1 СИСТЕМЫ С ДЕЦЕНТРАЛИЗОВАННЫМ ХРАНЕНИЕМ ДАННЫХ

Традиционно большие наукометрические системы имеют централизованную архитектуру. Примерами могут служить базы наукометрических данных Web of Science, Scopus, а также РИНЦ и ИАС «ИСТИНА» [1]. Однако масштабирование системы с централизованной архитектурой для сотен тысяч и миллионов пользователей осложняется как техническими, так и организационными факторами. В настоящем разделе собран перечень подобных сложностей. Представлены предложения по их решению путём децентрализации системы — а именно, путём отказа от требования единой централизованной базы данных и единого неделимого экземпляра системы.

1.1 Задачи наукометрических систем

Далее представлены задачи, которые решаются системами поддержки принятия управленческих решений на основе наукометрии, к которым относится ИАС «ИСТИНА». Эти задачи отражают специфику, приводящую к усложнению процессов масштабирования и сопровождения системы.

Общие задачи включают в себя:

- сбор и верификацию данных о результатах научно-исследовательской, педагогической и инновационной деятельности (далее — результаты);
- сопровождение отдельных бизнес-процессов, включая деятельность диссертационных советов, проведение стимулирующих конкурсов.

В организациях высшего образования и науки с помощью наукометрии решаются задачи:

- оценки эффективности работников;
- персонального и коллективного стимулирования;
- сбора отчётных материалов и формирования отчётов для предоставления в федеральные органы исполнительной власти.

На уровне органов государственного управления, включая главных распорядителей бюджетных средств, а также Российскую Академию Наук, как структуру, обладающую полномочиями научно-методического руководства исследованиями в стране, к задачам, для которых целесообразно задействовать системы поддержки принятия управленческих решений на основе наукометрии относятся:

- экспертиза результатов научных исследований и разработок;
- оценка эффективности организаций науки и высшего образования (далее — организаций).

1.2 Сложности при масштабировании наукометрических систем

Сопровождение системы требует финансовых ресурсов, которые для систем поддержки принятия решений, как правило, получают путём прямой оплаты. Но в случае, когда

организация оплачивает использование системы, подразумевается, что система будет хотя бы в ограниченном объёме дорабатываться под потребности клиентов, включая следующие.

- Необходимость учитывать индивидуальные бизнес-процессы. Ряд крупных организаций науки и высшего образования имеют свои Положения о присуждении учёных степеней, о диссертационных советах.
- Необходимость учитывать специфические индикаторы, актуальные не для всех организаций: клинические испытания в медицине, подготовка материалов в ответ на запросы государственных органов власти, организаций или граждан.
- Наличие ведомственной отчётности, специфичной, например, для Минздрава.
- Необходимость интеграции с внутренними информационными системами и базами данных — например, кадровые базы данных для верификации сотрудников и базы данных педагогической нагрузки для организаций высшего образования;
- Для крупных организаций актуален вопрос поддержки брендирования системы, включая персональный домен, свой логотип и своё название.

Опыт пилотной эксплуатации ИАС «ИСТИНА» в институтах Российской Академии Наук показывает, что учёт специфики отдельных ведомств возможен, но, предположительно, на масштабах всех организаций науки и высшего образования страны потребует существенного усложнения системы. К числу ограничивающих факторов можно отнести следующие.

- наличие в коде специфических доработок для десятков министерств и ведомств;
- наличие в коде модулей интеграции с сотнями внутренних информационных систем и баз данных;
- необходимость отдельных организаций включать/отключать предметно-ориентированные подсистемы;
- необходимость для отдельных организаций включать/отключать специфические индикаторы;
- вовлечение администраторов с полным доступом к системе в решение большого объёма задач по сопровождению.

С технической точки зрения эксплуатация крупной централизованной системы предъявляет трудновыполнимые требования к производительности. Централизованная база данных (БД) подразумевает транзакционный доступ пользователей, одновременно работающих с системой. Как показали предыдущие исследования, централизованная БД в этой связи затрудняет географическое распределение, что снижает отказоустойчивость системы, а для потенциальных интерактивных приложений может снизить качество работы с системой в организациях, которые расположены в Сибири и на Дальнем Востоке. Поддержка распределённых реплик БД требует реализации сложных алгоритмов синхронизации с труднопрогнозруемым поведением под нагрузкой. Масштабирование на миллионы пользователей с периодами пиковой нагрузки подразумевает использование сложных алгоритмов кэширования со сложными алгоритмами инвалидации кэша. С ростом количества пользователей производительность наблюдается нелинейное падение производительности в случаях, когда БД перестаёт помещаться в память того или иного уровня.

