать клиента в организации скрытого канала, то вместо ожидаемого изменения направления движения следует осуществлять таковое случайным образом. В данном режиме, устанавливаемом персонально для "подозрительного игрока", движение по плоскости не перестает быть хаотичным. Однако предположительный приемник вместо изначальной информации будет считывать шум.

Заключение

В части 1 настоящей статьи введено понятие структуры частичного стирания. Введение данной абстракции обосновано поставленной задачей передачи информации через блуждания на плоскости. Блуждания по плоскости сведены к блужданиям в $\mathbb{Z}^k$. Представлено также обсуждение методов борьбы с организацией скрытых каналов блужданий по плоскости.

В части 2 статьи структура частичного стирания будет рассматриваться абстрагировано от блужданий по плоскости. Выражаясь более определенно, речь пойдет о протоколе передачи информации в канале частичного стирания, называемом "равномерным кодом". Данный протокол надстраивает поверх указанного канала частичного стирания так называемый канал полного стирания. Поясним, что канал полного стирания задается множеством символов, для каждого из которых приписана вероятность того, что данный символ не будет распознан приемником, т. е. вероятность замены данного символа на некий зарезервированный символ стирания. Для указанного канала полного стирания построена математическая модель. Будут представлены и доказаны утверждения, относящиеся к оценке среднего числа циклов передачи. Также будет представлен конкретный пример, для которого проведены оценки пропускной способности.

Список литературы

1. **Lampson B. W.** A note on the confinement problem // Communications of ACM. — 1973. — Vol. 16, No. 10. — P. 613—615.
2. **McFarland J.** Covert Channels: An Overview // Preprint. 2017.
3. **Salwan N., Singh S., Arora S., Singh A.** An Insight to Covert Channels // arXiv:1306.2252. 2013.
4. **Грушо А. А.** Скрытые каналы и безопасность информации в компьютерных системах // Дискретная математика. — 1998. — Т. 10, № 1. — С. 3—9.
5. **Тимонина Е. Е.** Скрытые каналы (обзор) // Jet Info. — 2002. Т. 14, № 114. — С. 3—11.
6. **Zander S., Armitage G., Branch P.** Covert channels in multiplayer first person shooter online games // Proc. 33rd IEEE Conf. LCN. — 2008. — P. 215—222.
7. **Murdoch S., Zielinski P.** Covert Channels for Collusion in Online Computer Games // IH'04: Proceedings of the 6th International Conference on Information Hiding. 2004. — P. 355—369.
8. **Казаков И. Б.** Критерий надежности канала с запрещениями // Интеллектуальные системы. Теория и приложения. — 2019. — Т. 23, № 2. — С. 33—55.
9. **Казаков И. Б.** Кодирование в скрытом канале перестановки пакетов // Программная инженерия. — 2018. — Т. 9, № 4. — С. 163—173.
10. **Казаков И. Б.** Структура графа на множестве перестановок S_n, задаваемая моделью ошибки в скрытом канале перестановки пакетов // Интеллектуальные системы. Теория и приложения. — 2018. — Т. 22, № 2. — С. 53—79.
11. **Ван-дер-Варден Б. Л.** Алгебра. — М.: Мир, 1976. — 648 с.

Transmission of Information in Channels Specified by Structures of Partial Erasure. Part 1

I. B. Kazakov, i_b_kazakov@mail.ru, Lomonosov Moscow State University, Moscow, 119234, Russian Federation

Corresponding author:

Kazakov Ilia B., Postgraduate Student, Lomonosov Moscow State University, Moscow, 119234, Russian Federation
E-mail: i_b_kazakov@mail.ru

Received on May 06, 2020
Accepted on July 08, 2020

We investigate covert channels that use players' moves to transmit information. We assume that there exits noise generated by periods of invisiblility of the transmitting player (due to network failures, game mechanics, etc.); thus there arises the problem of reliable transmission of information over a noisy channel. The formal model that addresses this problem is referred to as a structure of partial erasure. In the framework of the model we prove the criterion of equivalence of a pair of trajectories on the receiving end of the channel. We also discuss a number of methods that prevent organization of covert channels of the type considered.

Keywords: *covert channels, walk on a plane, walk in $\mathbb{Z}^k$, structure of partial erasure*

For citation:

Kazakov I. B. Transmission of Information in Channels Specified by Structures of Partial Erasure. Part 1, *Programmnaya Ingeneria,* 2020, vol. 11, no. 5, pp. 277—284

DOI: 10.17587/prin.11.277-284

References

1. **Lampson B. W.** A note on the confinement problem, *Communications of ACM*, 1973, vol. 16, no. 10, pp. 613 — 615.
2. **McFarland J.** Covert Channels: An Overview, Preprint, 2017.
3. **Salwan N., Singh S., Arora S., Singh A.** An Insight to Covert Channels, arXiv:1306.2252. 2013.
4. **Grusho A. A.** Hidden channels and information security in computer systems, *Diskretnaya matematika,* 1998, vol. 10, no. 1, pp. 3—9 (in Russian).
5. **Timonina E. E.** Covert channels (survey)), *Jet Info,* 2002, vol. 14, no. 114, pp. 3—11 (in Russian).
6. **Zander S., Armitage G., Branch P.** Covert channels in multiplayer first person shooter online games, *Proc. 33rd IEEE Conf. LCN*, 2008, pp. 215—222.
7. **Murdoch S., Zielinski P.** Covert Channels for Collusion in Online Computer Games, *IH'04: Proceedings of the 6th International Conference on Information Hiding*, 2004, pp. 355—369.
8. **Kazakov I. B.** Reliabilty criterion for channels with prohibitions, *Intellektual'nye sistemy. Teoriya i prilozheniya*, 2019, vol. 23, no. 2, pp. 33—55 (in Russian).
9. **Kazakov I. B.** Coding in a Covert Channel of Data Packages' Permutations, *Programmnaya Ingeneria,* 2018, vol. 9, no. 4, pp. 163—173 (in Russian).
10. **Kazakov I. B.** The structure of the graph induced on the set of permutations S_n by an error model in a covert channel based on permutation of packets, *Intellektual'nye sistemy. Teoriya i prilozheniya,* 2018, vol. 22, no. 2, pp. 53—79 (in Russian).
11. **Van der Waerden B. L.** *Algebra,* Moscow, Mir, 1976, 648 p. (in Russian).

УДК 004.942 + 621.311.25:621.039.620.192 DOI: 10.17587/prin.11.285-295

О. М. Гулина[1], д-р техн. наук, проф., omgulina18@mail.ru,
Н. Л. Сальников[2], д-р техн. наук., проф., salnickovnickolay@yandex.ru,
В. П. Семишкин[1], д-р техн. наук, semishkin@grpress.podolsk.ru,
М. Н. Типикина [2], магистрант, tipikinamariya@mail.ru,
[1] АО ОКБ "ГИДРОПРЕСС", Подольск,
[2] ИАТЭ НИЯУ МИФИ, Обнинск

Разработка комплекса программ для управления ресурсом механических элементов АЭС в условиях эрозионно-коррозионного износа

Статья содержит результаты исследования, направленного на создание отечественной системы поддержки принятия решений по управлению ресурсом оборудования и трубопроводов атомных электростанций (АЭС), предназначенной для расчета интенсивности процесса эрозионно-коррозионного износа (ЭКИ) и оценки остаточного ресурса элементов АЭС. В качестве такого ресурса рассматривается продолжительность безопасной эксплуатации трубопровода на допустимых параметрах от данного момента времени до момента достижения прогнозируемого предельного состояния.

В РФ разработка программных средств по расчету скорости ЭКИ и утонения стенок трубопроводов АЭС проводится с начала 1990-х гг. Существующая версия программного комплекса расчета характеристик ЭКИ реализована в виде слабо связанных между собой программных модулей (отдельных программ). У специалистов, работающих в этой области, пока нет общего представления архитектуры подобных систем. Поэтому любая модернизация и дополнение функциональных возможностей системы сводится к переписыванию старых или написанию новых, слабо связанных со старыми, программ.

Статья содержит результаты исследования возможности реализации программного комплекса управления ресурсом в условиях ЭКИ в виде совокупности подключаемых (plug-in) модулей. Такой подход к построению системы позволит без больших ресурсозатрат расширять функциональные возможности системы без переписывания кода уже существующего ее ядра путем добавления или удаления модулей. На основе таких представлений авторами разработан ряд модулей, предназначенных для расчета характеристик ЭКИ и визуализации данных контроля.

Ключевые слова: *эрозионно-коррозионный износ, мониторинг, обработка результатов контроля, визуализация результатов контроля, остаточный ресурс*

Введение

Обеспечение безопасной и надежной работы пассивных тепломеханических элементов атомных электростанций (АЭС) включает методики принятия решений по управлению ресурсом в течение всего жизненного цикла. Работоспособность элементов трубопроводных систем энергоблоков первого и второго контуров, эксплуатирующихся в условиях эрозионно-коррозионного износа (ЭКИ), существенно влияет на стоимость и безопасность работы АЭС в целом. В связи с этим важным является наличие достоверной и долгосрочной практики наблюдений за состоянием критических элементов оборудования, а также применение математических методов прогнозирования, опирающихся на физические и/или эмпирические модели исследуемого процесса.

Актуальными с отмеченных выше позиций становятся вопросы прогнозирования состояния элементов оборудования и трубопроводов в условиях ЭКИ, а также вопросы обработки данных контроля толщин стенок и снижения консерватизма этих оценок.

Современный подход к управлению ресурсом включает: мониторинг толщин стенок элементов оборудования и трубопроводов в местах, наиболее подверженных ЭКИ; обработку результатов таких замеров; визуализацию поврежденного элемента и построение тренда. Эти места и элементы определяются на основании расчетов по прогнозным моделям, учитывающим как геометрию элемента,

состав металла, так и условия эксплуатации. Расчет остаточного ресурса, под которым понимается продолжительность безопасной эксплуатации трубопровода на допустимых параметрах от данного момента времени до момента достижения прогнозируемого предельного состояния, выполняется по полученным характеристикам интенсивности ЭКИ и на основании расчета допустимой толщины стенки. Таким образом определяются периодичность и объем контроля в рамках планово-профилактических работ.

Эрозионно-коррозионный износ, или *Flow Accelerated Corrosion* (*FAC*) является наиболее распространенным механизмом повреждения механических элементов оборудования и трубопроводов второго контура реактора. Износ стенки трубопровода происходит в результате воздействия потоков как однофазной[1], так и двухфазной[2] сред. Защитный слой, образующийся на внутренней поверхности стали в результате коррозионного процесса, смывается непрерывным потоком воды или пароводяной смеси.

Этот процесс происходит довольно медленно, однако при долгосрочном воздействии может привести к заметному утонению стенок труб и выходу из строя трубопровода. В этом аспекте ЭКИ существенно отличается от других процессов, так как даже существенные размывы внутренней поверхности приводят не к течи, а сразу к разрушению. Поэтому в работах, посвященных исследованию этого процесса, подчеркивается, что модель "течь перед разрушением" к описанию процесса не применима [1, 2]. Поэтому важным является отслеживание в процессе мониторинга приближения остаточной толщины стенки к минимально допустимому значению.

Установлено, что на интенсивность ЭКИ существенно влияют следующие факторы:

- температура и скорость среды;
- показатели водно-химического режима (pH, концентрация O_2, используемый амин);
- химический состав металла оборудования (весовое содержание в металле Cr, Cu и Mo);
- геометрические характеристики (типоразмер трубопровода, коэффициент Келлера);
- длительность эксплуатации оборудования (τ, лет);
- влажность пара (для паропроводов).

Для прогнозирования интенсивности ЭКИ в мировой практике наибольшее распространение получил программный продукт CHECWORKS на основе модели Чексала—Гурвица, учитывающей перечисленные факторы. Также широко используются программные продукты COMSY, DASY, WATHEC. Российские программные средства ЭКИ-02, ЭКИ-03 разработаны на той же основе [3]. Опыт эксплуатации программных систем, реализованных на основе этой модели, показывает, что интенсивность ЭКИ максимальна при следующих условиях:

- температуре 130...180 °C;
- низком pH;
- низкой концентрации кислорода;
- сложной геометрии элемента;
- низкой концентрации Cr и Cu;
- высокой скорости теплоносителя.

Одновременное сочетание неблагоприятных факторов может привести к ускоренному износу. Очевидно, что подход к прогнозированию интенсивности ЭКИ должен быть комплексным, включающим:

- обработку данных эксплуатационного контроля толщин стенок;
- хранение данных контроля для всех элементов;
- определение минимально допустимой толщины стенки и скорости ЭКИ в особых зонах;
- визуализацию данных контроля;
- прогнозирование скорости ЭКИ для однофазных и двухфазных потоков;
- прогнозирование остаточного эксплуатационного ресурса и даты следующего контроля.

На рис. 1 приведена схема методики обработки данных контроля толщины стенки. Она включает следующие процедуры:

- анализ протоколов замеров;
- первичную обработку полученных данных;
- определение мест локального повреждения;
- прогнозирование остаточного ресурса.

Для визуализации данных контроля и результатов расчетов строятся двух- и трехмерные развертки (карты толщин). На основании полученных результатов определяется дата следующего контроля для каждого рассмотренного элемента.

Для снижения неопределенности при оценке скорости ЭКИ требуется разработка специальных приемов обработки данных контроля. Например, более 70 методов существуют на АЭС США для оценки скорости ЭКИ по данным контроля, а наиболее широко используются 11 из них [4]. Основная цель — снизить излишний консерватизм оценок. В силу существующей неопределенности [5] такое снижение возможно за счет разработки новых методик оценки скорости ЭКИ, учитывающих как технологию изготовления, так и тип элемента: гиб, прямой участок, околошовная зона и т. п. Ряд работ авторов посвящен разработке таких методик [5, 6].

Обработка данных контроля включает расчет характеристик ЭКИ, определение размеров зон размыва, а также оценку скорости ЭКИ и скорости отложений продуктов коррозии (рис. 2).

Для решения перечисленных выше задач предлагается автоматизированная система в виде программного комплекса, состоящего из некоторого центрального хранилища данных, специально разработанного единого интерфейса пользователя (ввод и просмотр информации о замерах, расчет скорости ЭКИ, анализ ресурса оборудования) и ряда модулей, реализующих различные функции по учету, анализу и расчету остаточного ресурса оборудования АЭС, подверженного ЭКИ [7].

Следует отметить, что в настоящее время в России не существует единого программного комплекса оценки характеристик ЭКИ подобного зарубежным COMSY, CHECWORK, BRT-CICERO. Как правило, каждый исследователь рассчитывает ограниченное

[1] Однофазная среда — жидкая или газообразная среда без посторонних включений.

[2] Двухфазная среда — одно из веществ мелко распределено во втором.

