

## СОВМЕСТНАЯ ПРОГРАММНАЯ ОБРАБОТКА РАЗНОРОДНЫХ КОНСТРУКТОРСКИХ ДОКУМЕНТОВ В УЧЕБНОМ ИТ-ПРОЕКТИРОВАНИИ

А. С. Гусаренко • В. В. Миронов

**Аннотация.** В статье рассматриваются вопросы обработки разнородных конструкторских документов в автоматизированной системе учебного проектирования в области информационных технологий. Обсуждаются информационные аспекты совместной работы преподавателей в роли разработчиков и консультантов и студентов в роли исполнителей. Выявляются основные категории обрабатываемых данных, такие как офисные документы, размеченный текст, письма и сообщения, базы данных. Анализируются особенности обработки данных для характерных функций учебного проектирования. Рассматриваются технологические аспекты обработки разнородных данных: анализируется разнообразие форматов документов, локальные и удаленные источники и приемники данных, формат-ориентированные средства обработки, применяемые методы доступа к файлам, архивам, веб-сервисам и СУБД. Строится обобщенная модель обработки документов с использованием формат-ориентированных объектов, а также ER-модель данных, участвующих в обработке. Делается вывод о том, что для снижения трудоемкости разработчику необходим новый уровень абстракции данных, названный виртуальными документами.

**Ключевые слова:** учебное проектирование; внешняя модель данных; интеграция гетерогенных данных; docx; xlsx; vsdx; NoSQL, SQL.

### ВВЕДЕНИЕ

В современных условиях повышается значимость цифровых средств образовательной деятельности. Внедрение систем дистанционного обучения, разработка порталов учебного назначения обуславливают возможности автоматизации разработки конструкторской документации. На ранних этапах стояла задача удовлетворения потребности студентов и преподавателей в обмене информацией через элементы учебной деятельности – лекции, тесты, задания, материалы для лабораторных работ, обратная связь с преподавателем.

Современные требования предлагают двигаться дальше – в направлении создания цифровой среды для персонализации обучения с учетом индивидуальных траекторий обучаемых. Это обуславливает необходимость совместной обработки разнородной информации – текстовой, графической, мультимедийной, что, в свою очередь, требует совершенствования математического и программного обеспечения образовательной среды.

Целью данной работы является анализ данных учебного проектирования в сфере информационных технологий (ИТ-проектирования) в плане их разнородности (гетерогенности), методов и средств их обработки, с тем чтобы построить единую модель совместной обработки, которую в дальнейшем можно было бы использовать как основу для разработки математического и программного обеспечения автоматизированной информационной системы учебного назначения.

В качестве прототипа используется портал поддержки учебного проектирования [1], дополняющий систему дистанционного обучения, расширяя возможности работы с разнородными данными, интегрируя их на базе технологий ситуационно-ориентированного подхода к обработке гетерогенных документов [2]. Такой подход дает возможность обозначить роли участников процесса, их ключевые функции [3, 4], а также источники используемых данных.

## 1. ФУНКЦИОНАЛЬНЫЕ АСПЕКТЫ ОБРАБОТКИ ИНФОРМАЦИИ

Учебное ИТ-проектирование предполагает выработку у обучаемых умений и навыков создания конструкторских документов (КД), раскрывающих принимаемые технические решения по тем или иным аспектам проекта. Оно вовлекает в этот процесс участников – преподавателей и студентов. Сам процесс проектирования сопровождается обменом данными и их обработкой. Рассмотрим подробнее существенные аспекты и особенности этого процесса.

### 1.1. Общая модель учебного проектирования

**Участники процесса.** Основными участниками учебного проектирования являются преподаватели и студенты (рис. 1). Каждый из участников исполняет свои роли и реализует свои функции в соответствии с целью проектирования, статусом и должностными обязанностями.



Рис. 1. Совместное использование данных в АИС учебного проектирования.

*Преподаватель в роли методиста* обеспечивает подготовку учебного проектирования. Он выполняет функции создания и обновления учебного контента – разрабатывает варианты заданий, методических материалов, образцов и шаблонов документов, форм отчетов. В целом эти функции выполняются на предварительных этапах учебного проектирования, до выдачи заданий исполнителям. В процессе собственно проектирования преподаватель-методист может корректировать и совершенствовать методические материалы.

*Преподаватель в роли руководителя и консультанта* обеспечивает непосредственное управление учебным проектированием: выдачу заданий и шаблонов конструкторских документов; контроль освоения учебного материала по проекту; консультирования по отдельным этапам проектирования и по проекту в целом; технический контроль и нормоконтроль разрабатываемой документации; оперативный учет хода процесса по отдельным студентам и студенческим группам; составление соответствующих отчетов.