Ряд сложностей связан с вопросами организационного характера. А именно:

- необходимость учитывать различия в подписке на доступ к международным цитатно-аналитическим базам данных (WoS, Scopus, отраслевые БД);
- необходимость учитывать различия в подписке на доступ к российским сервисам (eLIBRARY, Антиплагиат);

– наличие более 100 000 пользователей системы означает более жёсткие критерии к защите персональных данных;
– необходимость локального хранения конфиденциальных данных организации.

Кроме того, естественным ограничивающим фактором масштабирования централизованной системы может стать модель ввода данных «снизу вверх». Руководство каждой организации на практике настаивает на том, чтобы профессиональный профиль работника организации в системе содержал только результаты, релевантные для этой организации. Это приводит к разного рода негативным сценариям. Например, у некоторых пользователей системы ответственные за сопровождение данных от организации систематически удаляют записи о результатах, полученных пользователями при работе в других организациях ранее. Профиль одного пользователя редактируют десятки ответственных: по каждому из подразделений, где он работает, работал или подаёт заявку на конкурс; по каждому из диссертационных советов, в который он входит; по каждой НИР, в которой он участвует. Это приводит к так называемым «войнам правок», которые были исследованы ранее на примере открытой энциклопедии «Википедия». Уже при количестве пользователей системы порядка 100 тыс. в число популярных запросов на развитие системы входит возможность заблокировать запись о результате от дальнейших изменений.

Перечисленные в настоящем разделе сложности актуальны для системы «ИСТИНА». Они будут более актуальны при более широком распространении системы и, вероятно, усугубятся при попытке распространения системы за пределы России. В этой связи, возможность масштабирования системы в её централизованном виде вызывает опасения в связи с аккумулированием системой сложности.

1.3 Децентрализованные социальные сети

В настоящее время консорциум W3C и интернет-сообщество разрабатывают и поддерживают ряд спецификаций моделей данных и протоколов, которые предназначены для эффективного взаимодействия децентрализованных систем, построенных по принципу социальной сети на основе web-технологий. Такие спецификации позволяют объединить множество существующих в настоящее время разрозненных источников наукометрических данных в единую глобальную децентрализованную систему.

К числу таких спецификаций относятся следующие.

– модель данных microformats2 [2] определяет расширяемые семантические аннотации для данных в формате HTML с помощью идентификаторов CSS-классов;
– модель данных h-feeds в составе microformats2 определяет структурированные наборы изменений в виде HTML-страницы с краткой информацией об изменённых объектах;
– протокол уведомления об упоминаниях Webmention [3] используется для простого автоматизированного распространения сведений об упоминаниях одних ресурсов на других и об изменениях в ресурсах, которые ссылаются на данный (при этом могут использоваться и сторонние узлы обработки упоминаний Webmention);
– в совокупности модель microformats2 и протокол Webmention позволяют учитывать контекст упоминания;
– протокол WebSub [4] допукает подписку на уведомления об изменениях в данном ресурсе.

Эти спецификации можно считать развитием предложенных ранее и уже достаточно хорошо изученых технологий linked data и концепции semantic web, а также схемы описания RDF. Однако существующие реализации технологии linked data требуют использования отдельных технических средств или дублирования данных по отношению к отображаемым в HTML-интерфейсе информационной системы, что повышает порог входа в децентрализованную систему. Преимуществом модели microformats2 в этом отношении является минимальный объём дополнительных действий, которые требуется выполить. В некоторых случаях microformats2-аннотации можно использовать непосредственно для оформления HTML-страницы. В контексте известных результатов следует упомянуть также проект SOLID Тима Бернерса-Ли [5].

1.4 Модель децентрализованной наукометрической системы

Описанные выше протоколы могут быть использованы для создания децентрализованной наукометрической системы.

В рамках такой системы каждый субъект — отдельный учёный, научная организация или её структурное подразделение — имеет свой экземпляр данных, который связан ссылками с другими экземплярами, в том числе — в наукометрических базах данных и на сайтах журналов. Данными этого экземпляра субъект полностью и единолично распоряжается, и за корректность этих данных субъект несёт личную ответственность.