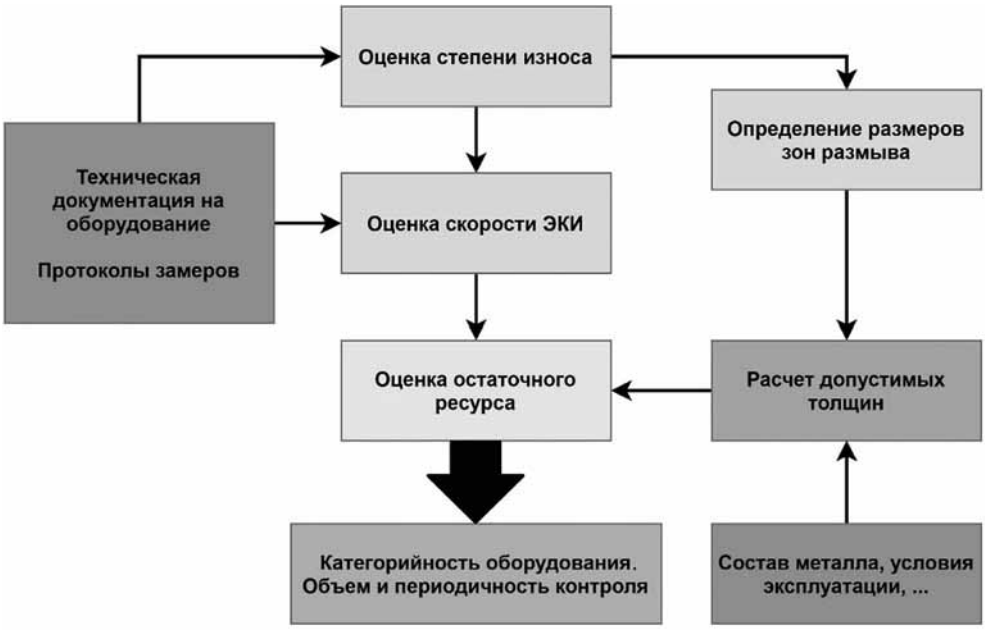


Рис. 1. Схема обработки данных контроля

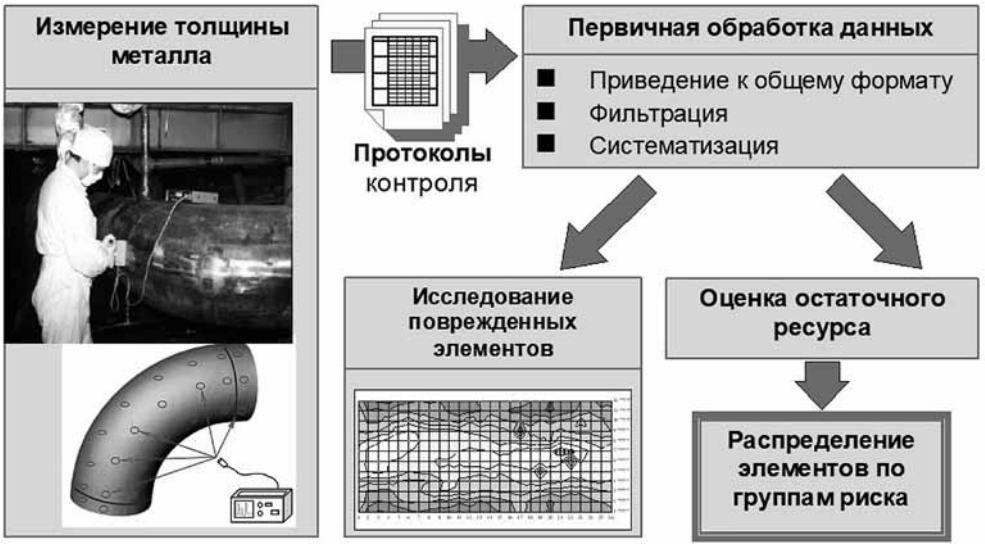


Рис. 2. Обработка данных эксплуатационного контроля

число характеристик ЭКИ, а существующие программные средства выполняют расчет лишь ограниченного набора этих характеристик. Упомянутые выше программные коды ЭКИ-02 и ЭКИ-03 выполняют расчет скорости ЭКИ в различные моменты времени, значения утонения стенки за расчетный интервал времени (для однофазной и двухфазной среды соответственно). Однако они не решают, например вопросы хранения и визуализации данных контроля, а также расчета остаточного ресурса.

Следует подчеркнуть, что существует большое число методик обработки данных контроля (методик расчета характеристик), при этом активно разрабатываются новые. Поэтому система должна быть *расширяемой*, позволяющей без дополнительных ресурсозатрат добавлять новые или модернизировать уже существующие методики расчета и визуализации. Представленный далее подход позволит создавать системы управления ресурсом, независящие от уже используемых на практике моделей или методик оценки повреждений.

Архитектура системы в виде подключаемых модулей

Реализация программного комплекса в виде совокупности подключаемых (*plug-in*) модулей позволяет без больших ресурсозатрат (включая квалификацию разработчика) менять состав системы без переписывания уже существующего ядра системы путем добавления или удаления модулей.

Система (программный комплекс) в виде подключаемых (*plug-in*) модулей состоит из следующих элементов:

- *MainProgram* — основное приложение или ядро системы, к которому будут подключаться плагины;

- *Interface* — определяет интерфейс для плагинов, необходимый для единообразной работы с ними;
- *Plugin*1, *Plugin*2, ... — подключаемые модули, обеспечивающие дополнительные функциональные возможности.

Плагины реализуются в виде библиотек общего пользования и все указанные элементы помещаются в отдельные *сборки* (файлы с расширением .exe и .dll). Для разработки плагинов не нужно иметь доступа к исходному коду приложения, поэтому они могут быть созданы сторонними разработчиками. Применение сборок позволяет многократно использовать типы, определенные внутри сборки, не зависящим от языка образом. Можно, например, создать библиотеку кода на C# и повторно использовать ее при написании кода на любом языке программирования .NET.

На рис. 3 представлена диаграмма классов расширяемой системы с отмеченными на ней сборками.

Основной модуль (MainProgram.exe) ведет диалог с пользователем (форма MainForm) и выполняет загрузку подключаемых модулей (класс PluginLoader).

Интерфейс (IPlugin), который должны реализовывать все подключаемые модули, помещен в сборку DataContarct.dll. Каждый плагин должен иметь имя Name и метод, возвращающий форму с результатами работы (GetForm). Использование интерфейса также позволяет ввести абстракцию модулей, скрыть внутренние детали. Главная программа оперирует с абстрактными конструкциями, все детали выносятся за пределы основной программы, что позволяет легко модернизировать и развивать систему.

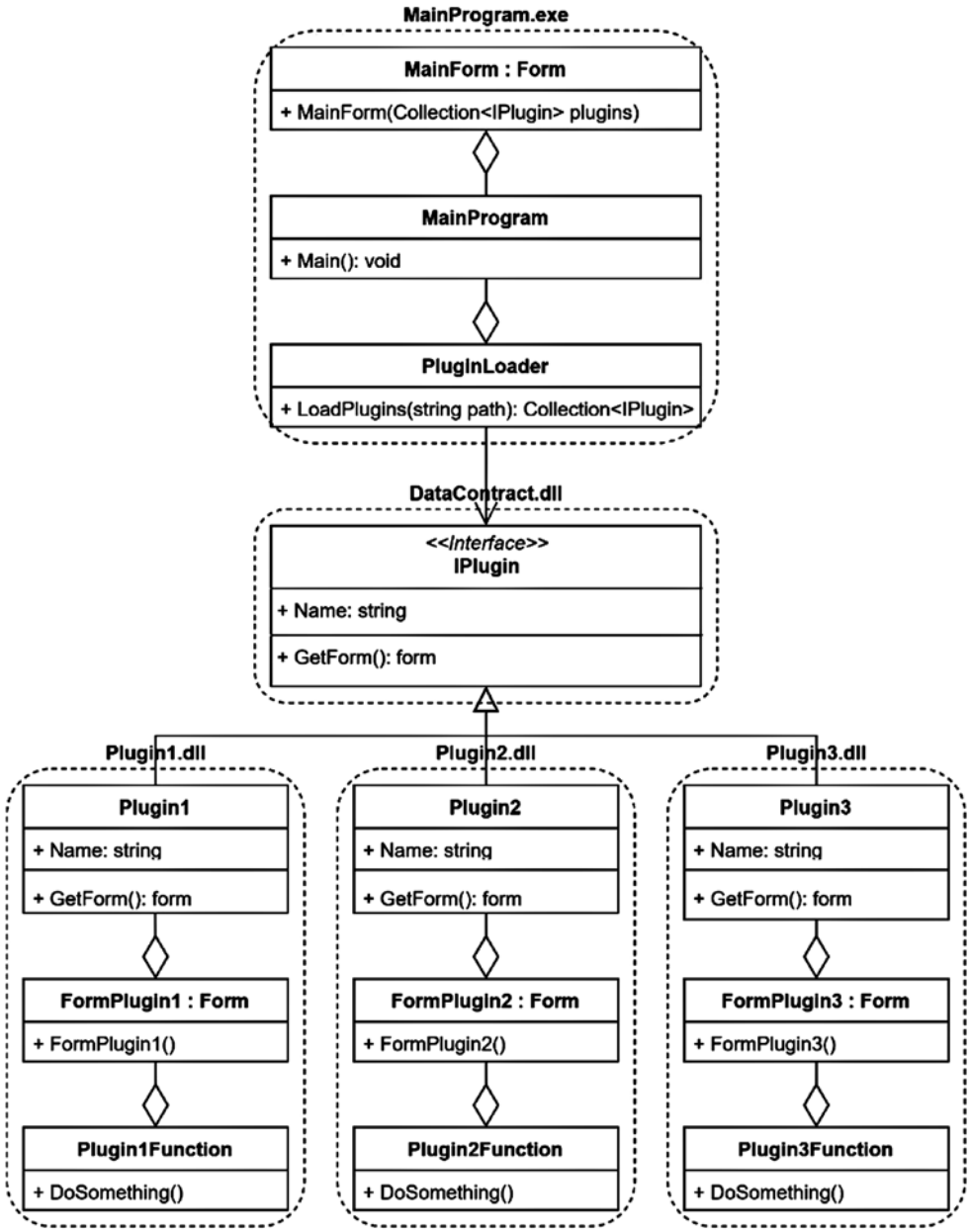


Рис. 3. Диаграмма классов расширяемой системы с отмеченными на ней сборками

Подключаемые модули (Plugin1.dll, Plugin2.dll, Plugin3.dll) выполняют некоторые индивидуальные задачи — построение модели визуализации или расчет характеристик ЭКИ в соответствии с некоторой выбранной методикой. Сборки модулей содержат:

- класс Plugin, наследуемый от описанного выше интерфейса IPlugin;
- форму FormPlugin для вывода на экран результатов работы;
- класс PluginFunction, реализующий функциональность модуля (построение моделей, расчеты).

Реализация системы в виде подключаемых модулей

Схема программного комплекса расчета характеристик ЭКИ, выполненная в соответствии с описанным выше архитектурным решением, показана на рис. 4.

Слой хранения данных включает в себя некоторое центральное хранилище (например, базу данных) и считыватель данных — отдельный программный модуль, предоставляющий данные по запросу главной программы (ядра системы).

Слой отображения данных содержит ядро системы (главную программу или управляющий модуль), который ведет диалог с пользователем и выполняет загрузку подключаемых модулей.

Интерфейс — описание методов и свойств, которые должны реализовывать все подключаемые модули.

Подключаемые модули — отдельные программные модули, выполняющие различные задачи по обработке и отображению данных контроля состояния трубопроводов.

Вспомогательные модули — программные модули, предназначенные для реализации некоторых общих, используемых другими модулями, функций.

Следует расширить диаграмму классов, представленную на рис. 3, путем добавления *считывателя данных*.

Данные контроля (таблицы протоколов замеров) можно хранить различными способами: в письменном виде для дальнейшего ручного ввода, в виде текстовых файлов, электронных таблиц, в базе данных и т. д. В настоящее время широкое распространение имеет практика хранения данных контроля в виде файлов Excel.

Таким образом, считыватель должен быть реализован так, чтобы его легко можно было заменить или добавить еще один. Для этого необходимо исключить зависимость от него подключаемых модулей и главной программы. Этого можно добиться за счет создания *абстрактной фабрики данных* — модуля DataFactory.dll, который по заданному типу источника создает соответствующий экземпляр (рис. 5).

На рис. 5 представлено два считывателя данных:

- ExcelReader.dll для чтения данных из файлов формата Excel, содержащих таблицу протокола замеров с их координатами;
- TextReader.dll для чтения данных из текстовых файлов (.txt), содержащих полную информацию об элементе (результаты замеров, номинальную толщину трубопровода, диаметр, ...).

Рис. 4. Архитектурное решение

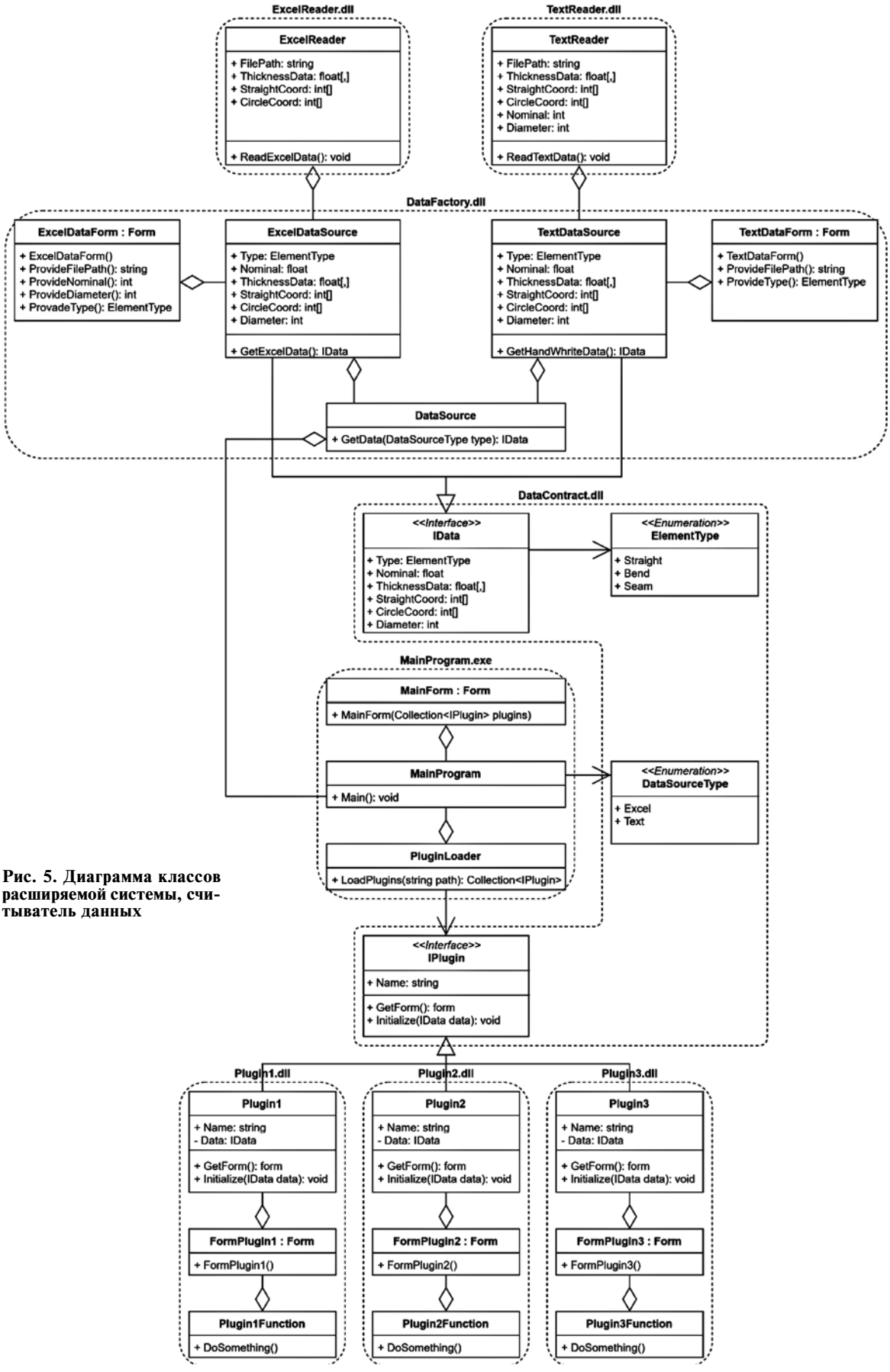


Рис. 5. Диаграмма классов расширяемой системы, считыватель данных

Классы модуля DataFactory.dll реализуют все необходимое для работы с конкретным типом источника. Для ExcelData.dll нужна форма (ExcelDataForm) для ввода пути к файлу Excel, номинального значения, диаметра элемента и его типа. Для TextData.dll нужна форма (TextDataForm) для ввода пути к файлу .txt, содержащему все необходимые данные.