*Студент в роли исполнителя проекта* получает задания, шаблоны и образцы КД, указания и консультации. По результатам этапов проектирования разработанные КД предоставляются на проверку. По результатам контроля КД производится их исправление и доработка. Может предусматриваться подтверждение студентом успешного освоения учебного материала.

Указанная деятельность сопровождается соответствующим документооборотом.

*Внешние и внутренние данные.* Под внешними данными (документами) здесь подразумеваются документы в электронной (как правило) или бумажной форме, циркулирующие между участниками процесса учебного проектирования. Внешние данные отличаются от внутренних, хранимых в электронном виде в машинной памяти и недоступных непосредственно участникам проектирования. Внутренние данные содержат информацию, извлекаемую из внешних документов и/или вставляемую в них в процессе обработки.

Внешние данные (см. рис. 1) характеризуются значительной разнородностью:

- *текстовые документы* – содержат преимущественно текст, которые отображаются и редактируются с помощью текстовых процессоров;
- *графические документы* – содержат схемы с условными графическими обозначениями или чертежи с изображением в масштабе формы и устройства предметов, которые отображаются и редактируются с помощью графических процессоров;
- *электронные таблицы* – взаимосвязанные, преимущественно числовые данные, ориентированные на учет и анализ, которые редактируются и отображаются с помощью табличных процессоров в сложной табличной форме, а также в виде графиков и диаграмм;
- *презентации* – сведения о результатах учебного проектирования, ориентированные на их наглядное представление, обоснование и защиту, которые отображаются и редактируются с помощью процессоров презентаций;
- *экранные формы* – фактические данные, которые участники процесса учебного проектирования вводят в базу данных информационной системы в интерактивном режиме;
- *отчеты* – фактические и сводные данные, которые участники процесса учебного проектирования выводят из базы данных информационной системы на экран или на печать;
- *письма и сообщения* – данные, обеспечивающие коммуникацию участников, которые создаются и передаются с помощью электронной почты и мессенджеров;
- *мультимедиа* – иллюстративный графический, аудио- или видеоматериал методического характера, который создается и редактируется соответствующими мультимедийными инструментами.

Таким образом, разнообразие и разнородность документов, создаваемых, редактируемых и используемых в ходе учебного проектирования, порождают соответствующее разнообразие применяемых для этого инструментов. Рассмотренное разнообразие представлено в укрупненном виде, более детальная классификация, ориентированная на характер содержащейся информации и инструменты обработки, представлена на рис. 2 (см. также [5]).

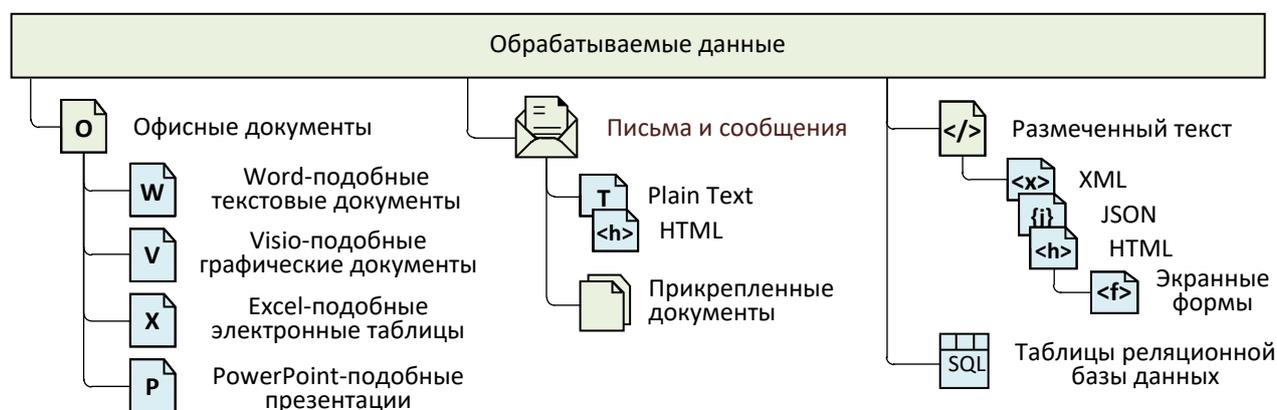


Рис. 2. Основные категории обрабатываемых данных.

*Офисные документы* – категория документов, которые традиционно поддерживаются в пакетах офисных приложений, таких как Microsoft Office [6]. Названия категорий соответствуют именам типичных процессоров, которые используются для их создания и обработки

в широко распространенном офисном пакете Microsoft Office: Word-подобные, Visio-подобные и т. д. Этим подчеркивается наличие множества процессоров, выполняющих аналогичные функции (в офисных пакетах OpenOffice, LibreOffice, Р7-Офис, МойОфис и др.).

