УДК 001, 004.4 DOI: 10.17587/prin.8.448-455

В. А. Васенин, д-р физ.-мат. наук, проф., зав. каф., **М. А. Занчурин,** мл. науч. сотр.,
А. С. Козицын, канд. физ.-мат. наук, вед. науч. сотр., **М. А. Кривчиков,** канд. физ.-мат. наук,
ст. науч. сотр., e-mail: maxim.krivchikov@gmail.com, **Д. А. Шачнев,** аспирант,
МГУ имени М. В. Ломоносова

Архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования

Механизмы управления научно-технической и образовательной деятельностью являются предметом особого внимания государства. С появлением технологий и средств пакетных коммуникаций, со становлением метасети Интернет и информационных технологий на ее основе, к началу XXI века сложились предпосылки создания систем, способных оперативно собирать, хранить и обрабатывать большие объемы наукометрических данных. В настоящей статье рассмотрены архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования на примере информационно-аналитической системы "ИСТИНА".

Ключевые слова: *информационно-аналитические системы, наука и образование, наукометрия, инженерия программ, распределенные системы, архитектура программного обеспечения*

Механизмы управления научно-технической и образовательной деятельностью являются предметом особого внимания любого государства [1]. Причина в том, что от эффективности таких механизмов напрямую зависят темпы научно-технического прогресса и, как следствие, показатели развития национальной экономики [2]. К числу внешних факторов, которые определяют развитие науки и образования, как правило, относят факторы политического, социального и экономического характера. Именно они формируют базовые предпосылки, без которых эффективное управление наукой и образованием невозможно. Не менее значимыми в вопросах такого управления являются факторы внутренние. Такими факторами являются наличие: единства и преемственности между различными стадиями образовательного процесса (школа — вуз — подготовка кадров высшей квалификации); высокого уровня развития издательского дела и других форм оперативного и широкого обмена научными идеями и результатами (семинары, конференции, симпозиумы) ученых и педагогов; регуляторов адекватной оценки и поощрения ученых, инженеров-инноваторов и педагогов.

К числу индикаторов, на основе которых реализуются последние из перечисленных факторов, относятся:

- количественные и качественные показатели "выпускаемой продукции", включая публикации и патенты, число подготовленных кадров различной квалификации и т. п.;
- объемы педагогической деятельности в виде прочитанных курсов лекций и проведенных научных семинаров;
- объемы и качество выполненных научно-исследовательских и опытно-конструкторских работ.

Эти индикаторы и построенные на их основе механизмы регулирования известны и активно использовались в XX веке. Однако отсутствие возможностей сбора и анализа таких показателей в большом объеме, а также их оперативной обработки значительно ограничивало действенность используемых регуляторов.

С появлением технологий и средств пакетных коммуникаций, со становлением метасети Интернет и информационных технологий на ее основе, к началу XXI века сложились предпосылки создания систем, способных оперативно собирать, хранить и обрабатывать большие объемы наукометрических данных. В начале 2000-х гг. появились центры индексирования и анализа библиометрических данных за рубежом — Web of Science (Thomson & Reuters), Scopus (Elsevier), Google Scholar, а также РИНЦ в России. На основе анализа коллекций таких данных стали создаваться информационно-аналитические системы для анализа эффективности (рейтинговых оценок) функционирования субъектов научно-технической деятельности. К их числу относятся Pure, SciVal (Elsevier), Converis (Thomson & Reuters), которые принято именовать Current Research Information Systems — CRIS.

Системы подобного назначения стали востребованы и в России. Многие из создаваемых для этих целей систем ориентированы на сбор и хранение, обработку и анализ данных отдельных вузов и научных центров. В качестве инструментальных средств для их создания и сопровождения использовались либо упомянутые выше зарубежные системы, либо си-

стемы, разрабатываемые в отдельных организациях под собственные, локальные потребности. В качестве проектов создания систем, которые изначально декларировали своей целью ориентацию на различные уровни управления наукой и образованием в России (от отдельных вузов и научных центров до регионального и национального), можно выделить "Карту российской науки" [3, 4] и информационно-аналитическую систему (ИАС) "ИСТИНА" (Интеллектуальная Система Тематического Исследования НАукометрических данных) [6].

Принимая во внимание отмеченную выше высокую востребованность подобных информационно-аналитических систем, их важную роль в управлении отечественной наукой и образованием, в настоящей статье рассматриваются архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования на примере ИАС "ИСТИНА". Далее применительно к ИАС "ИСТИНА" используется термин Система, что позволяет отделить ее от других упомянутых в тексте систем, в том числе — от используемых в составе ИАС "ИСТИНА" систем.

Общие принципы построения Системы

Теоретическая сторона (математические модели), а также алгоритмика и программные механизмы реализации и вопросы практической автоматизации отдельных бизнес-процессов в области наукометрии достаточно подробно рассмотрены в известных публикациях [4—13]. Однако не менее важными являются архитектурно-технологические особенности создания и сопровождения систем такого рода, а также вытекающие из них требования.

Основные требования к архитектуре Системы приведены далее.