В главной программе выбирается желаемый тип источника данных (DataSourceType). На основании выбранного типа модуль DataFactory.dll открывает необходимые источники, формирует и подготавливает данные. В результате на выходе модуля имеется ссылка на интерфейс IData, предоставляющий доступ к данным. Далее происходит инициализация загружаемых модулей (инициализация поля Data с помощью метода Initialize, реализуемого всеми модулями). Каждому модулю предоставляется ссылка на загруженные данные. Модуль обрабатывает полученные по ссылке данные и становится полностью независимым от того, откуда и как эти данные были получены. На основании моделирования и расчетов (методы Plugin1Function, Plugin2Function, Plugin3Function) каждый модуль предоставляет экранную форму (FormPlugin1, FormPlugin2, FormPlugin3) с результатами обработки данных.

Для того чтобы добавить новый считыватель необходимо лишь расширить модуль DataFactory.dll (добавить элементы, предоставляющие необходимые считывателю данные), при этом внесение изменений в главную программу и подключаемые модули не требуется.

Результат программной реализации

Модульное приложение, предназначенное для обработки данных контроля (см. рис. 1), реализовано с использованием описанного выше архитектурного решения на языке C# платформы .NET. На рис. 6 (см. вторую сторону обложки) представлено окно главной программы, содержащее дочернее (вложенное) окно вызываемого модуля.

Ядро приложения выполняет загрузку модулей и динамически формирует пользовательский интерфейс (набор кнопок на панели слева) в зависимости от набора подключаемых модулей. В частности, на рис. 6 (см. вторую сторону обложки), выполнена загрузка шести модулей (Табличное представление, Графики толщин, Интерполированные графики толщин, 2D-модель, 3D-модель, Характеристики ЭКИ).

Модуль "Табличное представление" (рис. 6, *а* — см. вторую сторону обложки). Результаты замеров толщин стенок элементов трубопроводов в протоколах замеров имеют вид матрицы с разным числом строк и столбцов в зависимости от типа элемента трубопровода (прямой участок, гиб, околошовная зона и т. д.). Для оценки остаточного ресурса элемента необходимы определенные преобразования и расчеты, однако общее представление о характере износа можно получить с помощью так называемых картограмм. На рис. 6, *а* приведен результат работы модуля "Табличное представление", выполняющего отображение таблицы протокола замеров, где строки — это замеры в осевом направлении, а столбцы — в окружном направлении, с выделением цветом измеренных толщин в зависимости от отклонения от номинального значения (ячейка, содержащая значение, меньшее номинального (утонение), окрашивается в оттенки красного цвета; ячейка, содержащая значение, большее номинального (отложение), — в оттенки зеленого; ячейка, содержащая значение, равное номинальному, окрашена в белый цвет).

Модули "2D-модель" и "3D-модель" (рис. 6, *б* и *в*, см. вторую сторону обложки). Модули предназначены для построения моделей внутренней поверхности трубопровода, с выделением дефектов с помощью цвета. Для этих модулей разработан и реализован алгоритм двумерной интерполяции цвета по заданному списку узлов как базы для построения трехмерных моделей [8].

Разработаны алгоритмы построения трехмерных моделей типовых элементов: прямых участков, гибов[1] и околошовных зон (рис. 6, *в*, см. вторую сторону обложки), окраска которых выполняется на основе созданных интерполяционных алгоритмов, которые позволяют получить наглядные трехмерные изображения внутренней поверхности и передать особенности *пространственного расположения* ЭКИ-повреждений с помощью цвета. Окраска осуществляется по принципу, который используется при окраске таблицы в модуле "Табличное представление".

Указанные модели не имеют аналогов среди существующих способов визуализации.

Модуль "Графики толщин" (рис. 6, *г*, см. вторую сторону обложки) предназначен для построения графиков распределения толщин (минимальных, средних, максимальных) в осевом и окружном направлениях, для выявления участков трубопровода, наиболее подверженных ЭКИ.

Модуль "Интерполированные графики толщин" (рис. 6, *д*, см. вторую сторону обложки). Для построения двумерных и трехмерных моделей выполнена реализация алгоритма двумерной интерполяции цвета по заданному списку узлов [8]. Механизм интерполяции может быть применен и при построении графиков. Концептуальная схема результата работы модуля "Интерполированные графики толщин" изображена на рис. 7, см. третью сторону обложки.

Интерполированные толщины представляют собой двумерную матрицу, столбцы которой соответствуют сечениям, перпендикулярным осевому направлению (рис. 7, слева), строки соответствуют осевым сечениям (рис. 7, справа).

На рис. 6, *г*, см. вторую сторону обложки, показана реализация данного модуля. С помощью подвижного маркера пользователь указывает сечение, в котором необходимо построить график. Управление маркером осуществляется с помощью элементов управления, расположенных над графиками.

Так как для гибов замеры в окружном направлении проводятся только на растянутой части (рис. 8), то и построение графиков осуществляется только для растянутой части (рис. 9, см. третью сторону обложки).

Модуль "Характеристики ЭКИ" (рис. 6, *е*, см. вторую сторону обложки) предназначен для расчета

[1] Гиб — колено, изготовленное в трубогибочном станке.

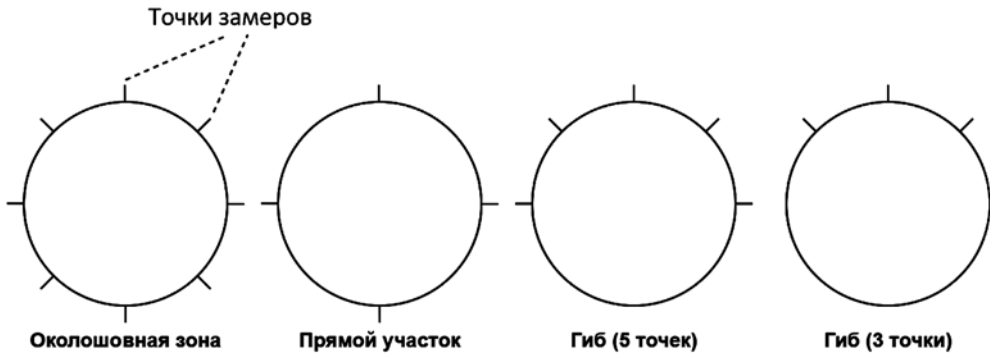


Рис. 8. Точки замеров в окружном направлении

следующих характеристик ЭКИ: размер зон размыва, характер зон размыва (локальный, равномерный), допустимая толщина стенки, скорость ЭКИ, остаточный ресурс.

Для определения зон размыва также используется ранее реализованный алгоритм двумерной интерполяции [8]. Локализация зон размыва осуществляется путем построчного сканирования матрицы интерполированных толщин. Для каждого дефекта вычисляется размер в осевом и окружном направлениях. Далее в соответствии с РД ЭО 1.1.2.11.0571—2010 "Нормы допускаемых толщин стенок элементов трубопроводов из углеродистых сталей при эрозионно-коррозионном износе" [9] выполняется расчет перечисленных выше характеристик ЭКИ.

Интерфейс модуля "Характеристики ЭКИ" включает две вкладки. Вкладка "Обнаруженные дефекты" содержит таблицу с данными об обнаруженных дефектах и картограмму с отмеченными на ней дефектами (рис. 6, *е*, слева). При нажатии на строку, соответствующую некоторому дефекту, границы дефекта выделяются полужирной линией.

Вкладка "Характеристики ЭКИ" содержит формы для ввода дополнительных данных об элементе и результаты расчета (рис. 6, *е*, справа).

Модуль прогнозирования

На данном этапе разработки данный модуль реализован как отдельное программное средство, которое в дальнейшем будет включено в ранее описанный программный комплекс. Данный модуль аналогичен зарубежным компьютерным кодам семейства CHEC и российским программным средствам ЭКИ-02 и ЭКИ-03. Для реализации ПС ПРЭК (программное средство прогнозирования эрозии-коррозии) выбран язык JAVA, позволяющий реализовать все требуемые функции, он является объектно-ориентированным и кроссплатформенным. Программа включает в себя две автономных составляющих — для однофазной и двухфазной сред. Разработан интерфейс, позволяющий удобно использовать функции программы. Программа реализует расчеты скорости ЭКИ и утонения стенки элементов трубопроводов, позволяет строить, визуализировать и сохранять расчетные таблицы (рис. 10, 11).

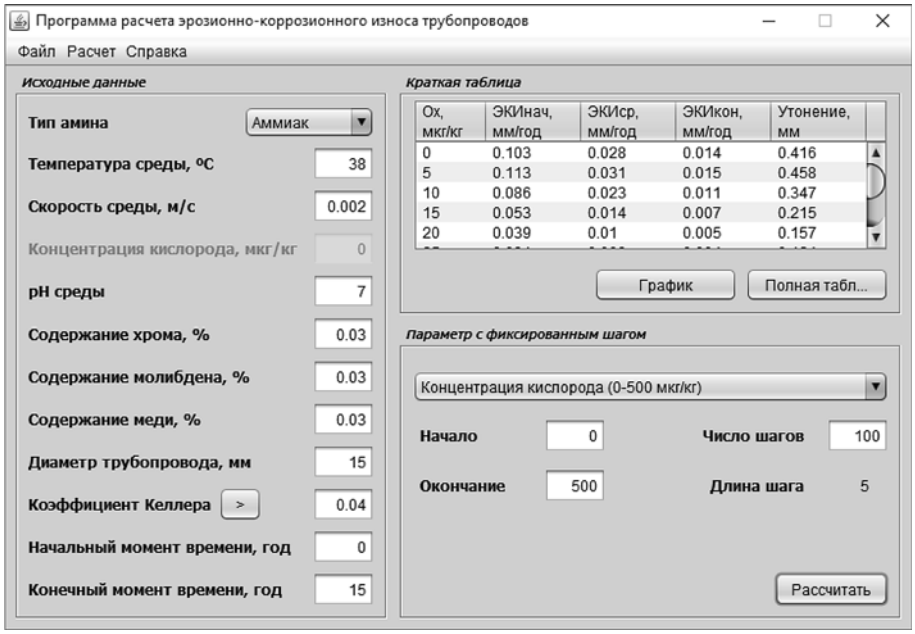


Рис. 10. Пользовательский интерфейс модуля прогнозирования (для однофазной среды)

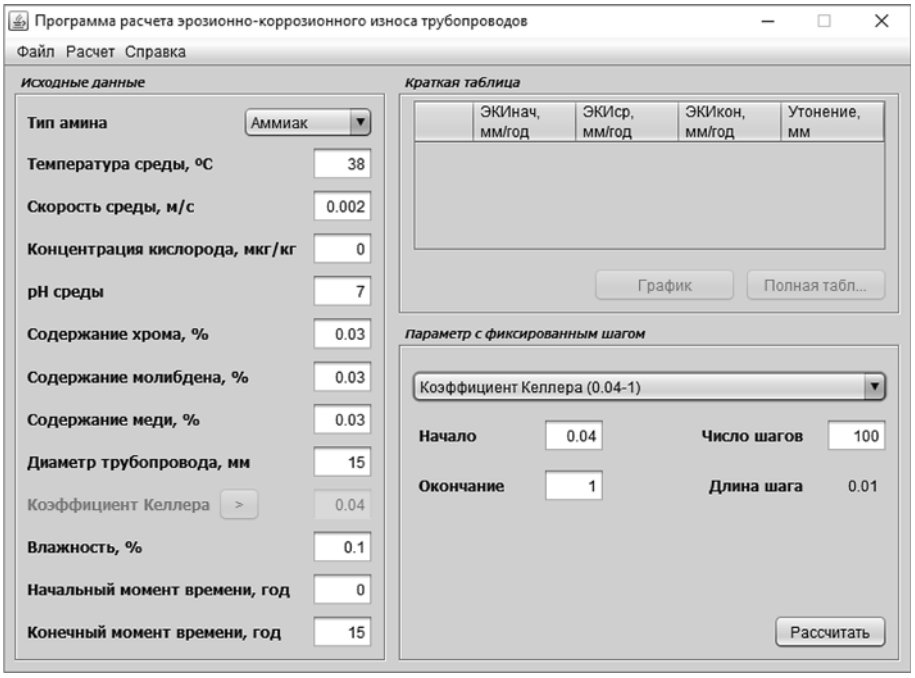


Рис. 11. Пользовательский интерфейс модуля прогнозирования (для двухфазной среды)

По сравнению с прототипами расширены границы применения модели Чексала—Гурвица, лежащей в основе прогнозной модели, для реакторных установок РБМК-1000 (ректор большой мощности канальный), увеличен временной интервал прогнозирования до 100 лет.

С помощью разработанного программного средства ПРЭК выполнены верификационные расчеты для РБМК-1000 (ректор большой мощности канальный), ВВЭР-1000 (водо-водяной энергетический реактор), ВВЭР-440 и BWR (*Boiling Water Reactor*).

На рис. 12 приведены графики верификации для двух АЭС с РБМК-1000 для трубопроводов питательной воды (ТПВ) и паропроводов (ПП). Максимальная разница между оценкой по данным контроля и данными расчета — 25 %.

На основе разработанного программного комплекса и кода ПРЭК выполнено исследование показателей эрозионно-коррозионного износа для гиба трубопровода питательной воды типоразмера 273×16 мм АЭС с ВВЭР-440 и гиба трубопровода питательной воды типоразмера 457×25 мм шведской АЭС Форсмарк-1 [10]. Сравнительный анализ показал удовлетворительное согласие расчета и данных контроля (см. таблицу).

Различие результатов расчета и измерения для АЭС с BWR составляет 6 %, для АЭС с ВВЭР-440 — 9 %.

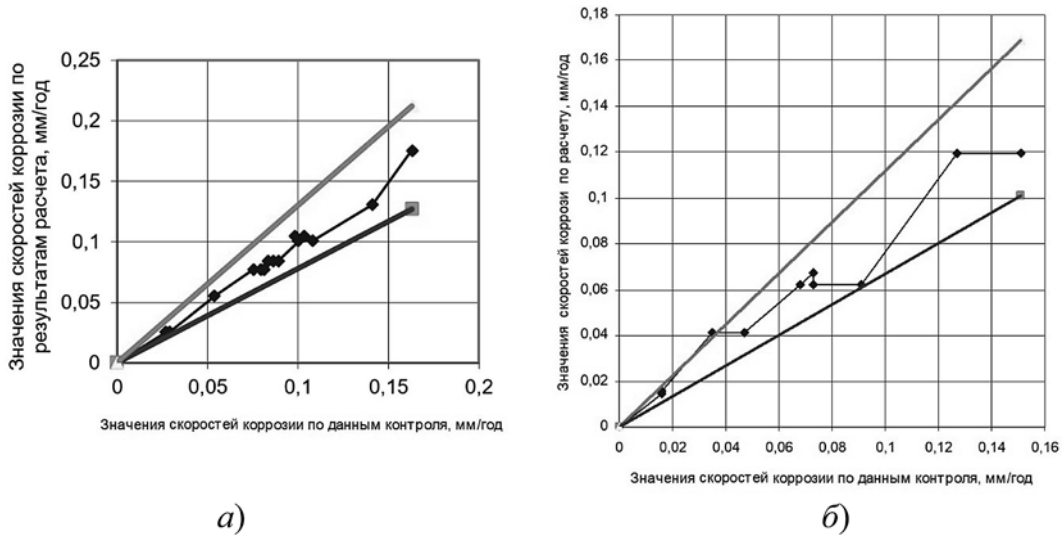


Рис. 12. Графики верификации:
а — ТПВ АЭС № 1; *б* — ПП АЭС № 2 (прямые линии — верхняя и нижняя граница соответствия)

Исходные и расчетные значения толщин и утонений, мм

Показатели (исходные и расчетные)	АЭС с BWR	АЭС с ВВЭР-440
Исходная толщина	23,4	15,7
Толщина (средняя)	21,6	13,8
Утонение (среднее)	1,8	1,9
Расчетное утонение	1,684	2,081

Разработка программного комплекса и прогнозной модели типа ПРЭК позволяет оперативно обрабатывать данные замеров, оценивать интенсивность ЭКИ и остаточный ресурс, а прогноз и адаптация кода к каждому блоку позволяют существенно уменьшить объемы контроля и, соответственно, снизить расходы — временные и финансовые.