Наукометрические данные доступны по протоколу HTTP и хранятся в едином открытом формате на основе HTML — например, с семантическими аннотациями согласно спецификации microformats2. Экземпляры системы связаны между собой гиперссылками, которые также имеют семантические аннотации, отражающие связь между дублирующимися данными в разных экземплярах. Верификация корректности данных производится децентрализованно заключается в сравнении данных, представленных в источниках с разным уровнем доверия. Например, административная верификация данных подразделением эквивалентна наличию этих данных в экземпляре системы подразделения.

Наличие гиперссылок позволяет экземплярам системы автоматически уведомлять друг друга об обновлениях в данных по протоколу webmentions. Каждый экземпляр может использовать отдельные средства защиты, в зависимости от требований к защите информации, которые предъявляются к тому или иному экземпляру системы. Отдельные экземпляры системы могут представлять собой защищённые зеркала с дополнительными данными, такими, как персональные данные или данные, представляющие коммерческую тайну (финансовые данные). Такие защищённые узлы могут быть подписаны по протоколу WebSub на весь поток изменений в другом (открытом) узле. Таким образом, например, защищённый узел, содержащий персональные данные, может автоматически получать обновления от открытого узла, на котором публикуются только общедоступные наукометрические данные.

Отдельные глобальные сервисы, которым нужен полный набор данных для анализа, реализуются в форме специализированных поисковых роботов. Такие поисковые роботы в постоянном режиме собирают обновления исходных наукометрических данных на всех известных им узлах, и сохраняют их в виде, наиболее эффективном для задачи, которая решается сервисом, без необходимости следовать единой жёсткой модели данных.

Схема взаимодействия по предложенной модели представлена на рис. 1.1.

Децентрализованная архитектура сводит задачи масштабирования и защиты системы на миллионы пользователей к задаче обеспечения эффективной и безопасной работы узла в рамках одной крупной организации или её структурного подразделения.