1. Архитектура Системы должна быть модульной и масштабируемой как на макроуровне ее описания, так и на нижележащих уровнях. Такая структура направлена на адекватное отображение процессов, подлежащих автоматизации.

2. Архитектура Системы должна быть иерархически организована (структурирована) и процессно-ориентирована на каждом из уровней структурной иерархии, включая следующие блоки на отдельных уровнях:

- функционально замкнутые блоки (подсистемы) Системы, ориентированные на автоматизированную реализацию макропроцессов, реализующих функции в рамках этого блока;
- функционально обособленные компоненты (модули, приложения) в составе отдельных блоков Системы, ориентированные на реализацию соответствующих этому компоненту функций (процессов);
- компонент Системы, реализующий функции монитора безопасности для разграничения доступа к отдельным модулям Системы, к базам данных и отдельным данным в таких базах;
- компонент Системы, реализующий связные (передача данных) и интегрирующие функции в Системе.

3. Архитектура Системы должна с достаточной полнотой отражать все автоматизируемые в рамках Системы процессы, которые востребованы практикой наукометрии, а также задачи, поставленные в настоящее время и на прогнозируемую перспективу в сфере подготовки к принятию управленческих решений в этой области.

4. Архитектура Системы должна учитывать процессы взаимодействия (обмен данными, запросы и т. п.) с базами различного рода вспомогательных данных, дополняющих, актуализирующих и конкретизирующих (уточняющих) данные, которыми располагает собственно Система, имея в виду как внутренние, так и внешние по отношению к организации базы.

При разработке, развитии и модернизации ИАС "ИСТИНА", при ее сопровождении по назначению соблюдаются перечисленные выше архитектурные принципы и вытекающие из них общие требования к Системе. Они реализуются в том объеме, который адекватен текущему уровню реализации настоящего проекта.

Далее на примере ИАС "ИСТИНА" кратко сформулируем технологические принципы создания и развития Систем, аналогичных ей по назначению и условиям использования.

1. Соблюдение положений нормативных документов РФ применительно к созданию, эксплуатации и развитию национально значимых систем, в том числе с учетом перспектив использования отечественного программного обеспечения.

2. Соблюдение основных положений и рекомендаций к инженерии программ на всех этапах жизненного цикла Системы. В настоящее время в жизненном цикле отдельных компонентов ИАС "ИСТИНА" предусмотрены и реализуются: предпроектные исследования; оценка эффективности предлагаемых решений; проектирование; программная реализация; тестирование и т. п. Как правило, каждому ресурсоемкому компоненту в ИАС "ИСТИНА" и Системе в целом соответствуют: модели; алгоритмы; документация в формате Единой системы программной документации (ЕСПД); код с комментариями.

3. При вторичной (по запросу) обработке данных Система должна в максимально возможной степени использовать механизмы взаимодействия с базами данных и содержащимися в них данными (в том числе с точки зрения их конфиденциальности), которые необходимы для выполнения запроса.

Это требование позволит активно использовать данные как "де-факто" существующих баз данных, так и вновь разрабатываемых без ущерба для конфиденциальности части данных в этих базах. Здесь следует отметить то обстоятельство, что существующие и используемые на практике модели и программные средства обеспечения информационной безопасности (в первую очередь, разграничения доступа к ресурсам) таких сложно организованных систем не могут удовлетворить требованиям, которые следуют из положений этого принципа. Однако определенные результаты исследований и практической реализации на этом направлении из-

вестны, в частности, модель, описанная в работе [4], разработанная непосредственно на основе опыта реализации компонентов разграничения доступа в Системе.

4. Механизмы (математическое, алгоритмическое и программное обеспечение) должны учитывать различные уровни конфиденциальности данных, которыми располагает Система, и те базы данных, к которым она может обращаться.

Механизмы логического разграничения доступа к данным в ИАС "ИСТИНА" опираются на формальные модели их описания, которые отличаются от традиционно принятых в "классической" информационной безопасности, аккумулируют мировой опыт создания такого сорта механизмов в социальных сетях и учитывают особенности наукометрии как проблемной области.

5. Система должна поддерживать интеграционные механизмы, позволяющие извлекать необходимые данные по запросу пользователей из других, в том числе, удаленных в сети Интернет баз данных (БД), с их защитой от несанкционированного доступа и верификацией, с установлением их соответствия ("привязкой" или "аффилиацией") к отдельным персонам (далее — персоналиям) в БД Системы.

В настоящее время с той или иной степенью завершенности в ИАС "ИСТИНА" реализованы следующие механизмы интеграции данных:

- механизмы обмена информацией по телекоммуникационным каналам между пользователями Системы и базами данных (БД внутренними и внешними по отношению к Системе) с использованием единой модели логического разграничения доступа к ресурсам Системы;
- механизмы ввода данных в Систему в режимах "снизу вверх" (от конечного пользователя) и "сверху вниз" (из ранее сформированных источников) с их первичной обработкой, верификацией, "привязкой" к персоналиям и размещением в БД Системы под контролем модели логического разграничения доступа;
- механизмы оперативного вывода результатов вторичной обработки и агрегирования данных по запросам пользователей Системы (включая отдельных персоналий, ответственных лиц от структурных подразделений и ректората, отвечающих за сопровождение данных в Системе) под контролем модели логического разграничения доступа.