Заключение

В статье представлены результаты исследования по созданию отечественной системы поддержки принятия решений по управлению ресурсом оборудования и трубопроводов АЭС, осуществляющей расчет интенсивности процесса ЭКИ, а также оценивающей остаточный ресурс элементов.

Реализован комплексный подход к прогнозированию интенсивности ЭКИ, включающий: обработку данных контроля; определение минимально допустимой толщины стенки и скорости ЭКИ в особых зонах; визуализацию данных контроля; прогнозирование скорости ЭКИ для однофазных и двухфазных потоков; прогнозирование остаточного эксплуатационного ресурса и даты следующего контроля.

В качестве архитектурного решения выбрана компоновка системы как единого программного комплекса. Такое решение позволяет расширять функциональные возможности программного комплекса за счет добавления новых подключаемых модулей без необходимости внесения изменений в другие модули и ядро системы.

Разработаны:

— модули визуализации данных замеров с восстановлением недостающих данных,

— модули расчета скорости ЭКИ и остаточного эксплуатационного ресурса.

Предлагаемое решение создает основу для построения объединенного интерактивного комплекса "**Система поддержки принятия решения в условиях ЭКИ**", позволяющего оптимизировать объем и периодичность контроля элементов оборудования АЭС. Описанная в статье идея демонстрирует общие подходы в проектировании систем управления ресурсом независимо от рассматриваемых механизмов старения или методик оценки повреждений.

Список литературы

1. **Horowitz J., Smith D.** Recommendation for an Effective Flow-Accelerated Corrosion Program (NSAC-202L-R4). EPRI / 3002000563. Technical Report, November 2013. 94 p.
2. **Muhammadu M. M., Sheriff J., Hamzah E.** A Review of Literature for the Flow Accelerated Corrosion of Mitred Bends // International Journal of Emerging Technology and Advanced Engineering. — 2013. — Vol. 3, Issue 8. — P. 663—677.
3. **Нафталь М. М., Бараненко В. И., Гулина О. М.** Использование программных средств для расчета эрозионно-коррозионного износа оборудования и трубопроводов АЭС // Теплоэнергетика. — 2014. — № 6. — С. 73—80.
4. **Бараненко В. И., Гулина О. М., Сальников Н. Л., Мурзина О. Э.** Обоснование расчетов скорости эрозионно-коррозионного износа и остаточного ресурса трубопроводов АЭС по данным эксплуатационного контроля // Ядерная энергетика — 2016. — № 2. — С. 55—65.
5. **Бараненко В. И., Гулина О. М., Сальников Н. Л.** Расчет скорости коррозии и остаточного ресурса элементов трубопроводов АЭС по данным контроля // Ядерная энергетика. — 2017. — № 4. — С. 83—94.
6. **Щербаков А. В., Гулина О. М., Сальников Н. Л.** Программный комплекс расчета допустимых толщин стенок элементов оборудования АЭС в условиях эрозионно-коррозионного износа // Ядерная энергетика. — 2014. — № 2. — С. 62—69.
7. **Гулина О. М., Федотов В. А.** Разработка системы поддержки принятия решений по прогнозированию ресурса оборудования АЭС в условиях эрозионно-коррозионного износа // Программная инженерия. — 2014. — № 8. — С. 9—16.
8. **Гулина О. М., Типикина М. Н., Типикин Н. Г.** Программно-математическая модель визуализации данных толщинометрии трубопроводов АЭС // Программная инженерия. — 2019. — Т. 10, № 5. — С. 226—233.
9. **РД ЭО** 1.1.2.11.0571—2010. Нормы допускаемых толщин стенок элементов трубопроводов из углеродистых сталей при эрозионно-коррозионном износе. — М.: ВНИИАЭС. — 2012. — 104 с.
10. **Sweden** country presentation IAEA FAC RCM 2. On the risk of using grid measurements when close to allowable minimum wall thickness. Erlangen, 2014.06.05.

Development of Software Package for Managing the Lifetime of Mechanical Elements of Nuclear Power Plants under Flow-Accelerated Corrosion (FAC)

O. M. Gulina[1], omgulina18@mail.ru, **N. L. Salnikov**[2], salnickovnickolay@yandex.ru,
V. P. Semishkin[1], semishkin@grpress.podolsk.ru,
M. N. Tipikina[2], tipikinamariya@mail.ru,
[1]OKB "GIDROPRESS", Podolsk, 142103, Russian Federation
[2]Obninsk Institute for Nuclear Power Engineering (IATE MEPHI), Obninsk, 249040, Russian Federation

Corresponding author:

Tipikina Maria N., Graduate Student, Obninsk Institute for Nuclear Power Engineering (IATE MEPHI), Obninsk, 249040, Russian Federation
E-mail: tipikinamariya@mail.ru

Received on June 24, 2020
Accepted on August 10, 2020

This article focuses on results of developing the system for managing the life time of equipment and pipelines of nuclear power plants (NPP). The system is designed to calculate the intensity of Flow-Accelerated Corrosion (FAC) and a residual lifetime of elements.

In Russia, the development of software for calculating the speed of FAC and value of thinning of the walls of pipelines of NPP has been carried out since the early 90s of the last century. The existing version of the software package for calculating the characteristics of FAC is implemented of loosely coupled software modules (standalone programs). There is no single concept of architecture, therefore, any modernization and addition of functionality comes down to rewriting old programs or writing new programs, loosely connected with the old ones.

This article contains results of research of the feasibility of implementing a software package for monitoring FAC in the form of a set of plug-in modules. This approach allows one to easily add new functionality without overwriting the existing kernel by adding or removing modules. Based on the general concept, several modules for calculating the characters of FAC and visualizing the control data have been developed.

Keywords: *erosion-corrosion wear, monitoring, processing of control results, visualization of control results, residual lifetime*

For citation:

Gulina O. M., Salnikov N. L., Semishkin V. P., Tipikina M. N. Development of Software Package for Managing the Lifetime of Mechanical Elements of Nuclear Power Plants under Flow-Accelerated Corrosion (FAC), *Programmnaya Ingeneria,* 2020, vol. 11, no. 5, pp. 285—295

DOI: 10.17587/prin.11.285-295

References

1. **Horowitz J., Smith D.** Recommendation for an Effective Flow-Accelerated Corrosion Program (NSAC-202L-R4). EPRI / 3002000563. Technical Report, November 2013. 94 p.

2. **Muhammadu M. M., Sheriff J., Hamzah E.** A Review of Literature for the Flow Accelerated Corrosion of Mitred Bends, *International Journal of Emerging Technology and Advanced Engineering*, 2013, vol. 3, issue 8, pp. 663—677.

3. **Naftal' M. M., Baranenko V. I., Gulina O. M.** Use of Software Tools for Calculating Flow Accelerated Corrosion of Nuclear Power Plant Equipment and Pipelines, Thermal Engineering, 2014, vol. 61, no. 6, pp. 456—463.

4. **Baranenko V. I., Gulina O. M., Sal'nikov N. L., Murzina O. Je.** Substantiation of FAC Rate and Service Life Estimation UNDER Operation Control Data, *Jadernaja jenergetika*, 2016, no. 2, pp. 55—65 (in Russian).

5. **Baranenko V. I., Gulina O. M., Sal'nikov N. L.** Flow-Accelerated Corrosion Rate and Residual Life Time Estimation for the Components of Pipeline Systems at NPPS Based on Control Data, *Jadernaja jenergetika*, 2017, no. 4, pp. 83—94 (in Russian).

6. **Shherbakov A. V., Gulina O. M., Sal'nikov N. L.** Calculation Code to Estimate Admissible Thickness of NPP Equipment Components under Flow-Accelerated Corrosion, *Jadernaja jenergetika,* 2014, no. 2, pp. 62—94 (in Russian).

7. **Gulina O. M., Fedotov V. A.** Development of Decision Support System on the Prediction of the Life of NPP Equipment under Erosion-Corrosion Conditions, *Programmnaya Ingeneria,* 2014, no. 8, pp. 9—16 (in Russian).

8. **Gulina O. M., Tipikina M. N., Tipikin N. G.** Mathematical Model of Data Visualization of the NPP Pipelines Thickness and Software Implement, *Programmnaya Ingeneria*, 2019, vol. 10, no. 5, pp. 226—233 (in Russian).

9. **RD EO 1.1.2.11.0571—2010.** Normy dopuskaemyh tolshhin stenok jelementov truboprovodov iz uglerodistyh stalej pri jerozionno-korrozionnom iznose. — M.: VNIIAJeS. — 2012. — 104 p.

10. **Sweden country presentation IAEA FAC RCM 2.** On the risk of using grid measurements when close to allowable minimum wall thickness. Erlangen, 2014.06.05.

УДК 004.4, 004.7 DOI: 10.17587/prin.11.296-308

О. Б. Петрина, науч. сотр., petrina@cs.petrsu.ru, Петрозаводский государственный университет

Ранжирование информации на основе семантической сети при построении цифровых сервисов персонализированного сопровождения деятельности музея

Исследованы задачи персонализированного сопровождения деятельности музея, для решения которых предложено использовать цифровые сервисы. Сервисы реализуют отбор источников информации из музейного хранилища с использованием дополнительного уровня — семантического. Семантический уровень создается над каталого-ориентированным информационным хранилищем музея. Разработана онтологическая модель, в соответствии с которой представляется информация в виде семантической сети. Предложена модель локального ранжирования объектов семантической сети, на основе которой разработан алгоритм ранжирования. Представлена программная реализация алгоритма локального ранжирования в цифровых сервисах персонализированного сопровождения деятельности музея на примере Музея истории Петрозаводского государственного университета. Сервисы предоставляют персонализированные рекомендации посетителям музея при изучении экспозиции, что требует различных вариантов ранжирования экспонатов и отбора экспонатов с наибольшим рангом.

Ключевые слова: *ранжирование, семантическая сеть, цифровые сервисы, персонализация, сопровождение деятельности музея*

Введение

В области разработки цифровых сервисов информационного обеспечения музея в мире уже накоплен значительный опыт [1]. Первые музейные информационные системы использовались для совершенствования организации учета музейных предметов. Виртуальные музейные коллекции включают как метаданные и описание источников, так и цифровые копии музейных предметов [2]. В настоящее время развиты возможности персонализированного доступа к музейным коллекциям [3]. Например, цифровые сервисы в области электронного туризма позволяют оповещать заинтересованных пользователей о проводимых выставках, помогать планировать индивидуальный маршрут посещения музея и предлагать участие в интерактивных мероприятиях [4].

Персонализация сервиса достигается в том числе с использованием смартфонов — посетитель музея через смартфон может изучить текущую ситуацию в выставочных залах для планирования своего маршрута и навигации по экспозициям [5]. Дополнительные возможности персонализации при изучении музейной коллекции достигаются за счет создания индивидуального профиля пользователя, когда посетитель музея регистрирует свои предпочтения и эти данные используются для разработки персонализированных виртуальных и музейных туров [6]. Такого рода персонализация информационного сопровождения музея востребована и историками-исследователями, стремящимися к получению комплекса информации об источнике [7]. Исследователю может потребоваться обращение не только к непосредственному содержанию источника, но и к контексту его создания и бытования, что представлено в других, связанных с ним источниках.

Возможности традиционного способа автоматизации поиска — в музейной информационной системе по ключевым словам — весьма ограничены, так как из поля зрения исследователя выпадают те источники, которые не имеют формально полного совпадения по искомым признакам. Такой способ не дает удобного решения по сортировке выявленных источников в зависимости от степени их релевантности заданной задаче исследования, а также количественных оценок степени релевантности для выполнения отбора нужных источников. В статье исследуется усовершенствование персонализированного выявления и отбора нужных источников информации за счет сведения к математической задаче ранжирования [8] исходных прикладных задач, которые могут возникать при информационном сопровождении деятельности музея: задача выявления экспонатов для последующего изучения, задача навигации, задача вывода новых знаний.

Решение задач сопровождения деятельности музея рассматривается на примере Музея Истории Петрозаводского государственного университета (ПетрГУ) [7]. В качестве источников информации выступают экспонаты Музея Истории ПетрГУ. Разработана онтологическая модель для семантического представления информации музейных коллекций [9]. Онтологическая модель предназначена для связывания историко-культурной информации об экспонатах и связанных сущностях в музее в семантическую сеть. Семантическую сеть можно анализировать с помощью алгоритмов, выявляя релевантную информацию для заданных параметров.

Для вычисления оценок релевантности экспонатов разработана модель локального ранжирования [10], реализованная в виде алгоритма в цифровых сервисах музея. Алгоритм ранжирования вычисляет оценки релевантности (в виде числового значения — ранга) экспонатов на основе заданных критериев, представленных в профиле пользователя. Указанные критерии сопоставляют с имеющимся описанием экспоната — чем больше соответствий критериям, тем выше ранг. Вычисленные ранги позволяют отсортировать экспонаты и предоставить исследователю список наиболее релевантных из имеющегося множества. За счет использования значений рангов также возможно управлять длиной результирующего списка, сужая или расширяя диапазон допустимой значимости отбираемых источников.

В статье сформулированы задачи, возникающие при информационном сопровождении деятельности музея. Для решения обозначенных задач предложено выполнить семантическое связывание информации на основе методов онтологического моделирования. Далее представлена математическая модель локального ранжирования экспонатов в семантической сети. Описаны цифровые сервисы персонализированного сопровождения деятельности музея. Для построения каждого сервиса предложен конкретный вариант выполнения ранжирования экспонатов в семантической сети. Представлен алгоритм локального ранжирования, реализованный в программной многоагентной системе сервисов персонализированного сопровождения деятельности музея. Предложены формальные показатели качества выполнения алгоритма и выполнены начальные эксперименты.

Задачи сопровождения деятельности музея

Рассмотрим следующие три задачи, возникающие при информационном сопровождении деятельности музея [7]: задача выявления экспонатов (информационного сопровождения), задача навигации (построение траектории изучения источников), задача вывода новых знаний (обогащение источников новыми информационными описаниями). Решению этих задач может способствовать построение списка источников, рекомендуемых к изучению в рамках исследуемой темы на основе параметров, заданных исследователем и меняющихся в процессе работы при уточнении цели исследования или критериев формирования его источниковой базы. Традиционно отбор источников ведет сам посетитель музея, но это требует объемной ручной работы. Изменение выборки источников в случае, если в ходе изучения отдельных документов исследователь понимает, что он не учел важные критерии отбора, предполагает дополнительные значительные трудозатраты.

Задача выявления экспонатов. При изучении посетителем экспонатов ему рекомендуется набор релевантных источников из музейной коллекции. В музее набор определенных экспонатов и связей между ними дает возможность раскрыть характерные черты эпохи. Необходимо автоматизировать процесс отбора экспонатов, требуемых для изучения поставленного вопроса. Не требуется ручной и трудоемкий для человека перебор большого числа визуальных документов в музейной коллекции и можно сконцентрироваться на анализе выявленного комплекса как совокупности.