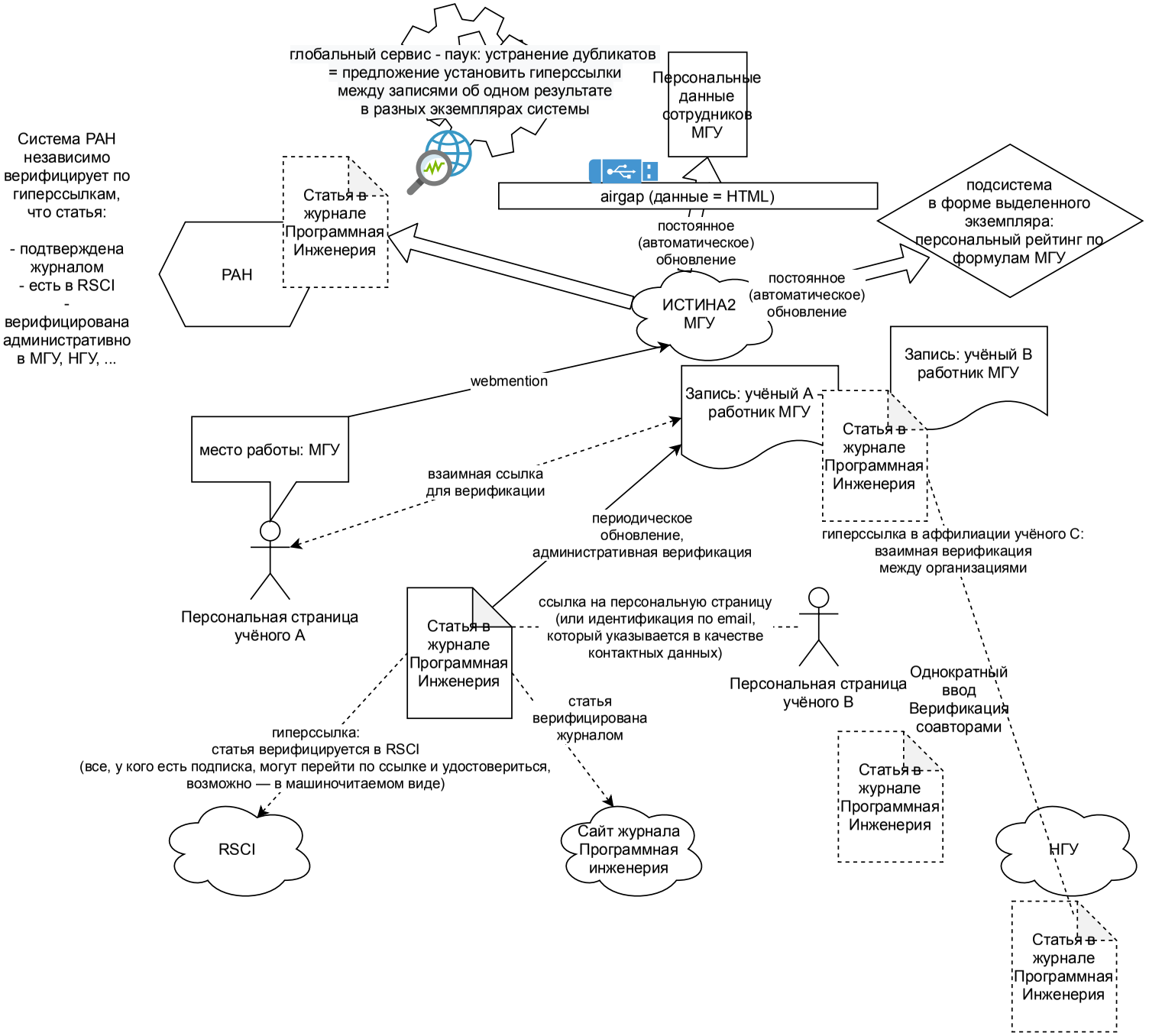

Рисунок 1.1 — Модель децентрализованной наукометрической системы

В терминах открытых протоколов единообразно могут быть представлены задачи распределённого ввода и автоматического импорта из внешних баз данных, а также задачи автоматизированной и административной верификации данных и сбора отчётной информации на всех уровнях административной иерархии, от отдельных кафедр и лабораторий до федеральных органов исполнительной власти и государственных информационных систем. Использование связанных данных позволит сопровождать агрегированные отчётные данные ссылками на первичную информацию для анализа и независимой верификации. Отдельные узлы системы могут быть легко адаптированы под специфические потребности отдельных организаций. С помощью прокси-сервисов в глобальную систему может быть включён широкий класс существующих информационных систем, от международных баз наукометрических данных, до внутренних информационных систем отдельных организаций.

1.5 Анализ децентрализованной модели

Проанализируем возможные преимущества и недостатки децентрализованной модели. С точки зрения производительности децентрализованная модель допускает полностью статические экземпляры системы, данные в которых хранятся в виде файлов на жёстком диске и в неизменном виде передаются по запросу. С помощью широко используемых средств, такого рода статические экземпляры могут выдавать данные по запросы для такого количества пользователей, которое допускается каналом связи с учётом используемых алгоритмов сжатия. Это означает, что возможен автоматизированный обмен данными между различными экземплярами системы.

Связующие сервисы системы, которые реализуют протоколы webmentions и websub, являются тривиальными и могут распространяться в виде отдельных сервисов, не связанных непосредственно с хранилищем данных. По сравнению с существующей реализацией на примере системы «ИСТИНА», снижаются требования к производительности сервисов ввода: нет необходимости выявлять всех соавторов, или, при передаче данных между экземплярами системы, это можно делать в отложенном пакетном режиме.

С позиций безопасности децентрализованная модель позволяет параллельно поддерживать «открытый» и «защищённый» экземпляры системы с расширенным набором данных. Распространение данных в формате HTML с относительными гиперссылками делает возможным — и простым в реализации — создание экземпляров системы, которые функционируют без сетевого подключения. Т.к. HTML-данные не содержат исполнимого кода, их можно выгружать из экземпляров системы, доступных по сети, и переносить в автономный защищённый экземпляр на носителе данных.

Данные в децентрализованной системе могут быть расположены на странице с произвольными стилями и с почти произвольной вёрсткой. Тем самым расширяется область применения системы: страницы издательств и журналов, конференций и семинаров могут использовать ту же разметку и выступать в качестве дополнительных верифицирующих узлов. Существующие информационные системы с web-интерфейсом, например, Web of Science или Scopus, могут включаться в децентрализованную наукометрическую систему с помощью прокси-сервисов, дополняющих по простым правилам HTML-вывод нужными CSS-классами

В децентрализованной наукометрической системе все данные размещены локально в организации, организация распоряжается ими и может ограничивать доступ к ним. У каждого экземпляра данных есть свой ответственный за сопровождение информации, или несколько ответственных из одного структурного подразделения. Все изменения неизбежно

проходят через ответственного. Перенос данных между экземплярами системы становится штатной процедурой

К числу недостатков следует отнести следующие.