Архитектура системы

В настоящем разделе представлены основные положения архитектуры ИАС "ИСТИНА", которые реализуют на практике требования и принципы, изложенные в предыдущем разделе.

Архитектура Системы отражает онтологическую модель наукометрии в ее представлении как предметной области, на которую Система ориентирована. Основные понятия (сущности, объекты), которые используются в приложениях наукометрии и составляют ее тезаурус, систематизированы, едины для всех приложений Системы и хранятся в ее БД. Отношения (связи) между этими объектами, составляющие таксономию нижнего уровня наукометрии, реализуются реляционными механизмами базы данных. Приложения как объекты верхнего уровня таксономии наукометрии (проблемной области) взаимодействуют между собой по заранее принятым правилам, которые реализуются с помощью механизмов шаблона проектирования "модель—представление—поведение" (MVC) и механизмов объектно-реляционного отображения (ORM). При этом используются единообразные объекты и отношения между ними. Таким образом, естественно реализуются механизмы интеграции используемых в Системе данных и приложений.

Интегрирующие механизмы Системы реализованы на основе реляционной БД, структура которой отражает основные аспекты онтологии, а также взаимодействующих с ней приложений. Понятия, используемые Системой, представлены в виде отношений реляционной БД. В коде Системы использованы средства объектно-реляционного отображения, которые представляют понятия системы в виде набора объектов и связей между ними.

Основу Системы составляет ядро — набор программных модулей и подсистем, реализующих базовые (системообразующие) функциональные возможности, которые напрямую (во многом) определяют и качество (показатели качества) Системы в целом. Ядро содержит базовые классы, методы, формы и функции, используемые в других модулях. Оно обеспечивает единый интерфейс и базовый каркас добавления, просмотра, редактирования и удаления результатов научно-педагогической деятельности работников. В ядре реализованы механизмы поиска похожих объектов, которые используются для подбора, в частности, похожих работников, журналов и статей. В ядре реализуются также и другие общие функциональные возможности, например, поддерживающие политику безопасности Системы, в частности, механизмы разграничения доступа к различным категориям данных Системы. С позиций описания модели предметной области, ядро Системы содержит базовые классы, определяющие следующие понятия: результат научной или педагогической деятельности (результат); авторство результата или другая связь, описывающая отношение работника к результату (например, "официальный оппонент диссертации").

На следующем уровне архитектуры расположены два базовых приложения: "организации" и "работники", а также подсистема логического разграничения доступа. Модель данных приложения "организация" описывает административную структуру как организации в целом, так и ее структурных подразделений. В состав этого приложения входит также механизм реализации действий должностных лиц, которым делегирована роль ответственных за сопровождение информации в Системе от организации и отдельных ее структурных подразделений. Приложение "работники" описывает понятия, связанные с работниками, включая следующие: профессиональный профиль работника в Системе; аффилиация работника

с организацией и ее структурным подразделением (место работы и должность); сведения об ученой степени и ученом звании работника. Приложения "организации" и "работники" являются базовыми приложениями потому, что с ними связаны все остальные приложения Системы, которые находятся выше по иерархии. Подсистема логического разграничения доступа реализует реляционную модель логического разграничения доступа к объектам системы на основе уже имеющихся в системе отношений.

Основным уровнем архитектуры являются приложения класса "результаты деятельности". Каждое приложение описывает структуру понятий, связанных с отдельным типом результатов научно-инновационной и педагогической деятельности работников (например, публикации), или общую сущность, связанную с таким типом (например, журналы). Эти приложения опираются на общий каркас базовых понятий, классов и функций Системы, которые представлены в ее ядре. На следующем уровне иерархии архитектуры Системы расположены подсистемы, отвечающие за анализ данных, накопленных на предыдущих уровнях, и за их интеграцию с данными, полученными из внешних систем. К таким подсистемам относятся: подсистема верификации данных; подсистема расчета персонального рейтинга; подсистема подготовки отчетных материалов. Подсистема верификации включает механизмы получения и обработки показателей цитирования отдельных статей из Web of Science, из Scopus, а также поиск статей в этих системах. На этом же уровне иерархии расположены подсистемы, реализующие бизнес-процессы по тем или иным аспектам деятельности организации, включая: автоматизацию подготовки и проведения конкурсных процедур; online-подачу документов и конкурсного избрания на научные и профессорско-преподавательские должности; сопровождение деятельности диссертационных советов; учет эффективности использования учебно-научного оборудования.