Задача навигации. При изучении выявленного комплекса экспонатов посетителю рекомендуется план изучения (маршрут, траектория и т. п.). Так, в музее посетитель должен определять, к какому следующему экспонату обратиться для дальнейшего углубления исследования. Необходимо автоматизировать процесс поддержки принятия решений по отбору экспонатов, чтобы была возможность охватить всю важную для посетителя информацию из музейной коллекции.

Задача вывода новых знаний. При изучении экспонатов посетитель музея может обратить внимание на имеющиеся информационные лакуны. Так, в музее при изучении фотографий исследователь может дополнять коллекцию экспертной информацией: а) какие персонажи и б) в каком контексте представлены на фотографии. Необходимо автоматизировать выявление отсутствующих фактов (информационного описания) и скрытых неочевидных фактов и закономерностей. В первом случае пользователю рекомендуется выполнить заполнение обнаруженных информационных пробелов в описании источника (например, отсутствует информация, кто изображен на фотографии). Во втором случае исследователю рекомендуется выполнить дополнительный анализ имеющейся информации для экспертной интерпретации (например, несколько фотографий соответствуют одному и тому же событию).

Решение описанных задач сопровождения деятельности музея рассматривается на примере музея Истории Петрозаводского государственного университета [6]. Музей Истории ПетрГУ представляет собой типичный небольшой музей. Помимо традиционных витрин в музее представлены 12 цифровых дисплеев различных размеров с изменяющимися изображениями экспонатов за более чем 75-летнюю историю ПетрГУ (с 1940 г.). Значительную часть музейного фонда составляют фотографии. Их содержание разнообразно и затрагивает самые разные темы: персональные биографии, историю университета и его отдельных структурных подразделений (факультетов и кафедр), историю науки и техники, культуру и быт и т. д., что можно охарактеризовать

общим термином "история повседневности университета". В музейной информационной системе (МИС) представлены цифровые образы значительной части фотоколлекции, а цифровые сервисы позволяют работать с этим массивом.

Решение описанных задач сопровождения деятельности предлагается выполнить с использованием ранжирования экспонатов в цифровых сервисах информационного сопровождения деятельности музея. Ранжирование позволит присвоить экспонатам ранги, что автоматизирует процесс отбора релевантных экспонатов для каждой задачи.

Семантическое представление информации музейных коллекций

Выполнение ранжирования требует семантического представления информации в пригодном для автоматической обработки виде. Предлагается использовать семантическую сеть для структурирования информации музейных коллекций. Семантическую сеть можно анализировать с помощью алгоритмов, выявляя релевантную информацию по заданным параметрам и тем самым предоставляя пользователю сервисов персонализированный доступ к музейной коллекции.

Для решения задач сопровождения деятельности музея ранее было предложено выполнить семантическое связывание информации [11] на основе методов онтологического моделирования. Онтологическая модель дает описание предметной области в доступной для автоматической обработки форме [7]. В случае музейной коллекции описание строится вокруг музейных артефактов в виде понятий и правил, утверждений об этих понятиях [12]. Онтологическими классами являются как музейные экспонаты, так и другие значимые сущности: персоналии, географические объекты, исторические события и др. Онтология позволяет идентифицировать каждый отдельный предмет и описать его связь с другими. Рассмотрим онтологическую модель [9] для представления информации музейных коллекций, которая является основой для построения семантической сети экспонатов и связанных сущностей. Анализ семантической сети математическими методами позволит автоматизировать поиск релевантных экспонатов, результаты которого будут использованы для составления персонализированных рекомендаций в музее.

Основная предметная область, экспонаты которой представлены в музее, относится к истории повседневности университета. Музейные экспонаты представлены как в физически осязаемой форме (например, мебель, одежда), так и в электронной форме (оцифрованная фотография, запись интервью). Разработанная онтологическая модель сосредоточена на существующих экспонатах в МИС Музея Истории ПетрГУ [9]. Для представления исторической информации об экспонатах была использована известная онтология верхнего уровня CIDOC CRM и дополнена уникальной частью.

Все классы, имеющие в своем названии букву "*E*", заимствованы из онтологии CIDOC CRM. На представленном фрагменте онтологической модели (рис. 1) экземпляры класса *E71 Man-Made Thing*

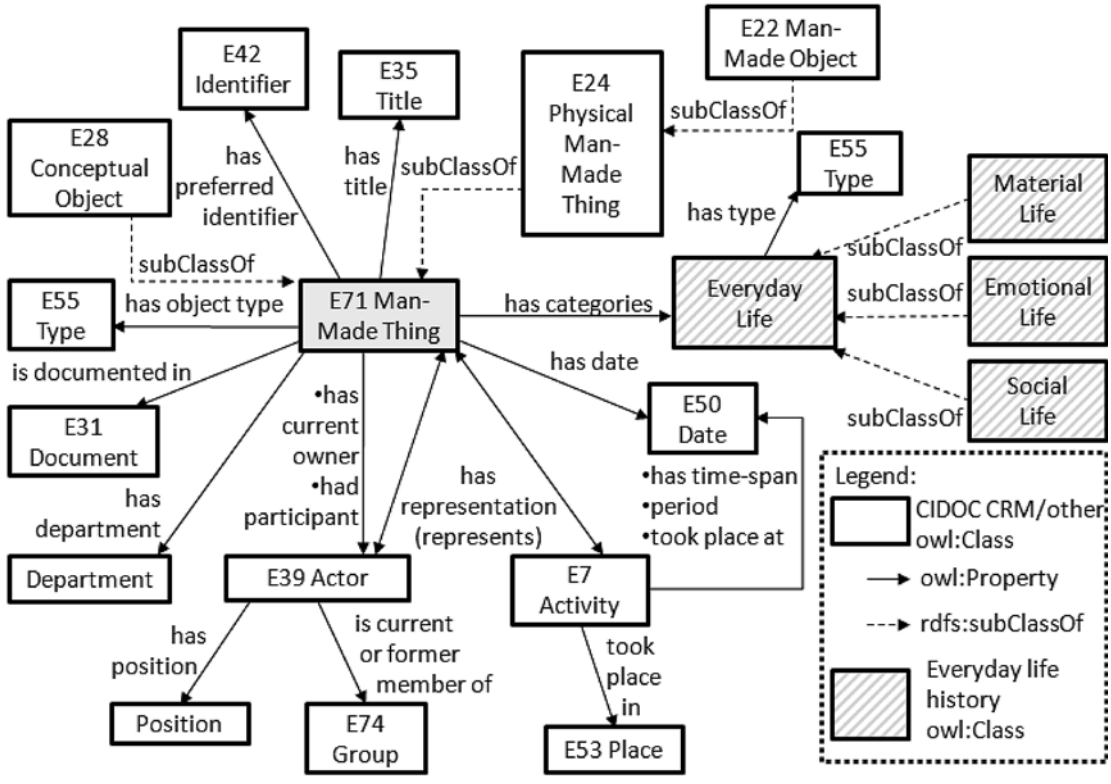


Рис. 1. Фрагмент онтологической модели

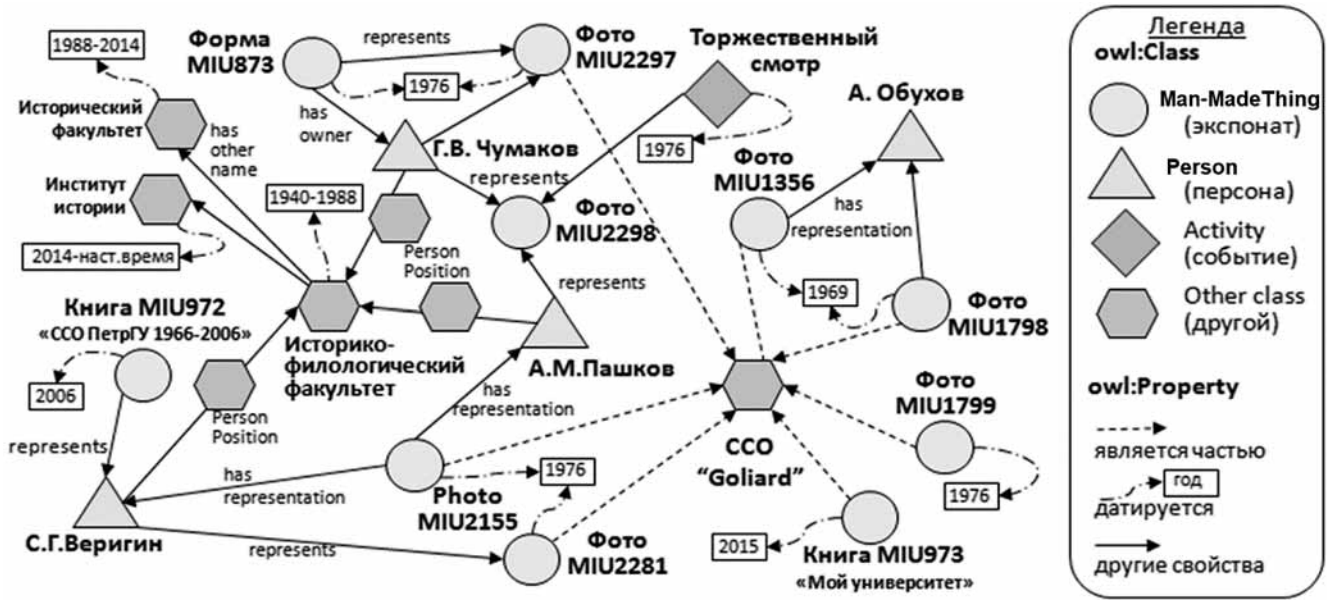


Рис. 2. Пример семантической сети

(далее *Man-Made Thing*) являются идентифицируемыми экспонатами, описание которых содержится в МИС. Уникальная часть включает в себя классы и их свойства для описания следующих сущностей: специфика университетского музея (класс *Department* для описания факультетов, класс *Group* для обозначения социальной группы, например, стройотряды, класс *Everyday Life* для обозначения выделенных структур повседневности университетской жизни), пользовательские профили (класс *Profile*) и смежная информация о персоналиях (класс *PersonPosition* для описания должности, класс *PersonEducation* для указания сведений об образовании), ранги в соответствии с требуемыми сервисами (класс *Rank*).

Разработанная онтологическая модель определяет структуру хранимой информации и используется для построения семантической сети. Семантическая сеть описывает объекты, явления и понятия предметной области с помощью сетевых структур, основанных на теории графов. Семантическая сеть представляет орграф, узлы соответствуют физическим и цифровым экспонатам, связанным с ними событиям и людям, а связи отражают взаимосвязь объектов (рис. 2).

Такое структурированное и связанное представление информации позволяет применить алгоритмы ранжирования для выявления и отбора нужной информации [13]. Содержание информационного хранилища зарегистрировано как онтологическая база знаний истории повседневности ПетрГУ [14] и используется при построении цифровых сервисов персонализированного сопровождения деятельности музея. Метрики семантической сети музея истории ПетрГУ представлены в табл. 1.

Из онтологической модели можно выделить фрагмент, описывающий пользовательский профиль. Пользовательский профиль определяет условия персонализации для выполнения алгоритма ранжирования с учетом интересов данного пользователя. На рис. 3 представлена схема связей онтологического класса *Profile* с классом *Man-Made Thing* через другие классы. В онтологической модели пользовательский профиль представлен следующими свойствами класса Profile.

1. Свойство *has person location* — "памятные места пользователя" связано с классом *Place*, который в свою очередь, связан свойством *has place* с классом *Man-Made Thing.*

Таблица 1

Показатели семантической сети истории повседневности ПетрГУ

Показатель	Метрика	Значение
Объекты семантической сети	Общее число объектов, из которых	729
	— экспонаты (экземпляры класса *Man-Made Thing*)	297
	— персоны (экземпляры класса *Person*)	60
Размер онтологической модели	Число классов	33
	Число свойств	112
Размер используемого RDF-хранилища	Число RDF-троек	70 226

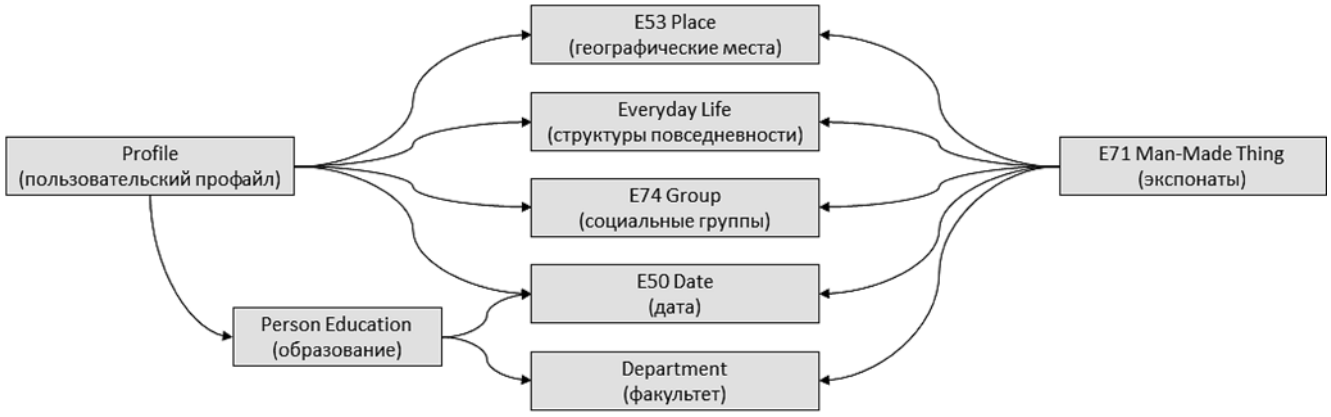


Рис. 3. Связи онтологических классов *Profile* и *Man-Made Thing*

2. Свойство *has Department* — "факультет, на котором обучался пользователь" связан с классом *Department*, который, в свою очередь, связан свойством *has department* — "упоминаемый объект или экспонат относится к факультету" с классом *Man-Made Thing.*

3. Свойство *Joined* — дата поступления в университет и *Graduated* — дата окончания университета связаны с классом *Date*, который, в свою очередь, связан свойством *has date* — дата создания экспонатов класса *Man-Made Thing.*

4. Свойство *has Hobby* — увлечения пользователя в студенческие годы связано с классом *Everyday Life.* Экземпляры класса *Everyday Life* представляют собой наименования разработанных структур повседневности университетской жизни. Каждый экземпляр класса *Man-Made Thing* связан с классом *Everyday Life* многозначным свойством *has categories.*

5. Свойство *has Group* — отношение пользователя к социальной группе, представленной онтологическим классом *Group.* В частности, в Музее Истории ПетрГУ представлено множество экспонатов, относящихся к стройотрядовскому движению. Экземпляры класса *Man-Made Thing* связаны с классом *Group* свойством *has SB* (от англ. "*Student construction brigades*" — студенческий строительный отряд). Экспонат проходит однозначную классификацию по отношению к стройотряду, поэтому потребовалось явное выделение свойства *has SB,* иначе принадлежность к стройотряду можно было бы рассматривать как одну из тем экспоната в свойстве *has Topic.*

Таким образом, в текущем разделе была представлена разработанная онтологическая модель, основанная на онтологии верхнего уровня CIDOC CRM и дополненная уникальной частью с учетом специфики Музея Истории ПетрГУ и разрабатываемых сервисов. На основе разработанной модели выполняется структурирование информации в виде семантической сети. В дальнейшем будет показано, каким образом можно применить алгоритм ранжирования в семантической сети для выявления и отбора нужной информации.