По поводу графических документов следует отметить, что здесь рассматривается векторная графика, ориентированная на создание схем и диаграмм с помощью условных графических обозначений. Именно такие графические документы характерны для учебного проектирования в области информационных технологий. Не рассматриваются графические документы, ориентированные на чертежи, что характерно для проектирования, например, в области машиностроения (то есть не учитываются такие графические инструменты, как AutoCAD, КОМПАС-3D, NanoCAD и т. п.).

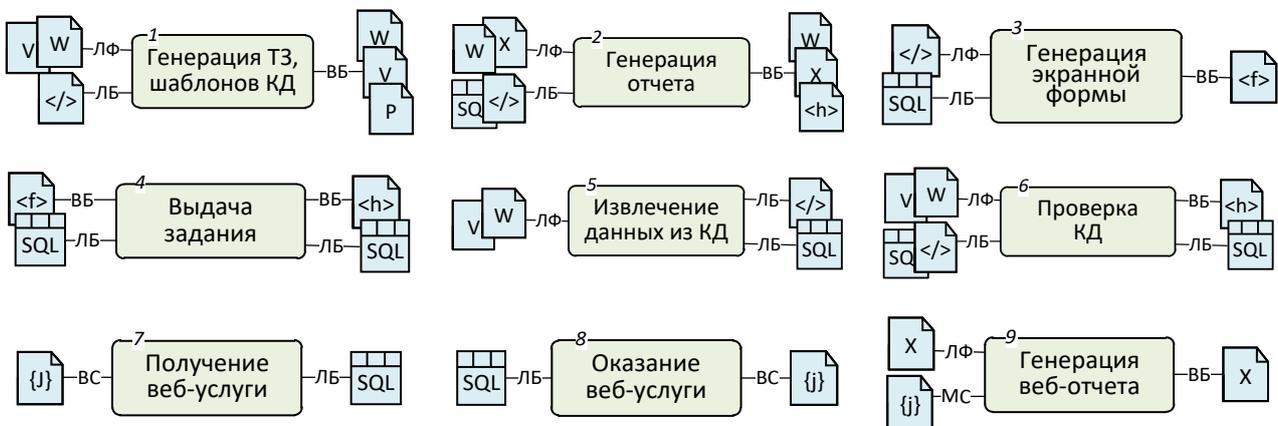
*Письма и сообщения* – категория документов, передаваемых через почтовые сервисы или мессенджеры. Сами сообщения передаются в виде текста – простого или отформатированного с помощью HTML-разметки. К сообщениям могут быть прикреплены документы (вложения) различного рода.

*Базы данных* – категория внутренних данных, хранимых в таблицах реляционных баз данных, а также в виде размеченного текста (в форматах XML, HTML, JSON) в файлах под управлением файловой системы и базах данных под управлением СУБД. Доступ к базам данных осуществляется с помощью запросов на языках, предоставляемых используемой СУБД (например, в реляционных базах данных – на структурированном языке запросов SQL).

### 1.2. Особенности совместной программной обработки документов

В информационных системах учебного ИТ-проектирования часто возникают задачи, связанные с совместной программной обработкой документов, когда на основании исходных документов и/или внутренних данных требуется программным путем сформировать (сгенерировать) результирующие документы и/или внутренние данные [6]. Как исходные, так и результирующие документы/данные могут быть разнородными в плане форматов представления информации. Кроме того, они могут быть разнородными в плане доступа к информации: локальные файлы или базы данных, размещенные на веб-сервере; веб-браузеры на стороне клиентов-пользователей; удаленные веб-сервисы; локальные микросервисы. Эти обстоятельства существенно влияют на реализацию программной обработки.

На рис. 4 представлены характерные функции совместной программной обработки, выявленные при анализе учебного ИТ-проектирования (для каждой функции слева указаны входные, а справа – выходные документы/данные, в разрывах соединительных линий указано расположение информации).



**Рис. 3.** Характерные функции программной обработки документов:

ЛФ – локальный файл; ЛБ – локальная база данных; ВБ – веб-браузер;  
 ВС – веб-сервис; МС – микросервис.