Приложения и подсистемы интегрированы с ядром Системы и между собой на логическом и программном уровнях путем использования общего тезауруса, описывающего понятия предметной области. Ранее в настоящем разделе было продемонстрировано разделение понятий по отдельным приложениям. В случае, если в одной из подсистем требуется использовать понятия, определяемые в другой подсистеме (например, подсистема расчета персонального рейтинга опирается на определения результатов деятельности), классы, которые описывают эти понятия, импортируются в данную подсистему из приложений, соответствующих таким понятиям. В логической структуре БД такие связи, как правило, реализуются отношениями "многие к одному" (например, несколько статей могут быть опубликованы в одном журнале) и отношениями "многие ко многим" (например, авторство статьи описано отношением "многие ко многим", которое определяет аффилиацию работника — автора с результатом научной деятельности — статьей). На уровне основного кода Системы механизмы объектно-реляционного отображения транслируют такие отношения в связи, доступные разработчику непосредственно из кода. Приведенный выше пример аффилиации автора со статьей порождает в коде такие связи, как "публикации работника" (а также "статьи работника в журналах", "статьи работника в сборниках" и "тезисы докладов автора", представляющие собой подмножества публикаций работников), "авторы статьи", "записи об авторстве статьи" (дополнительно хранят информацию о порядке упоминания авторов в библиографической ссылке и об именах авторов в том виде, в котором они приведены в библиографической ссылке). Механизмы интеграции Системы с внешними источниками данных (такими как библиографические базы данных Web of Science и Scopus) представлены в модели данных в виде отображения между внутренними идентификаторами объектов в Системе и внешними идентификаторами в источниках данных, которые соответствуют тому или иному объекту.

Одной из естественных моделей представления объектов в базе данных Системы является графовая модель. В этой модели различные объекты представляются вершинами графа, а связи между объектами различных типов — ребрами графа с указанием типа. Например, авторы статьи и сама статья связаны отношением "авторство", а статья связана с журналом или сборником отношением "опубликована в". Часто одни и те же два класса могут иметь связи разных типов, например, работник может быть докладчиком на конференции или членом программного комитета этой конференции. Необходимость использовать графовую модель возникает в различных приложениях и модулях Системы, в частности, при вводе данных, при верификации результатов, при расчете персонального рейтинга и при составлении различных отчетов. Другим примером графа, возникающего в Системе, является граф ключевых слов и разделов науки, с помощью которого можно определить семантическую близость двух понятий, классифицировать их по отношению к тому или иному разделу науки, а также осуществить поиск по ключевым словам.

Для удобства работы с графовыми моделями в рамках Системы ведется разработка редактора онтологий. Данный редактор позволяет пользователям совместно работать над пополнением графов большого размера, он основан на стандартах Консорциума Всемирной паутины, что дает возможность соединить графы, редактируемые в редакторе, с графами, уже разработанными вне ИАС "ИСТИНА". Редактор онтологий позволяет каждому пользователю создать несколько собственных рабочих пространств ("страниц онтологии"), создавать и редактировать вершины графа внутри этих страниц и ребра, их соединяющие, и "склеивать" эти страницы между собой. Редактор имеет механизм версионного контроля, что позволяет просматривать изменения, которые вносятся пользователями, и узнавать, кто из них создал тот или иной участок графа.

Отметим, что большинство функций этого модуля пока находятся в стадии исследовательской разработки. Однако эти исследования рассматриваются разработчиками Системы как одно из наиболее перспективных направлений развития систем такого рода.

Технологические аспекты разработки

Ранее в числе основных принципов создания и развития Системы было упомянуто соблюдение основных положений и рекомендаций к инженерии программ на всех этапах ее жизненного цикла. В настоящем разделе описаны отдельные технологические аспекты процессов разработки Системы, которые играют важную роль в реализации этого принципа и в то же время в той или иной степени отражают специфику крупных информационно-аналитических систем.

Управление исходным кодом. В процессах разработки и модернизации ИАС "ИСТИНА" используется система GitLab для управления репозиторием исходного кода. Эта система предоставляет веб-интерфейс к набору репозиториев исходного кода ИАС "ИСТИНА". Из числа функциональных возможностей системы GitLab в рамках технологических процессов разработки ИАС "ИСТИНА" следует выделить встроенные средства рецензирования кода. Аналогично популярному проприетарному аналогу — системе GitHub, в GitLab выделено понятие "запросов на слияние" — наборов изменений исходного кода с текстовым обоснованием таких изменений. Содержание каждого такого запроса разрабатывается в отдельной ветке системы контроля версий, независимо от других модификаций кода Системы. После завершения работы над новыми функциональными возможностями или исправлением недостатков Системы автор запроса на слияние передает его на рецензию кому-либо из других членов коллектива, которые не вносили изменений в эту ветку кода, но обладают знаниями об особенностях модифицируемых фрагментов кода. Напрямую (без запросов на слияние и рецензирование кода) вносить изменения в основную ветку запрещено техническими средствами. Такая схема работы имеет следующие достоинства: все члены коллектива получают возможность ознакомиться с новыми изменениями кода Системы; повышается вероятность выявления опечаток и тривиальных ошибок в коде.

В рамках совершенствования технологических процессов разработки в 2017 г. предполагается внедрить практики непрерывной интеграции. Для каждого запроса на слияние (т. е. каждого набора изменений, реализующих те или иные новые функциональные возможности или исправляющих недостатки Системы) средствами системы GitLab будет выполняться набор автоматических тестов, включающий первичное "дымовое" тестирование (*smoke testing*). Такое тестирование заключается в запуске основных компонентов Системы и загрузке главной страницы интерфейса пользователя. Оно предназначено для выявления ошибок в конфигурации Системы, например, для случая, когда разработчик использовал в коде новую программную библиотеку, но при этом не включил ее в список зависимостей. Набор автоматических тестов должен также включать статический анализ кода с использованием средства pyflakes, которое позволит выявить потенциальные опечатки, синтаксические ошибки, ошибки в именовании переменных и в импорте библиотечных функций, также он должен включать в себя юнит-тесты и интеграционные тесты.