Локальное ранжирование экспонатов в семантической сети

Отбор экспонатов необходим при построении экспозиции сотрудником музея с учетом тематики и предполагаемых интересов посетителей [7]. При посещении экспозиции посетителю может понадобиться тематический отбор экспонатов с учетом индивидуальных интересов для определения последовательности обхода в экспозиционном помещении музея. При изучении отдельного экспоната заинтересованным посетителям также может требоваться тематический отбор близких экспонатов к заданному экспонату для определения наиболее значимых свойств. Для выполнения тематического отбора необходимо выполнить ранжирование экспонатов по степени значимости к интересам изучающего. Рассмотрим модель локального ранжирования экспонатов.

Математическая задача ранжирования предполагает упорядочивание набора объектов по степени их релевантности заданному параметру [15]. Значение ранга объекта количественно характеризует степень релевантности. Такой отбор может быть основан на локальном ранжировании, когда ранг объекта вычисляется по отношению к выделенному объекту (попарное сравнение), без учета семантических взаимосвязей с другими объектами [10]. Локальное ранжирование проявляется в соответствии с задачами сопровождения деятельности музея следующим образом: а) отбор наиболее близких по тематике экспонатов для центрального экспоната экспозиции; б) отбор наиболее интересных для заданного посетителя экспонатов из представленных в текущей экспозиции; в) отбор наиболее интересных для заданного посетителя информационных фактов (свойств) при изучении конкретного экспоната.

Для выполнения ранжирования экспонатов музея предлагается модель, которая описывает семантическое сопоставление сущностей для вычисления оценок релевантности. Например, схожесть двух экспонатов в музее или степень близости профиля посетителя музея с конкретным экспонатом. Данный

ранг также необходим для составления персональных рекомендаций.

Модель локального ранжирования определяется следующими формальными параметрами. В основе модели локального ранжирования лежит разработанная концептуальная схема представления данных, обозначим O — онтология. Пусть U — это множество профилей пользователей. Каждый профиль $u \in U$ представлен экземпляром онтологического класса *Profile*. Обозначим за M множество экспонатов. Каждый экспонат $m \in M$ представлен экземпляром онтологического класса *Man-Made Thing*. Обозначим c — профиль пользователя $c \in U$ или заданный экспонат $c \in M$, относительно которого совершается ранжирование. Важно отделить объект (экспонат или профиль), относительно которого вычисляются ранги, от остальных экспонатов. Для обозначения множества значений всех свойств онтологических классов возьмем P. Тогда p_k^c — значение свойства k для объекта (экспонат или профиль). Значением свойства p_k^c является конечный набор $n \geqslant 1$ значений, т. е. $p_k^c = \{p_{k1}^c,\ p_{k2}^c,\ ...,\ p_{kn}^c\}$. Аналогично, свойства экземпляра класса *Man-Made Thing* обозначим множеством $p_k^m = \{p_{k1}^m,\ p_{k2}^m,\ ...,\ p_{kl}^m\}$, где $l \geqslant 1$ — число значений свойства p_k^c. Однозначным свойством будем называть свойство, мощность множества значений которого равна единице, т. е. $n = |p_k^c| = 1,\ l = |p_k^m| = 1$. Многозначные свойства — это свойства, мощность множества значений которых больше единицы: $n = |p_k^c| > 1,\ l = |p_k^m| > 1$.

В основе предлагаемой модели локального ранжирования лежит семантическое сопоставление информации. Чем сильнее семантически связаны фрагменты информации, тем выше значение ранга. Метод семантического сопоставления информации идентифицирует данные, которые являются семантически связанными. Проводится попарное семантическое сопоставление значений свойств среди экземпляра выбранного объекта (профиля пользователя, представленного классом *Profile* или заданного экспоната — *Man-Made Thing*) и экземпляров онтологического класса *Man-Made Thing*, описывающего музейный экспонат. Таким образом, $f(p_k^c,\ p_k^m)$ — функция семантического сопоставления свойств попарно сравниваемых объектов.

Семантическое сопоставление происходит среди значений свойств. Выделяется несколько типов свойств: однозначные, многозначные, обратные и датирующие. Для разных типов свойств функция может принимать разный вид.

1. Посимвольное сравнение.

Для большинства значений свойств выполняется посимвольное сравнение. Для однозначных свойств значение функции сопоставления будет равно 1. Для многозначных свойств рассчитывается мощность множества пересечения двух множеств значений свойств:

$$f_{sym}(p_k^c,\ p_k^m) = |p_k^c \cap p_k^m|. \quad (1)$$

2. Сравнение обратных свойств.

Обратное свойство предполагает, что существует свойство, обратным к которому является данное свойство [16]. Значения обратных свойств сравниваются в соответствии с бинарной функцией:

$$f_{rev}(p_k^c,\ p_k^m) = \begin{cases} 1, & \text{если свойства взаимообратны,} \\ 0, & \text{иначе.} \end{cases} \quad (2)$$

3. Проверка совпадения временного периода.

Для датирования экспоната — экземпляра класса *Man-Made Thing* используется одно свойство, а для обозначения периода обучения в университете посетителя экземпляра класса *Profile* используются два свойства для начала периода и для окончания. Все три свойства однозначные и имеют числовое значение. Проверка вхождения даты создания экспоната в период обучения в университете пользователя происходит следующим образом:

$$f_{dat}(p_k^c,\ p_g^c,\ p_d^{m_i}) = \begin{cases} 1, & \text{если } p_k^c \leqslant p_d^{m_i} \text{ и } p_g^c \geqslant p_d^{m_i}, \\ 0, & \text{иначе.} \end{cases} \quad (3)$$

Ранг между объектом c и экспонатом m_i равен сумме значений всех функций семантического сопоставления свойств и вычисляется по формуле

$$r_{cm_i} = f_{sym} + f_{rev} + f_{dat}. \quad (4)$$

Представленная модель ранжирования позволяет вычислить оценки релевантности для объектов семантической сети для заданного объекта. Результаты ранжирования могут быть использованы для решения выделенных задач информационного сопровождения деятельности музея.

Применение ранжирования для решения задач информационного сопровождения деятельности музея

Локальное ранжирование объектов семантической сети применяется для решения задач сопровождения деятельности музея. В основе построения цифровых сервисов [7] используется семантическая сеть музея. Предлагается использовать модель локального ранжирования экспонатов на основе семантической сети для решения конкретной задачи сопровождения деятельности музея в следующих трех сервисах: изучения музейного фонда, изучения экспоната и пополнения. Для каждого сервиса определен вариант ранжирования экспонатов на основе предложенной выше модели, которые будут составлять алгоритм локального ранжирования экспонатов.

Сервис изучения музейного фонда предназначен для отбора экспонатов, представляющих наибольший интерес для исследователя. Результат предоставляется исследователю на его смартфоне или на экране в помещении музея. Соответствующие критерии ранжирования задаются в профиле пользователя сервиса, например, в виде ключевых слов (интересующие тематические рубрики), периодов времени, регионов. Эти задаваемые исследователем интересы определяют критерии для последующего ранжирования источников. После выполнения ран-

жирования набор рекомендуемых источников составляется из числа источников наибольшего ранга. Число источников в наборе определяется заданными ограничениями (например, планируемым временем для изучения).

Сервис изучения музейного фонда допускает адаптацию предложенного набора рекомендованных источников в зависимости от текущего состояния исследования. При изучении некоторого источника выявляемые исследователем факты и связи могут вноситься в профиль пользователя. Выполняется повторное ранжирование с обновлением набора рекомендуемых источников для изучения исследователем.

В сервисе изучения музейного фонда реализовано локальное ранжирование для определения семантической связности экспонатов с пользовательским профилем. Ранжирование экспонатов происходит на основе информации в профиле пользователя, если необходимо построить персональную программу посещения для заданного пользователя. Вычисляется ранг для каждого экспоната на основе профиля пользователя. Ранг экспоната зависит от профиля пользователя — чем больше сходство описания экспоната с профилем пользователя, тем выше ранг экспоната. Семантическое сопоставление значений свойств выполняется среди экземпляров класса *Man-Made Thing* и класса *Profile*, представленных в табл. 2.

В результате для конкретного пользователя формируется множество рангов экспонатов. Отсортировав значения рангов по убыванию, можно получить наиболее значимые для пользователя экспонаты. Такое ранжирование используется сервисом изучения музейного фонда для создания персонализированной программы посещения. Сервис формирует конкретную тематическую экспозицию из группы тесно связанных экспонатов. Экспонаты в этой тематической экспозиции оцениваются в соответствии с профилем пользователя. Сервис посещения предоставляет возможность представить персональную программу, где экспонаты семантически согласованы с профилем пользователя.

Сервис изучения экспоната предназначен для отбора экспонатов и оперативного отображения их цифрового изображения на экранах рядом с вещественным экспонатом. В том числе могут использоваться смартфон или планшет, на которых отображаются связанные с данным экспонатом другие источники. При отображении возможно дополнение описательной информацией. В результате сервис создает своего рода виртуализацию, когда выставка с вещественными экспонатами в помещении дополняется цифровым представлением на основе связанных с экспонатом визуальных источников и информацией о них.

Для построения сервиса применяется ранжирование всех тех источников, которые связаны с заданным в семантической сети. Критериями ранжирования выступают описательные информационные поля экспоната (например, тематическая рубрика, дата создания, автор). Числовое значение ранга определяет тематическую близость источника к экспонату, т. е. источники наибольшего ранга определяют набор, который необходим для изучения экспоната в совокупности со смежной историко-культурной информацией. Пользователю предоставляется отсортированный список рекомендуемых экспонатов. При выборе некоторого экспоната из списка будет выведено на экран его изображение и дополнительная информация, поясняющая связь с экспонатом.

В сервисе изучения экспоната реализовано локальное ранжирование для определения семантической связности экспонатов. Семантическое сопоставление применяется для отражения семантической связности музейных экспонатов друг с другом. Ранг вычисляется для каждого экспоната относительно других экспонатов. Рассмотрим подробно семантическое сопоставление значений свойств для определения семантической связности музейных экспонатов. В качестве исходного параметра может выступать заданный экспонат, для которого необходимо определить похожие, семантически связанные экспонаты. Выполняется сопоставление среди экземпляров класса *Man-Made Thing*, представленных в табл. 3.

Таблица 2

Сопоставимые свойства классов *Profile* и *Man-Made Thing*

Тип свойства	Свойство класса *Profile*	Свойство класса *Man-Made Thing*	Функция семантического сопоставления
Однозначное	*has Department*	*has Department*	f_{sym} по выражению (1)
Многозначное	*has Department*	*other Department Name*	f_{sym} по выражению (1)
	has Group	*has SB*	
	has Hobby	*has Category*	
	has person location	*has Place*	
Временной	*Joined*	*has Date*	f_{dat} по выражению (3)
	Graduated	*has Date*	

Таблица 3

Свойства класса *Man-Made Thing*

Тип свойства	Свойство	Значение свойства	Функция семантического сопоставления
Однозначные	*has Author*	Создатель экспоната	f_{sym} по выражению (1)
	has Date	Дата создания экспоната	
	refers to Department	Факультет, к которому относится экспонат	
	has Owner	Собственник, бывший владелец экспоната	
	has SB	Принадлежность экспоната к стройотрядовскому движению	
Многозначное	*represents*	Существуют экспонат или персона, которые имеют отношение к текущему экспонату ("изображен на фотографии", "упомянут в интервью")	f_{sym} по выражению (1)
	has Topic	Тема, к которой может быть отнесен экспонат	
	has Place	Место съемки, если экспонат является фотографией	
	has Category	Структуры повседневности, к которым может быть отнесен экспонат	
Обратное	*is Part of*	Экспонат, составной частью которого является текущий экспонат	f_{rev} по выражению (2)
Обратное многозначное	*has Part*	Экспонат, который является частью текущего экспоната	f_{rev} по выражению (2)

Ранг экспоната складывается из суммы всех семантических сопоставлений. Чем больше совпадений свойств у двух экспонатов, тем выше ранг между ними. Такое ранжирование необходимо, когда исследователь просматривает текущие экспонаты. Сервис изучения экспоната может визуализировать описания рекомендуемых экспонатов на окружающих экранах или на мобильном устройстве, когда посетитель просматривает конкретный экспонат и изучает связанную с ним информацию.

Сервис пополнения предназначен для внесения новой информации в музейную коллекцию. Поддерживается возможность модификации семантической сети музея на основе результатов, получаемых исследователем при изучении экспозиции и представленных экспонатов. Таким образом, сервис предоставляет исследователю возможность быть не пассивным потребителем информации при изучении источников, но и обогащать имеющееся информационное описание. В частности, сервис пополнения важен для музейного персонала, поскольку позволяет дополнять описание хранимых предметов информацией от исследователей и делать их представление более содержательным.

Цифровые образы музейных предметов, хранящихся в музейном фонде, дублируются в мобильном приложении. После изучения экспоната у исследователя есть возможность уточнить информацию своими знаниями. Для этого в мобильном приложении предусмотрена возможность добавления комментариев (в виде текста, аудио- или видеоинформации) для внесения данных по конкретному экспонату. Сотрудник музея видит добавленную информацию и оценивает ее с точки зрения качества и ценности.

Ранжирование экспонатов относительно друг друга происходит после внесения новой информации. Добавление данных в информационную систему происходит сотрудником музея с формированием новых связей в семантической сети. Критериями ранжирования выступают новые семантические связи. Существенное изменение ранга между двумя источниками после добавления новой информации позволит исследователю выявить скрытые связи между изучаемыми источниками.

Сервис пополнения выполняет локальное ранжирование для определения субъективной связности экспонатов. Выполняется итеративное ранжирование экспонатов относительно конкретного экспоната и пользователя в ситуации, когда пользователь просматривает конкретный экспонат. Сначала происходит поиск экспонатов, похожих на конкретный экспонат, согласно ранжированию для определения семантической связности экспонатов. Затем эти экспонаты оцениваются в соответствии с профилем пользователя согласно ранжированию для определения семантической связности экспонатов с пользовательским профилем. В результате пользователь видит список экспонатов, похожих на текущий и упорядоченных относительно его профиля.

Таким образом, выделены три цифровых сервиса персонализированного сопровождения деятельности музея. Предложенное использование модели позволяет выделить конкретные варианты выполнения локального ранжирования для каждого сервиса.

Выделенные варианты формируют общий алгоритм локального ранжирования экспонатов семантической сети.

Алгоритм локального ранжирования

Алгоритм локального ранжирования используется в цифровых сервисах персонализированного сопровождения деятельности музея для вычисления оценок релевантности экспонатов в зависимости от решаемой задачи. Был проведен эксперимент для сервиса изучения музейного фонда Музея Истории ПетрГУ. Получены результаты ранжирования для выбранных пользователей и подсчитаны значения метрик оценки алгоритма.

Реализован алгоритм локального ранжирования вычисления оценок релевантности экспонатов. Входными данными алгоритма является семантическая сеть всех экспонатов и их свойств, заданный объект c — экспонат или профиль пользователя. В качестве выходных данных представлен сортированный список рекомендованных экспонатов, извлеченных из базы знаний. Программный агент реализует алгоритм локального ранжирования в соответствии со следующими шагами.

Шаг 1. Извлечь из семантической сети все свойства p_k^c заданного объекта c. Создать объект онтологического класса *Man-Made Thing* или *Profile* с его свойствами $\{p_{k1}^c, p_{k2}^c, ..., p_{kn}^c\}$.

Шаг 2. Получить список всех экспонатов $m \in M$ *manMadeThings_list* из семантической сети в качестве объектов онтологического класса *Man-Made Thing*.

Шаг 3. Получить информацию об экспонатах. Пока список экспонатов не пуст, выполняется функция, которая извлекает из семантической сети все свойства экспоната и их значения и записывает их в объект класса *Man-Made Thing* $p_k^m = \{p_{k1}^m, p_{k2}^m, ..., p_{kl}^m\}$.

Шаг 4. Для каждого m_i из списка экспонатов рассчитать ранг относительно заданного объекта c.