Прежде всего, в централизованной версии системы единая база данных представляет собой единый источник верифицированных данных, чего не наблюдается в децентрализованной системе. Однако данные в таком едином источнике эфемерны с учётом постоянных изменений, которые в них вносят различные субъекты.

В децентрализованных системах может встать вопрос низкой связности системы. Для разрешения этого вопроса необходимо внедрение в систему так называемых глобальных сервисов в форме специализированных поисковых роботов, которые устанавливают связи между экземплярами системы в фоновом режиме. Кроме того, необходимы повышенные требования к стабильности ссылок URL. В настоящее время при удалении записи информационные системы удаляют все сведения о ней. Таким образом, когда пользователь обращается по URL удалённого объекта, система возвращает ответ по протоколу HTTP со статусом 404 (Not Found) — не удалось найти запись — которая не позволяет различить, была ли эта запись в системе ранее. Более надёжным с точки зрения децентрализованной системы было бы более чётко следовать спецификации HTTP. А именно, при удалении записей после слияния дубликатов, URL удалённых записей должны возвращать ответ с HTTP-статусом 301 (Permanten Redirect) — постоянную переадресацию на оставшуюся после устранения дубликатов уникальную запись. При удалении ошибочных записей более корректным является ответ с HTTP-статусом 410 (Gone), который указывает, что запись с таким URL ранее была в системе, но была удалена. Наилучшей практикой будет оставлять текстовый комментарий, уточняющий причину удаления. Подобные лучшие практики и общая схема взаимодействия между экземплярами путём передачи данных с обязательным (но, возможно, автоматизированным) подтверждением включения новых записей обращают внимание на ещё один недостаток: потенциально более высокую нагрузку на ответственных за сопровождение информации.

В децентрализованных системах на основе семантических аннотаций наподобие microformats2, которые не следуют жёстко заданной схеме данных в отличие от, например, систем на основе онтологий в формате RDF, появляется риск расхождения в данных между разными экземплярами. С учётом отмеченных выше особенностей различных отраслей науки, подобные расхождения нельзя однозначно считать негативным фактором. Однако для их минимизации в числе глобальных сервисов целесообразно предусмотреть сервис-валидатор, который будет систематизировать известные вариации схемы данных между экземплярами системы и позволит поддерживать когерентность основных элементов схемы, используемых для описания базовых результатов научно-исследовательской, педагогической и инновационной деятельности, таких, как статьи в научных журналах, доклады и т.п.

Следует отметить ещё одну технологическую особенность децентрализованной системы. В системе такого рода предметно-ориентированные подсистемы (например, подсистемы сбора и генерации отчётныз материалов и подсистемы сопровождения деятельности диссертационных советов) скорее всего будут реализованы в форме отдельных сервисов — возможно, размещённых на отдельном домене и с отдельным интерфейсом пользователя. Таким образом, каждый сервис также будет являться отдельным узлом децентрализованной системы. Это позволит использовать в реализации каждого сервиса ту структуру данных, которая наиболее удобна именно для этого сервиса. Кроме того, упрощается задача разграничения доступа к сервисам: вместо громоздкой единой политики разграничения доступа, которая разграничивает доступ к десяткам подсистем, можно использовать более простые локальные политики в рамках каждой подсистемы, а доступ к самим подсистемам разгра-

ничивать с помощью ролевой модели или других средств защиты информации (включая физическое ограничение доступа, как отмечалось выше).

Недостатками такого архитектурного решения являются повышение задержки обновления данных в аналитических сервисах, так как фактически обновления (ввод и верификация данных) происходят на других узлах системы, и дублирование данных между разными подсистемами. Однако масштабы технической задержки обновления данных, без учёта вопросов административной верификации при поступлении новых данных (узлы-подсистемы, привязанные к узлам верификации, могут загружать данные автоматически), можно при работоспособных механизмах уведомлений узлов об изменениях будут порядка минут, что представляется вполне адекватным. Речь здесь идёт именно о дополнительной задержке в обновлении данных подсистемы. С учётом того, что подсистема может использовать оптимизированные структуры данных, время ответа на запрос по уже имеющимся в подсистеме данным должно сократиться, по сравнению с централизованной системой. Имеющиеся наработки по механизмам кэширования, которые описаны в настоящем отчёте, могут позволить подсистемам строить аналитические материалы непосредственно при поступлении данных и мгновенно отдавать имеющиеся результаты по запросу, в отличие от нынешней ситуации. Дублирование данных таким образом будет представлять собой их конвертацию в оптимизированный формат для каждой из подсистем — аналогично индексам в системах управления базами данных.