Перспективные направления для улучшения контроля в этой области включают использование гибридного полигона в сценариях автоматического тестирования, а также проверку на уровень покрытия основного кода Системы тестами.

Обработка обращений пользователей и сообщений об ошибках. Информационно-аналитическая система "ИСТИНА" используется крупнейшей научно-образовательной организацией России — Московским государственным университетом им. М. В. Ломоносова. В настоящее время в Системе работает 15 организаций и более 25 000 пользователей. Учитывая ограниченный размер коллектива разработчиков, традиционные способы взаимодействия с пользователями, такие как личное общение и связь с членами коллектива по электронной почте или по телефону, показали себя неэффективными. В связи с этим было принято решение о развертывании системы обработки обращений пользователей на основе системы Redmine. На каждом экране пользовательского интерфейса ИАС "ИСТИНА" в нижней части находится ссылка "Создать обращение в службу поддержки". При нажатии на эту ссылку открывается диалоговое окно, в котором пользователь может кратко описать суть вопроса, который у него возник, и указать предметную категорию обращения. После отправки обращения пользователем сервер приложений ИАС "ИСТИНА" с использованием интерфейса разработчика системы Remine создает новое обращение в этой системе, в которое, кроме пользовательского текста, включается также дополнительная информация. В состав такой дополнительной информации входит имя и место работы пользователя, адрес страницы, на которой было создано обращение. Настройки и расширения системы Redmine позволяют отправить пользователю email-сообщение с уведомлением о том, что обращение принято. С помощью ответа на это сообщение пользователь может предоставить дополнительную информацию по обращению и прикрепить к нему файлы с дополнительной информацией.

Система Redmine используется также для планирования новых задач по модернизации и сопровождению Системы. Ведутся работы по включению в систему обработки обращений лиц, ответственных за сопровождение информации в ИАС "ИСТИНА" в организациях, использующих Систему. Таким образом, ответственные за сопровождение информации смогут взять на себя первичную фильтрацию обращений и отвечать на ряд обращений организационно-административного характера, а также обрабатывать обращения, которые находятся в области их компетенции.

Для обработки ошибок ИАС "ИСТИНА" использует интегрированную версию компонента непрерывного мониторинга ошибок Sentry. В случае, если в работе Системы возникла ошибка (исключение в терминах языка Python), в журнал непрерывного мониторинга записывается сообщение об ошибке с сохранением параметров HTTP-запроса, который вызвал ошибки, и информацией об исключении, включая трассировку стека и состояние переменных. Кроме того, записи в компоненте Sentry группируются по имени функции, в которой произошла ошибка, что позволяет упростить анализ. На прак-

тике компонент непрерывного мониторинга ошибок позволяет разработчикам оперативно реагировать на возникшие регрессии в коде Системы.

Комплект документации Системы. Одним из аспектов, который часто обходят вниманием при разработке крупных, быстро изменяющихся программных систем, является создание и поддержание в актуальном состоянии комплекта документации по системе. Ранее комплект документации ИАС "ИСТИНА" составлялся с использованием средств LaTeX (для отчетов о научно-исследовательской работе) и редакторов, совместимых с Microsoft Word (документы ЕСПД). Тот факт, что документы Microsoft Word хранятся в репозитории исходного кода "непрозрачно", без возможности просмотра изменений, способствовал рассинхронизации документации с фактическим состоянием Системы. В начале 2017 г. процесс подготовки документов в формате ЕСПД, которые для ИАС "ИСТИНА" включают руководство пользователя, описание программы и руководство системного программиста, был переведен на средство генерации документации Sphinx.

Средство Sphinx принимает на вход легковесный текстовый язык разметки reStructuredText и генерирует по нему комплект документации в различных форматах. Это средство используется рядом крупных программных продуктов с открытым исходным кодом, включая язык Python, платформу Django и ядро операционной системы Linux.

Для руководства пользователя основным преимуществом Sphinx на практике оказалась возможность генерации документов в формате PDF и в виде портала документации на базе статических HTML-страниц с поиском на стороне клиента с использованием кода на языке JavaScript. Руководство пользователя в форме такого портала в настоящее время опубликовано на сайте http://docs.istina.msu.ru/ и обновляется в полуавтоматическом режиме из той же ревизии, что и основной код Системы. В сочетании с организационными мерами (проверка на предмет обновления документации при рецензировании запросов на слияние) это позволяет поддерживать руководство пользователя в актуальном состоянии. Версия документации в формате PDF использовалась для подготовки комплекта документации ЕСПД, кроме того, одна из версий руководства пользователя была издана ограниченным тиражом в печатном виде. Актуальная PDF-версия доступна на сайте руководства пользователя.