Шаг 4.1. $r_{cm} = 0$.

Шаг 4.2. Вычислить функцию семантического сопоставления $f(p_k^c, p_k^m)$ значений свойств согласно типу свойства.

Шаг 4.3. $r_{cm_i} += f(p_k^c, p_k^m)$.

Шаг 4.4. Если есть еще не сопоставленные свойства p_k, то вернуться к шагу 4.2.

Шаг 4.5. Подсчитанные значения рангов записать в матрицу.

Шаг 5. Преобразовать матрицу рангов в утверждения вида (s, p, o), чтобы согласно онтологической модели представить их в семантической сети. Создать утверждение вида (s, p, o), где s — заданный объект c; o — текущий экспонат m; p — онтологическое свойство *rank_value*. Выполнить проверку, если *rank_value* не существует, то создать новое свойство со значением r_{cm}, иначе значение свойства *rank_value* обнулить и присвоить новое значение ранга r_{cm}.

Шаг 6. Выполнить сортировку рангов. Наиболее релевантные экспонаты отобразить пользователю как рекомендации.

На рис. 4 представлена архитектурная схема и пронумерованы шаги алгоритма локального ранжирования. Программная инфраструктура системы локального ранжирования представляет собой многоагентную сервис-ориентированную информационную систему. Все сервисы персонализированного сопровождения деятельности музея реализуются как семантические сервисы интеллектуального пространства [17] и доставляются пользователям через специализированные клиенты. Сервисы реализуются в результате косвенного взаимодействия программных агентов. Агент-интерфейс МИС организует взаимодействие для извлечения метаданных о музейных экспонатах. Агент поиска организует взаимодействие с базами знаний для наполнения семантической сети информационными описаниями экспонатов и связанных сущностей. В результате работы этих двух агентов строится семантическая сеть музейной коллекции. Агент ранжирования выполняет анализ

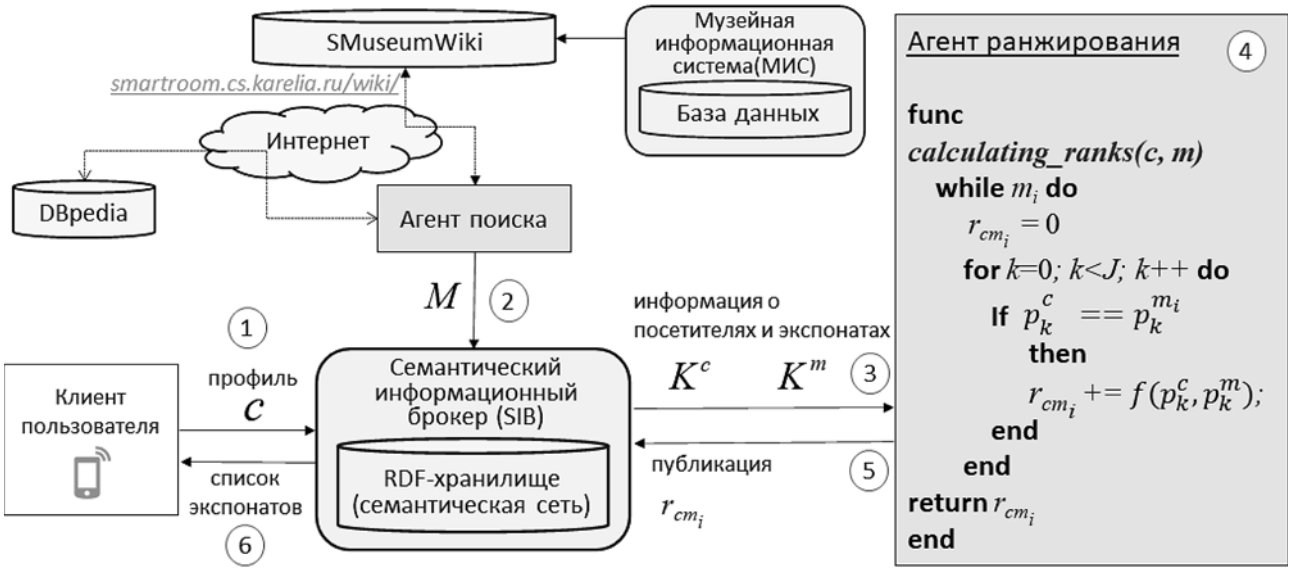


Рис. 4. Архитектурная схема системы локального ранжирования экспонатов

семантической сети и в зависимости от заданных параметров формирует персонализированные рекомендации.

Рассмотрим формальные показатели качества локального ранжирования. Известны ранги между заданным объектом c и экспонатами $m \in M$. Пусть R — множество рангов $r_{cm_i} \in R$. Релевантность определяется рангом экспоната. Множество экспонатов можно разделить на три кластера: наиболее релевантные, слабо релевантные и нерелевантные экспонаты. Обозначим R_{top} — кластер наиболее релевантных экспонатов, $r_i \in R_{top}$, $R_{top} \in R$. Тогда n_{top} — размерность кластера R_{top}. Обозначим R_{rest} — кластер слабо релевантных экспонатов, $r_j \in R_{rest}$, $R_{rest} \in R$; n_{rest} — размерность кластера R_{rest}. Экспонаты с нулевым значением ранга относятся к кластеру нерелевантных. Тогда $n_{rel} = n_{top} + n_{rest}$ — число экспонатов с ненулевым значением ранга.

Расстояние между двумя кластерами представляет разность между средними значениями релевантных классов и вычисляется по формуле

$$\varepsilon = r_{top}^{avg} - r_{rest}^{avg}. \quad (5)$$

Долю наиболее релевантных экспонатов можно вычислить по формуле

$$T = \frac{n_{top}}{n_{top} + n_{rest}}. \quad (6)$$

Значение параметра $0 < T < 1$ возрастает с увеличением числа наиболее релевантных экспонатов n_{top} и достигает единицы, когда все экспонаты с ненулевым рангом включены в кластер наиболее релевантных $n_{rel} = n_{top}$.

Относительная оценка общей релевантности среди найденных экспонатов вычисляется по формуле

$$C = \frac{r_{top}^{avg}}{r_{top}^{avg} + r_{rest}^{\text{avg}}}. \quad (7)$$

Относительная оценка общей релевантности обозначает покрытие наиболее релевантных экспонатов относительно экспонатов с ненулевым значением ранга, причем $0,5 < C < 1$. При $C = 1$ все экспонаты с ненулевым рангом включены в кластер наиболее релевантных $n_{rel} = n_{top}$.

Для каждого кластера можно подсчитать суммарный ранг группы, $\sum_{i=0}^{n_{top}} r_i$ и $\sum_{j=0}^{n_{rest}} r_j$ соответственно. Оптимальное разбиение можно получить, вычислив абсолютную разницу между суммарными рангами релевантной и менее релевантной групп. Наименьшая разница указывает на оптимальное разбиение:

$$\left| \sum_{i=0}^{n_{top}} r_i - \sum_{j=0}^{n_{rest}} r_j \right| \to \min. \quad (8)$$

Выполнены начальные эксперименты с реализацией локального ранжирования экспонатов в соответствии с предложенной моделью. В базе знаний [8] представлена семантическая сеть физических и цифровых экспонатов, связанных с ними исторических событий, персон и других объектов из области истории повседневности ПетрГУ. Всего в ранжировании принимали участие 296 экспонатов. В эксперименте участвуют профили пользователей — выпускников ПетрГУ разных лет и факультетов. Чем более подробно заполнен профиль пользователя, тем подобранные экспонаты будут больше и точнее связаны с пользователем. Описания четырех профилей представлены в табл. 4 (прочерк означает, что пользователь не указал данные). Один пользовательский профайл сопоставляется с каждым экспонатом согласно алгоритму локального ранжирования. В результате семантического сопоставления формируется матрица рангов, по строкам которой перечислены пользователи, по столбцам — экспонаты. Для оценки локального ранжирования необходимо проанализировать строку матрицы рангов: провести кластеризацию полученных рангов и экспертно проверить соответствие рекомендаций потребностям пользователей.

Сортировка каждой строки матрицы рангов дает группу экспонатов с наибольшим рангом, а остальные экспонаты формируют менее релевантную группу. Экспонаты со значением ранга, равным нулю, в ранжировании не участвуют. Размер получаемой группы релевантных экспонатов формально харак-

Таблица 4

Описание пользователей

Пользователь	Факультет	Годы обучения	Хобби	Памятные места	Группы
1	Математический	2009—2013	—	—	—
2	Математический	2010—2016	Занятия спортом, лыжи	с. Паданы	Отряд Goliard, Отряд Альфа, Отряд Аякс
3	Промышленного и гражданского строительства	1980—1985	Волейбол, турклуб Сампо	г. Петрозаводск	Отряд Goliard, Отряд Альфа, Отряд Гренадеры
4	Лесоинженерный	1983—1988	—	п. Вяртсиля	—

Таблица 5

Варианты выбора релевантной группы для различных пользователей

Пользователь	n_{top}	r_{top}^{avg}	n_{rest}	r_{rest}^{avg}	Критерий (5)	Критерий (6)	Критерий (7)	$\sum_{i=0}^{n_{top}} r_i$	$\sum_{j=0}^{n_{rest}} r_j$	Критерий (8)
1 $n_{rel} = 87$	2	2,500	85	0,835	1,665	0,023	0,750	5,0	71,0	66,0
	6	2,167	81	0,778	1,389	0,069	0,736	13,0	63,0	50,0
	11	**1,864**	**76**	**0,730**	**1,133**	**0,126**	**0,718**	**20,5**	**55,5**	**35,0**
	46	1,207	41	0,500	0,707	0,529	0,707	55,5	20,5	35,0
2 $n_{rel} = 99$	1	3,500	98	1,173	2,327	0,010	0,749	3,5	115,0	111,5
	3	2,833	96	1,146	1,688	0,030	0,712	8,5	110,0	101,5
	17	2,147	82	1,000	1,147	0,172	0,682	36,5	82,0	45,5
	21	**2,024**	**78**	**0,974**	**1,049**	**0,212**	**0,675**	**42,5**	**76,0**	**33,5**
	95	1,226	4	0,500	0,726	0,960	0,710	116,5	2,0	114,5
3 $n_{rel} = 160$	5	2,500	155	0,961	1,539	0,031	0,722	155	149,0	136,5
	11	2,227	149	0,919	1,308	0,069	0,708	149	137,0	112,5
	61	**1,631**	**99**	**0,626**	**1,005**	**0,381**	**0,723**	**99**	**62,0**	**37,5**
	86	1,448	74	0,500	0,948	0,538	0,743	74	37,0	87,5
4 $n_{rel} = 26$	4	2,000	22	0,864	1,136	0,154	0,698	22	19,0	11,0
	6	**1,833**	**20**	**0,800**	**1,033**	**0,231**	**0,696**	**20**	**16,0**	**5,0**
	18	1,278	8	0,500	0,778	0,692	0,719	8	4,0	19,0

теризует качество ранжирования: число рекомендуемых экспонатов не должно быть велико.

Распределение значений ранга для выбранных пользователей представлено на рис. 5 (см. третью сторону обложки). Результат ранжирования разбит по значению ранга. Можно увидеть, что экспонатов с наибольшим рангом небольшое число.

В табл. 5 представлены сводные экспериментальные данные выполнения локального ранжирования. Во втором столбце указано число экспонатов в кластере — наиболее релевантных, а в четвертом кластере — менее релевантных. Для каждого распределения считается суммарный ранг кластера. Абсолютное значение разницы между суммарным значением ранга указано в последнем столбце. Наименьшее значение разницы указывает на оптимальное распределение экспонатов на наиболее релевантные и менее релевантные относительно общего количества экспонатов с ненулевым рангом. Оптимальное разбиение по критерию (8) выделено полужирным шрифтом.

Например, для пользователя 1 по критерию оптимальности (8) рекомендовано 11 экспонатов с наибольшим значением ранга из 87 релевантных экспонатов. Расстояние между двумя кластерами при таком распределении составляет 1,133. В таком распределении доля наиболее релевантных экспонатов составляет 0,126, т. е. является незначительной относительно общего количества релевантных экспонатов. Относительная оценка общей релевантности составляет 0,718 и указывает на покрытие наиболее релевантных экспонатов относительно экспонатов с ненулевым значением ранга. Таким образом, размер группы наиболее релевантных экспонатов является небольшим.

В соответствии с разработанной моделью предложен алгоритм локального ранжирования экспонатов в семантической сети. Алгоритм реализован в программной многоагентной системе сервисов персонализированного сопровождения деятельности музея. Предложены формальные показатели качества ранжирования. Выполнены начальные эксперименты с реализацией алгоритма локального ранжирования экспонатов в сервисе изучения музейного фонда Музея Истории ПетрГУ.

Заключение

Представлена математическая модель локального ранжирования экспонатов в семантической сети для решения трех задач информационного сопровождения деятельности музея: задача выявления экспонатов, задача навигации, задача вывода новых знаний. Семантическая сеть строится на основе разработанной онтологической модели для семантического представления информации музейных коллекций. Решение задач сопровождения деятельности музея

было рассмотрено на примере цифровых сервисов в Музее Истории Петрозаводского государственного университета. Модель локального ранжирования реализована в виде алгоритма в цифровых сервисах музея. Алгоритм ранжирования вычисляет оценки релевантности экспонатов на основе заданных критериев, представленных в профиле пользователя. Выполнены экспериментальные исследования алгоритма локального ранжирования в соответствии с разработанной моделью оценки результатов ранжирования.

Научные результаты получены при финансовой поддержке проекта РФФИ № 19-07-01027. Статья подготовлена в рамках реализации Программы развития опорного университета для Петрозаводского государственного университета на 2017—2021 гг.

Список литературы

1. **Amato F., Moscato V., Picariello A.** et al. Big data meets digital cultural heritage: Design and implementation of scrabs, a smart context-aware browsing assistant for cultural environments// J. Comput. Cult. Herit. — 2017. — No. 10 (1). — P. 6:1—6:23.

2. **Поврозник Н. Г.** Виртуальный музей: сохранение и репрезентация историко-культурного наследия // Вестник Пермского Университета. История. — 2015. — № 4 (31). — С. 213—221.

3. **Kuflik T., Wecker A., Lanir J., Stock O.** An integrative framework for extending the boundaries of the museum visit experience: linking the pre, during and post visit phases // Information Technology & Tourism. — 2015. — No. 15 (1). — P. 17—47.

4. **Bowen J. P., Filippini-Fantoni S.** Personalization and the web from a museum perspective / D. Bearman and J. Trant (Eds.) // Museums and the Web 2004: Selected Papers from an International Conference. — Arlington, Virginia, USA, 31 March — 3 April, 63—78. Archives & Museum Informatics. URL: www.archimuse.com/mw2004/papers/bowen/bowen.html

5. **Ruotsalo T., Haav K., Stoyanov A.** et al. SMARTMUSEUM: A mobile recommender system for the Web of Data // Web Semantics: Science, Services and Agents on the World Wide Web 20. — 2013. — P. 50—67.

6. **Aroyo L., Brussee R., Rutledge L.** et al. Personalized Museum Experience: The Rijksmuseum Use Case // Museums and the Web. April 11-14, 2007 San Francisco, California. URL: https://www.museumsandtheweb.com/mw2007/papers/aroyo/aroyo.html

7. **Яловицына С. Э., Волохова В. В., Корзун Д. Ж.** Семантический подход к представлению информации музейных тематических коллекций // Вестник архивиста. — 2019. — № 1. — С. 235—246. DOI 10.28995/2073-0101-2019-1-235-246.

8. **Korzun D., Yalovitsyna S., Volokhova V.** Smart Services as Cultural and Historical Heritage Information Assistance for Museum Visitors and Personnel // Baltic Journal of Modern Computations. — 2018. — Vol. 6, No. 3. — P. 632—643.