При значительном упрощении кода, отвечающего за хранение и обмен данными, существенная часть сложности перейдёт в глобальные сервисы, в частности, в сервисы ввода и верификации. Кроме того, ранее уже отмечалась необходимость разработки новых глобальных сервисов для валидации, установления связи между системами и выделения дубликатов. Отдельные экземпляры этих сервисов могут обслуживать сразу несколько узлов, которые предназначены для хранения этих данных.

1.6 Экспериментальная реализация

На примере системы ИАС «ИСТИНА» отработаны отдельные элементы децентрализованной модели в рамках экспериментальной реализации.

Добавление необходимых CSS-классов для Microformats2, а также дополнительного набора данных в коде ИАС «ИСТИНА» требует менее 100 строк кода для представления материалов о научных публикациях и научно-исследовательских работах. Изменения в коде затрагивают только шаблоны Django, по которым система строит HTML-страницы. Возможно, для промышленной эксплуатации понадобятся дополнительные аннотации запросов к базе данных для сохранения производительности на её текущем уровне.

Выгрузка полного набора данных из макета в настоящее время не реализована. Предположительно, достаточно просто будет реализовать надстройку над задачей кэширования описаний результатов.

В качестве основы для клиентской части, которая выполняет разбор и преобразование данных в формате Microformats2, может быть использована библиотека mf2py [6]. Эта библиотека реализует функции на языке Python для загрузки по адресу URL и для разбора описаний в объекты данных языка Python (словари и списки) менее чем за 10 строк клиентского кода (рис. 1.2).

Отрисовка разобранных описаний обратно в аннотированный HTML-файл возможна менее чем за 10 строк клиентского кода (с использованием шаблонов Jinja2). Для сниже-

```
import mf2py
import mf2util
import sys

parsed = mf2py.Parser(url=sys.argv[1]).to_dict()
print(parsed)
```

Рисунок 1.2 — Пример законченного прототипа, выполняющего загрузку по адресу URL и разбор описаний в формате Microformats2

ния объёма кода прототипа можно рассмотреть также вариант генерации аннотированного HTML-фрагмента непосредственно из данных в фомате microformats2.

1.7 Выводы

На данный момент представляется целесообразным продолжать эксперименты в направлении разработки децентрализованных наукометрических систем. В качестве наиболее актуального практического приложения следует отметить задачу переноса данных между экземплярами системы.

Развитие исследований возможно в части разработки следующих макетов:

- макет сервиса для импорта данных в формате Microformats2 в систему, аналогичную ИАС «ИСТИНА»;
- макеты отдельных глобальных сервисов, в частности, сервиса ввода, сервиса генерации отчётов и подсчёта персонального рейтинга;
- исследование вопросов аутентификации, авторизации и разграничения доступа в децентрализованной системе, включая самодостаточную реализацию модели логического разграничения доступа.

Потенциальной целью для будущих исследований можно назвать получение протоипов основных сервисов системы общим объёмом менее 25 тыс. строк кода, каждый из сервисов не более 5 тыс. строк кода. По сравнению с ИАС «ИСТИНА» это эквивалентно сокращению объёма кода системы приблизительно в 4 раза.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Интеллектуальная система тематического исследования научно-технической информации («ИСТИНА») / В. А. Садовничий, С. А. Афонин, А. В. Бахтин и др. — Издательство Московского университета Москва, 2014. — 262 с.

2 microformats2 - Microformats Wiki. http://microformats.org/wiki/microformats2.

3 Parecki Aaron. Webmention. — 2017. — January. — https://www.w3.org/TR/2017/REC-webmention-20170112/.

4 Genestoux Julien, Parecki Aaron. WebSub. — 2018. — January. — https://www.w3.org/TR/2018/REC-websub-20180123/.

5 Berners-Lee Tim, Capadisli Sarven, Verborgh Ruben et al. The Solid Ecosystem. — 2020. — November. — https://solid.github.io/specification/.

6 Kyle Mahan, Tom Morris, Kartik Prabhu, et al. Microformats/Mf2py. — 2020. — October. https://github.com/microformats/mf2py/.