При формировании руководства системного программиста используется возможность генератора Sphinx подгрузки документации непосредственно из кода Системы на языке Python. Такая документация записывается в так называемых строках документации (*docstring*) — комментариях специального вида в коде на языке Python. Формат строк документации представляет собой расширенние языка reStructuredText. Более подробно он описан в документе PEP 287, входящем в набор официальных расширений спецификации языка Python. Средство Sphinx предоставляет дополнительные расширения языка разметки, включая директивы automodule, autoclass, autofunction и прочие, которые позволяют интегрировать строки документации отдельных модулей, классов и функций с общим описанием структуры кода модулей Системы в рамках руководства системного программиста.

Опыт перехода на новую версию платформы. Стандартной практикой для крупных современных программных систем является использование платформ (каркасов, *frameworks*), которые поддерживаются внешними разработчиками. Как правило, такие платформы представляют собой проекты с открытым исходным кодом, которые поддерживаются силами одной или нескольких крупных компаний. Примерами таких платформ с открытым исходным кодом для веб-ориентированных систем, написанных на различных языках программирования, служат ASP.NET Core MVC (язык программирования C#), Spring Framework (Java), Rails (Ruby) и, в случае ИАС "ИСТИНА", Django (Python).

Интерфейс разработчика (API) платформы в той или иной степени используется в значительной части исходного кода системы. Внешний характер развития и поддержки платформ при этом проявляется в том, что при выходе новой версии платформы поведение некоторых функций платформы меняется. Поддержка старой версии, которая включает в себя исправление ошибок, в том числе и потенциально влияющих на безопасность системы, как правило, завершается в достаточно короткие сроки. В таких случаях требуется внесение изменений в код системы для перехода на новую версию платформы. Для языков программирования, не имеющих статической типизации, переход осложняется отсутствием обратной связи от компилятора, которая позволяет оперативно обнаружить фрагменты кода, которые требуют изменений.

Весной 2017 г. коллектив разработчиков ИАС "ИСТИНА" выполнил переход с неподдерживаемой в настоящее время версии платформы Django 1.4 на версию Django 1.8, которая находится в состоянии долгосрочной поддержки (LTS). Учитывая непрерывный характер и высокие темпы внесения изменений в код ИАС "ИСТИНА", переход необходимо было осуществлять в основной ветке (иначе версии кода Системы могли бы разойтись) с сохранением совместимости кода с используемой на тот момент в режиме эксплуатации (*production*) версией Django 1.4. Переход был осуществлен путем поэтапного добавления поддержки для последующих версий платформы (1.5, 1.6, 1.7 и 1.8). После перевода эксплуатационного сервера на Django 1.8 поддержка версий 1.7 и более ранних была исключена.

Когда значительная часть изменений для поддержки Django 1.8 была внесена, был объявлен трехнедельный период бета-тестирования, в рамках которого большинство разработчиков перешли на версию Django 1.8 локально, кроме того, на новую версию был переведен тестовый сервер. На основе журнала HTTP-запросов Системы наиболее часто используемые запросы с эксплуатационного сервера были воспроизведены на тестовом сервере. Этап бета-тестирования позволил выявить и исправить значительную часть ошибок. На заключительном этапе на эксплуатационный сервер параллельно с основным окружением Django 1.4 было добавлено окружение для новой версии и эксплуатационный сервер был переключен на

Django 1.8. На этом этапе был выявлен еще ряд ошибок, которые были исправлены в течение 1—2 дней.

При разработке платформы Django используется двухэтапный период вывода устаревших компонентов интерфейса программиста из эксплуатации. Если компонент отмечен как устаревший в некоторой версии, его поддержка прекращается не ранее чем через две версии платформы. Например, компонент может быть отмечен устаревшим в версии 1.5, тогда он будет удален в версии 1.7.

Значительная часть ошибок на заключительном этапе перехода была выявлена с помощью упомянутой ранее системы Sentry. Данные трассировки стека в этих случаях, как правило, позволяли исправить эти ошибки оперативно и с минимальными объемами изменений. Некоторые ошибки определить было сложнее. В частности, в силу измененной обработки порядка отображения полей ввода для форм, на платформе Django версии 1.8 в редких случаях появлялись нежелательные поля. Ошибки такого рода могли быть обнаружены только путем сравнения результатов обработки Системой одного и того же запроса на платформе Django версии 1.4 и на платформе Django версии 1.8.

Обновление используемой платформы и внешних зависимостей не является однократным мероприятием. В частности, используемая в настоящее время версия платформы будет поддерживаться только до апреля 2018 г. Нужно отметить, что переход на более новую версию платформы следует рассматривать как сокращение технического долга Системы. Ожидается, что последующие обновления будут проходить с меньшими трудозатратами. В ближайшей перспективе предполагается внести в код изменения, устраняющие все случаи использования функций, объявленных устаревшими в платформе Django версий 1.7 и 1.8, а также обновить внешние зависимости до их последних версий. После этого до апреля 2018 г. будет выполнен постепенный переход на платформу Django версии 1.11. Следует отметить, что, начиная с версии 2.0, которая следует за версией 1.11, разработчики платформы Django перешли на новый цикл разработки, который упрощает процесс перехода для проектов, которые ориентируются на версии платформы с долгосрочной поддержкой.