9. **Petrina O. B., Korzun D. G., Volokhova V. V.** et al. Semantic approach to opening museum collections of everyday life history for services in Internet of Things environments // International Journal of Embedded and Real-Time Communication Systems. — 2017. — Vol. 8, No. 1. — P. 31—44.

10. **Петрина О. Б., Корзун Д. Ж.** Модель локального ранжирования для разработки информационных сервисов рекомендации экспонатов в умном музее // Цифровые технологии в образовании, науке, обществе: материалы XII всероссийской научно-практической конференции. 4—6 декабря 2018 г. — Петрозаводск, 2018. — С. 176—179.

11. **Варфоломеев А. Г., Иванов А. С.** Компьютерное источниковедение: семантическое связывание информации в репрезентации и критике исторических источников. — Петрозаводск: Изд-во ПетрГУ, 2013. — 204 с.

12. **Wang Y., Stash N., Aroyo L.** et al. Recommendations based on semantically enriched museum collections // Journal of Web Semantics: Science, Services and Agents on the World Wide Web. — 2008. — Vol. 6, No. 4. — P. 283—290. DOI: 10.1016/j.websem.2008.09.002.

13. **Печенкин В. В., Королев М. С., Димитров Л. В.** Прикладные аспекты использования алгоритмов ранжирования для ориентированных взвешенных графов (на примере графов социальных сетей) // Труды СПИИРАН. — 2018. — № 6 (61). DOI: 10.15622/sp.61.4.

14. **Свидетельство** о государственной регистрации базы данных 2017621001 Российская Федерация. Онтологическая база знаний истории повседневности Петрозаводского государственного университета / О. Б. Петрина и др.; правообладатель Петрозавод. гос. ун-т. № 2017621001; заявл. 03.07.2017; зарегистр. 01.09.2017.

15. **Fogli A., Sansonetti G.** Exploiting semantics for context-aware itinerary recommendation // Personal and Ubiquitous Computing. — 2019. — Vol. 23, No. 2. — P. 215—231. DOI: 10.1007/s00779-018-01189-7.

16. **Лапшин В. А.** Онтологии в информационных системах. Сервер учебных материалов ОИС РГГУ. — М., 2009. — 247 с. URL: http://isdwiki.rsuh.ru/moodle/pluginfile.php/128/course/section/36/bookLapshin.pdf

17. **Марченков С. А.** Автоматизация процессов программирования агентов на основе кодогенерации при построении семантических сервисов интеллектуальных пространств. Часть 1 // Программная инженерия. − 2019. — Т. 10, № 6. — С. 257—264. DOI: 10.17587/prin.10.257—264.

Information Ranking Based on a Semantic Network or Constructing Digital Services for Personalized Support of Museum Activities

O. B. Petrina, petrina@cs.petrsu.ru, Petrozavodsk State University, Petrozavodsk, 185910, Russian Federation

Corresponding author:

Petrina Oksana B., Junior Researcher, Petrozavodsk State University, Petrozavodsk, 185910, Russian Federation
E-mail: petrina@cs.petrsu.ru

Received on June 23, 2020
Accepted on July 27, 2020

This paper explores the tasks of personalized support of the museum, for which it is proposed to use digital services. Services implement the selection of information sources using an additional semantic level. The semantic level is created above the catalog-oriented information storage of the museum. An ontological model has been developed, in accordance with which information is presented in the form of a semantic network. The local ranking model of semantic network objects is proposed, on the basis of which a ranking algorithm is developed. The software implementation of the local ranking algorithm in digital services of personalized support is presented on the example of the Museum of History of Petrozavodsk State University. The services provide personalized recommendations to museum visitors when studying the exposition. It requires various options for ranking exhibits and selecting exhibits with the highest rank.

Keywords: *ranking, semantic network, digital services, personalization, cultural and historical heritage*

Acknowledgements: *The research was financially supported by RFBR (research project № 19-07-01027). The work is implemented within the Government Program of Flagship University Development for Petrozavodsk State University (PetrSU) in 2017-2021.*

For citation:

Petrina O. B. Information Ranking Based on a Semantic Network for Constructing Digital Services for Personalized Support of Museum Activities, *Programmnaya Ingeneria,* 2020, vol. 11, no. 5, pp. 296—308

DOI: 10.17587/prin.11.296-308

References

1. **Amato F., Moscato V., Picariello A., Colace F., Santo M. D., Schreiber F. A., Tanca L.** Big data meets digital cultural heritage: Design and implementation of scrabs, a smart context-aware browsing assistant for cultural environments, *J. Comput. Cult. Herit.*, 2017, vol. 10, no. 1, pp. 6:1—6:23.

2. **Povroznik N. G.** Virtual Museum: Preservation and Representation of Historical and Cultural Heritage, *Vestnik Permskogo Universiteta. Istoriya,* 2015, no. 4 (31), pp. 213—221 (in Russian).

3. **Kuflik T., Wecker A., Lanir J., Stock O.** An integrative framework for extending the boundaries of the museum visit experience: linking the pre, during and post visit phases, *Information Technology & Tourism*, 2015, vol. 15, no. 1, pp. 17—47.

4. **Bowen J. P., Filippini-Fantoni S.** Personalization and the web from a museum perspective / D. Bearman and J. Trant (Eds.), *Museums and the Web 2004: Selected Papers from an International Conference,* Arlington, Virginia, USA, 31 March — 3 April, 2004, pp. 63—78. Archives & Museum Informatics, available at: www.archimuse.com/mw2004/papers/bowen/bowen.html

5. **Ruotsalo T., Haav K., Stoyanov A., Roche S., Fani E., Deliai R., Makela E., Kauppinen T., Hyvonen E.** SMARTMUSEUM: A mobile recommender system for the Web of Data, *Web Semantics: Science, Services and Agents on the World Wide Web 20*, 2013, pp. 50—67.

6. **Aroyo L., Brussee R., Rutledge L., Gorgels P., Stash N., Wang Y.** Personalized Museum Experience: The Rijksmuseum Use Case, *Museums and the Web*, April 11—14, 2007, San Francisco, California, available at: https://www.museumsandtheweb.com/mw2007/papers/aroyo/aroyo.html

7. **Yalovicyna S. E., Volohova V. V., Korzun D. G.** Semantic Approach to Presenting Information of Thematic Museum Collections, *Vestnik arhivista*. 2019, no. 1, pp. 235—246. DOI: 10.28995/2073-0101-2019-1-235-246/ (in Russian).

8. **Korzun D., Yalovitsyna S., Volokhova V.** Smart Services as Cultural and Historical Heritage Information Assistance for Museum Visitors and Personnel, *Baltic Journal of Modern Computations*, 2018, vol. 6, no. 3, pp. 632—643.

9. **Petrina O. B., Korzun D. G., Volokhova V. V., Yalovitsyna S. E., Varfolomeyev A. G.** Semantic approach to opening museum collections of everyday life history for services in Internet of Things environments, *International Journal of Embedded and Real-Time Communication Systems*, 2017, vol. 8, no. 1, pp. 31—44.

10. **Petrina O. B., Korzun D. G.** Model' lokal'nogo ranzhirovaniya dlya razrabotki informacionnyh servisov rekomendacii eksponatov v umnom muzee, *Cifrovye tekhnologii v obrazovanii, nauke, obshchestve: materialy XII vserossijskoj nauchno-prakticheskoj konferencii,* 4—6 December 2018, Petrozavodsk, 2018, pp. 176—179 (in Russian).

11. **Varfolomeyev A. G., Ivanov A. S.** *Komp'yuternoe istochnikovedenie: semanticheskoe svyazyvanie informacii v reprezentacii i kritike istoricheskih istochnikov*, Petrozavodsk, Izd-vo PetrGU, 2013, 204 p. (in Russian).

12. **Wang Y., Stash N., Aroyo L., Gorgels P., Rutledge L., Schreiber G.** Recommendations based on semantically enriched museum collections, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2008, vol. 6, no. 4, pp. 283—290. DOI: 10.1016/j.websem.2008.09.002.

13. **Pechenkin V. V., Korolyov M. S., Dimitrov L. V.** Applied Aspects of Ranking Algorithms for Oriented Weighted Graphs (on the Example of Social Network Graphs), *Trudy SPIIRAN*, 2018, no. 6 (61). DOI: 10.15622/sp.61.4 (in Russian).

14. **Svidetelstvo** o gosudarstvennoj registracii bazy dannyh 2017621001 Rossijskaya Federaciya. Ontologicheskaya baza znanij istorii povsednevnosti Petrozavodskogo gosudarstvennogo universiteta / Petrina O. B. i dr.; pravoobladatel PetrSU. № 2017621001; zayavl. 03.07.2017; zaregistr. 01.09.2017 (in Russian).

15. **Fogli A., Sansonetti G.** Exploiting semantics for context-aware itinerary recommendation, *Personal and Ubiquitous Computing,* 2019, vol. 23, no. 2, pp. 215—231. DOI: 10.1007/s00779-018-01189-7.

16. **Lapshin V. A.** *Ontologii v informacionnyh sistemah*, Server uchebnyh materialov OIS RGGU. Moscow, 2009, 247 p., available at: http://isdwiki.rsuh.ru/moodle/pluginfile.php/128/course/section/36/bookLapshin.pdf (in Russian).

17. **Marchenkov S. A.** Computer-aided programming of software agents based on code generation in constructing semantic services of smart spaces. Part 1, *Programmnaya ingeneria*, 2019, vol. 10, no. 6, pp. 257—264, DOI: 10.17587/prin.10.257-264 (in Russian).

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

Сдано в набор 12.08.2020 г. Подписано в печать 24.09.2020 г. Формат 60×88 1/8. Заказ PI520
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 **О. М. Гулиной, Н. Л. Сальникова, В. П. Семишкина, М. Н. Тишикиной**

«РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММ ДЛЯ УПРАВЛЕНИЯ РЕСУРСОМ МЕХАНИЧЕСКИХ ЭЛЕМЕНТОВ АЭС В УСЛОВИЯХ ЭРОЗИОННО-КОРРОЗИОННОГО ИЗНОСА»

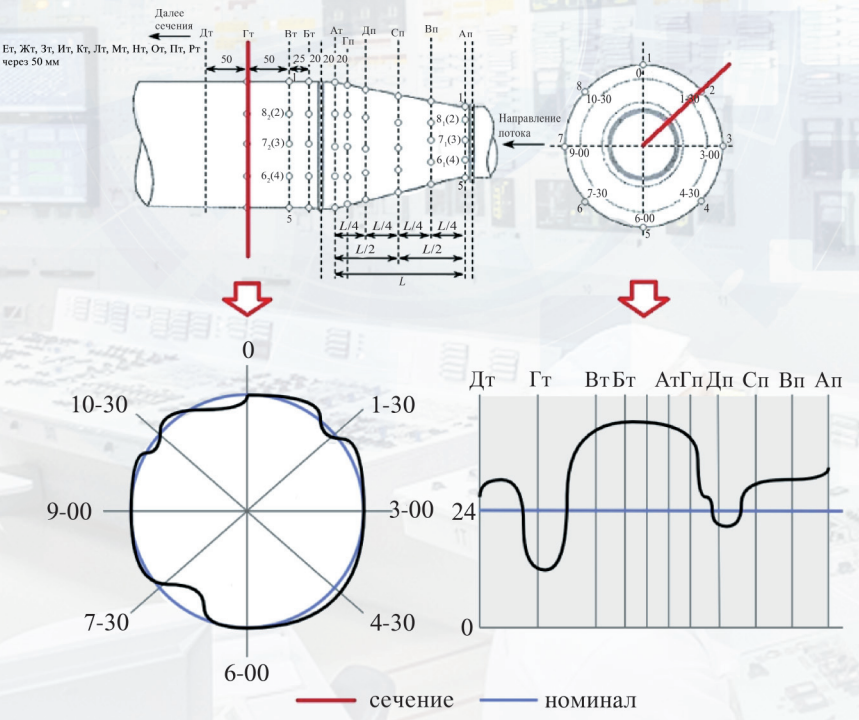
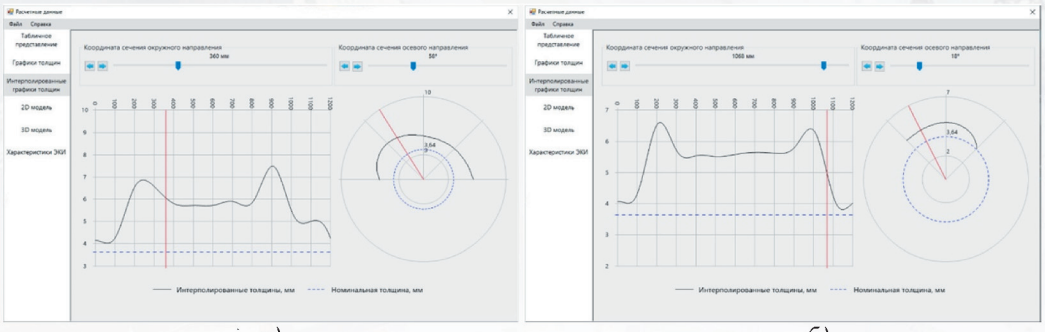


Рис. 7. Концептуальная схема модуля «Интерполированные графики толщин»



а) *б)*

Рис. 9. Результат работы модуля «Интерполированные графики толщин» для гиба:
а – пять точек; *б* – три точки

Рисунок к статье **О. Б. Петриной**

«РАНЖИРОВАНИЕ ИНФОРМАЦИИ НА ОСНОВЕ СЕМАНТИЧЕСКОЙ СЕТИ ПРИ ПОСТРОЕНИИ ЦИФРОВЫХ СЕРВИСОВ ПЕРСОНАЛИЗИРОВАННОГО СОПРОВОЖДЕНИЯ ДЕЯТЕЛЬНОСТИ МУЗЕЯ»

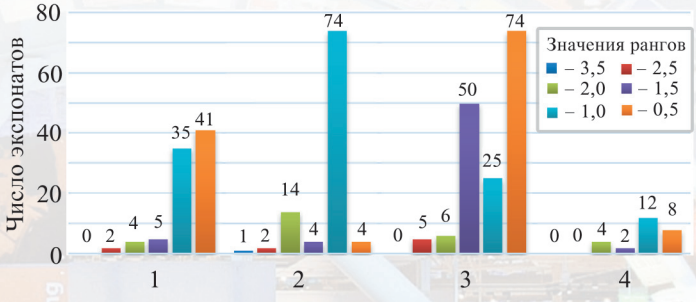


Рис. 5. Распределение значений рангов экспонатов для четырех пользователей

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

выпускает *научно-технические журналы*

Теоретический и прикладной научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях челове ческой деятельности.

Подписной индекс по Объединенному каталогу «Пресса России» – 22765

Подписной индекс по Объединенному каталогу «Пресса России» – 72656

Ежемесячный теоретический и прикладной научно-технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Междисциплинарный теоретический и прикладной научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопрос разработки и внедрения нанои микросистем в различные области науки, технологии и производства.

Подписной индекс по Объединенному каталогу «Пресса России» – 79493

Подписной индекс по Объединенному каталогу «Пресса России» – 79492

Ежемесячный теоретический и прикладной научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автомати ческого и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Научно-практический и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.

Подписной индекс по Объединенному каталогу «Пресса России» – 79963

Адрес редакции журналов для авторов и подписчиков:
107076, Москва, Стромынский пер., 4. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: (499) 269-55-10, 269-53-97. Факс: (499) 269-55-10. E-mail: antonov@novtex.ru

ISSN 2220-3397. Программная инженерия. 2020. Том 11. № 5. 257 – 308. Подписной индекс по Объединенному каталогу "Пресса России" – 22765