Список литературы

1. **Налимов В. В., Мульченко З. М.** Наукометрия. Изучение развития науки как информационного процесса. Физико-математическая библиотека инженера. М.: Наука, 1969. 192 с.
2. **Васенин В. А.** Модернизация экономики и новые аспекты инженерии программ // Программная инженерия. 2012. № 2. С. 2—17.
3. **Михайленко И. В.** Информационно-аналитическая система "Карта Российской науки" как инструмент научного мониторинга // Динамика систем, механизмов и машин. 2014. № 4. С. 75—78.
4. **Гончаров М. В., Михайленко И. В.** Наукометрические показатели, используемые в ИАС "Карта Российской науки". Методика расчета // Научные и технические библиотеки. 2016. № 12. С. 37—43.
5. **Васенин В. А., Голомазов Д. Д., Ганкин Г. М.** Архитектура, методы и средства базовой составляющей системы управления научной информацией "ИСТИНА — Наука МГУ" // Программная инженерия. 2014. № 9. С. 3—12.
6. **Васенин В. А., Афонин С. А., Козицын А. С., Голомазов Д. Д.** Система "ИСТИНА" для подготовки принятия решений на основе анализа наукометрической информации // Научный сервис в сети Интернет: Труды XVII Всероссийской научной конференции. ИПМ им. М. В. Келдыша, Москва, 2015. С. 51—62.
7. **Васенин В. А., Иткес А. А., Бухонов В. Ю., Галатенко А. В.** Модели логического разграничения доступа в многопользовательских системах управления наукометрическим контентом // Программная инженерия. 2016. Т. 7, № 12. С. 547—558.
8. **Васенин В. А., Зензинов А. А., Лунев К. В.** Использование наукометрических информационно-аналитических систем для автоматизации проведения конкурсных процедур на примере информационно-аналитической системы "ИСТИНА" // Программная инженерия. 2016. Т. 7, № 10. С. 472—480.
9. **Cobo M. J., López-Herrera A. G., Herrera-Viedma E., Herrera F.** Science mapping software tools: Review, analysis, and cooperative study among tools //Journal of the Association for Information Science and Technology. 2011. Vol. 62, N. 7. P. 1382—1402.
10. **Grivel L., Polanco X., Kaplan A.** A computer system for big scientometrics at the age of the World Wide Web // Scientometrics. 1997. Vol. 40. N. 3. P. 493—506.
11. **Johansson Å., Ottosson M. O.** A national current research information system for Sweden // 11th International Conference on Current Research Information Systems — CRIS 2012: Prague, Czech Republic, June 6—9, 2012. Agentura Action M, 2012. P. 67—71.
12. **Clements A., McCutcheon V.** Research data meets research information management: Two case studies using (a) Pure CERIF-CRIS and (b) EPrints repository platform with CERIF extensions // Procedia Computer Science. 2014. Vol. 33. P. 199—206.
13. **Jeffery K. G., Asserson A.** Supporting the Research Process with a CRIS // Enabling Interaction and Quality: Beyond the Hanseatic League. 2006. P. 121—130.

Architectural and Technological Aspects of the Development and Maintenance of Large Information Analysis Systems in the Area of Science and Education

V. A. Vasenin, vasenin@msu.ru, **M. A. Zanchurin,** maxim.zanchurin@gmail.com,
A. S. Kozitsyn, alexanderkz@mail.ru, **M. A. Krivchikov,** maxim.krivchikov@gmail.com,
D. A. Shachnev, mitya57@mitya57.me, Lomonosov Moscow State University, Moscow, 119991, Russian Federation

Corresponding author:

Vasenin Valery A., Dr. Sc., Professor, Head of the Chair, Lomonosov Moscow State University, Moscow, 119991, Russian Federation
E-mail: vasenin@msu.ru

Received on July 22, 2017
Accepted on August 02, 2017

Mechanisms for managing the scientific, technical and educational activity are a subject of a special attention of the government. With the emergence of packet communications technologies and tools, and the Internet meta-network and technologies based on it, at the beginning of 21st century there were all prerequisites ready for creating systems available to collect, store and analyze large amounts of scientometrical data in an immediate manner. The present article considers the architectural and technological aspects of the development and maintenance of such large information analysis systems in the area of science and education, based on the example of the "ISTINA" system (referred to as System throughout the text).

The main requirements to the System architecture include:

1) modularity and scalability, on the macro-level of system description as well as on the underlying levels;

2) hierarchical organziation and orientation on processes;

3) completeness of the reflection of all processes currently in demand and for the foreseeable future that are automatized within the System;

4) taking processes of interaction with external databases storing various kinds of auxiliary data into account.

The technological principles include:

• compliance with regulations relating to systems of national importance;

• adherence to the main provisions of software engineering at all stages of the System life cycle;

• using the mechanisms of collaboration and integration with the external and internal data bases taking the confidentiality requirements into account.

The System architecture reflects the ontological model of the scientometrics using its representation as the subject area on which the System is oriented. The integrating mechanisms of the System are implemented using a relational database, the structure of which reflects the main aspects of the ontology, and a set of applications using that database.

One of the natural models for representing the objects in the System database is the graph model. A visual ontology editor for working with graph is being developed at the moment. Investigations on using the ontologies in the system architecture are considered as one of the most perspective development directions of such systems by the developers.

With regards to technological aspects this article describes approaches to source code management and continuous integration; processing the users' requests and bug reports; tools for preparing the documentation packages for the System; experience with upgrading to the new version of Django framework.

Keywords: *information analysis systems, science and education, scientometrics, software engineering, distributed systems, software architecture*

For citation:

Vasenin V. A., Zanchurin M. A., Kozitsyn A. S., Krivchikov M. A., Shachnev D. A. Architectural and Technological Aspects of the Development and Maintenance of Large Information Analysis Systems in the Area of Science and Education, *Programmnaya Ingeneria*, 2017, vol. 8, no. 10, pp. 448—455.

DOI: 10.17587/prin.8.448-455

References

1. **Nalimov V. V., Mulchenko Z. M.** *Naukometrija. Izuchenie razvitija nauki kak informacionnogo processa. Fiziko-matematicheskaja biblioteka inzhenera* (Studying the science development as an information process. Physics and mathematics library of the engineer), Moscow, Nauka, 1969, 192 p. (in Russian).

2. **Vasenin V. A.** Modernizacija jekonomiki i novye aspekty inzhenerii programm (Economics modernization and new aspects of software engineering), *Programmnaya Ingeneria*, 2012, no. 2, pp. 2—17 (in Russian).

3. **Mikhailenko I. V.** Informacionno-analiticheskaja sistema "Karta Rossijskoj nauki" kak instrument nauchnogo monitoringa (Information analysis system "Russian science map"), *Dinamika sistem, mehanizmov i mashin*, 2014, no. 4, pp. 75—78 (in Russian).

4. **Goncharov M. V., Mikhailenko I. V.** Naukometricheskie pokazateli, ispol'zuemye v IAS "Karta Rossijskoj nauki". Metodika raschjota (Scientometrical parameters used in IAS "Russian science map". Methodics of calculation), *Nauchnye i tehnicheskie biblioteki,* 2016, no. 12, pp. 37—43 (in Russian).

5. **Vasenin V. A., Golomazov D. D., Gankin G. M.** Arhitektura, metody i sredstva bazovoj sostavljajushhej sistemy upravlenija nauchnoj informaciej "ISTINA — Nauka MGU" (Architecture, methods and tools of the base part of the "ISTINA — MSU Science" science information management system), *Programmnaya Ingeneria,* 2014, no. 9, pp. 3—12 (in Russian).

6. **Vasenin V. A., Afonin S. A., Kozitsyn A. S., Golomazov D. D.** Sistema "ISTINA" dlja podgotovki prinjatija reshenij na osnove analiza naukometricheskoj informacii (The "ISTINA" decision preparing system based on scientometrical data analysis), *Nauchnyj servis v seti Internet: Trudy XVII Vserossijskoj nauchnoj konferencii,* IPM im. M. V. Keldysha, Moscow, 2015, pp. 51—62 (in Russian).

7. **Vasenin V. A., Itkes A. A., Bukhonov V. Yu., Galatenko A. V.** Modeli logicheskogo razgranichenija dostupa v mnogopol'zovatel'skih sistemah upravlenija naukometricheskim kontentom (Models of logical access restriction in multi-user scientometrical content management systems), *Programmnaya Ingeneria*, 2016, vol. 7, no. 12, pp. 547—558 (in Russian).

8. **Vasenin V. A., Zenzinov A. A., Lunev K. V.** Ispol'zovanie naukometricheskih informacionno-analiticheskih sistem dlja avtomatizacii provedenija konkursnyh procedur na primere informacionno-analiticheskoj sistemy "ISTINA" (Using scientometrical information analysis systems for automating the contest procedures on example of "ISTINA" information analysis system), *Programmnaya Ingeneria*, 2016, vol. 7, no. 10, pp. 472—480 (in Russian).

9. **Cobo M. J., Lopez-Herrera A. G., Herrera-Viedma E., Herrera F.** Science mapping software tools: Review, analysis, and cooperative study among tools, *Journal of the Association for Information Science and Technology*, 2011, vol. 62, no. 7, pp. 1382—1402.

10. **Grivel L., Polanco X., Kaplan A.** A computer system for big scientometrics at the age of the World Wide Web, *Scientometrics,* 1997, vol. 40, no. 3, pp. 493—506.

11. **Johansson A., Ottosson M. O.** A national current research information system for Sweden, *11th International Conference on Current Research Information Systems*, CRIS 2012: Prague, Czech Republic, June 6—9, 2012, Agentura Action M, 2012, pp. 67—71.

12. **Clements A., McCutcheon V.** Research data meets research information management: Two case studies using (a) Pure CERIF-CRIS and (b) EPrints repository platform with CERIF extensions, Procedia Computer Science, 2014, vol. 33, pp. 199—206.

13. **Jeffery K. G., Asserson A.** Supporting the Research Process with a CRIS, *Enabling Interaction and Quality: Beyond the Hanseatic League*, 2006, pp. 121—130.