*Функция 1. Генерация ТЗ, шаблонов КД.* Это пример автоматизированного создания персонализированных заготовок конструкторских документов для выдачи студентам-исполнителям на различных этапах учебного проектирования. На основании графических и текстовых шаблонов (Visio-подобных  $V$  или Word-подобных  $W$ ) в виде локальных файлов ( $ЛФ$ ), а также варианта задания и персональных данных исполнителя (размеченный текст  $\langle / \rangle$ ) извлеченных из локальной базы ( $ЛБ$ ), на сервере генерируются результирующие заготовки конструкторских документов (Visio-подобных  $V$ , Word-подобных  $W$  или PowerPoint-подобных), которые отправляются студенту-исполнителю в веб-браузер ( $ВБ$ ) [8, 9].

*Функция 2. Генерация отчета.* Это пример формирования отчетов о ходе учебного проектирования в форме Word-подобных  $W$ , Excel-подобных  $X$  документов или HTML-страницы ( $\langle h \rangle$ ) в окне веб-браузера пользователя ( $ВБ$ ). В целом она сходна с функцией 1, но дополнительно использует  $SQL$ -данные из реляционной базы ( $ЛБ$ ).

*Функция 3. Генерация экранной формы.* Это пример формирования экранной HTML-формы ( $\langle f \rangle$ ) для отображения в окне браузера ( $ВБ$ ). Экранная форма генерируется на основе размеченных шаблонов и информации из реляционной базы. Элементы управления экранной формы позволяют пользователю вводить в информационную систему данные, документы и команды.

*Функция 4. Выдача задания.* Это пример формирования сведений о выданном студенту-исполнителю задания на учебное проектирование. Через экранную форму в браузере преподаватель-руководитель назначает определенным студентам определенные варианты заданий, что фиксируется в базе данных на веб-сервере и отражается в браузере в виде ответной HTML-страницы.

*Функция 5. Извлечение данных из КД.* В этом примере извлекается определенная семантическая информация из текстовых и графических конструкторских документов, загруженных в систему исполнителем на проверку. Извлеченные данные заносятся в локальную базу в виде размеченного текста и записей реляционной базы данных.

*Функция 6. Проверка КД.* В этом примере выполняется нормоконтроль конструкторских документов, загруженных в систему исполнителем на проверку. Для этого извлеченная семантическая информация (см. функцию 5) подвергается анализу в соответствии с определенными правилами. Результаты проверки фиксируются в базе данных ( $ЛБ$ ) и отображаются в веб-браузере ( $ВБ$ ) пользователя.

*Функция 7. Получение веб-услуги.* Это пример характерного взаимодействия с удаленным веб-сервисом. Данная конкретная функция предусматривает получение программным путем (то есть через API – Application Program Interface) комментариев пользователей, размещенных на удаленном видеохостинге для контроля освоения студентами учебного видеоконтента. Для выполнения функции формируется HTTP-запрос к веб-сервису. Входом является размеченный JSON-текст, полученный от веб-сервиса ( $ВС$ ) в ответ на HTTP-запрос, а выходом – соответствующие данные, помещенные в реляционную базу.

*Функция 8. Оказание веб-услуги.* Это пример функции, реализующей веб-сервис (в смысле сервис-ориентированного подхода). Функция запускается по HTTP-запросу, полученному от клиентского приложения, и формирует HTTP-ответ в форме размеченного текста (XML или JSON). Например, так организовано решение задачи анализа событий учебного проектирования на разных уровнях укрупнения – для отдельных студентов, для студенческих групп и т. д. Данные на уровне фактов извлекаются из реляционной базы, а сводные результаты выдаются в JSON-формате.

*Функция 9. Генерация веб-отчета.* Это пример формирования документа-отчета на основе информации, получаемой из веб-сервиса. В данном примере формируется сводная Excel-подобная таблица со статистическими сведениями по учебному проектированию. Результат формируется на основе шаблона электронной таблицы и текущих статистик, полученных от микросервиса, оказываемого веб-сервером.

Таким образом, программная обработка документов в ходе учебного ИТ-проектирования характеризуется большим функциональным разнообразием в плане:

- задач обработки данных;
- видов обрабатываемых документов и типов данных;
- расположения источников и приемников данных;
- методов получения и отправки данных.

В рамках отдельных функций требуются совместная программная обработка исходных разноформатных документов из различных источников и отправка результирующих документов различным приемникам с применением различных методов. Рассмотрим технологические аспекты этого многообразия.

## 2. ТЕХНОЛОГИЧЕСКИЕ АСПЕКТЫ СОВМЕСТНОЙ ОБРАБОТКИ ДОКУМЕНТОВ

Обеспечение совместной обработки документов сталкивается со сложностями, связанными с большим разнообразием не только на функциональном, но и на технологическом уровне. Процесс обработки должен учитывать различия форматов обрабатываемых документов, различия методов доступа к обрабатываемым документам, различия инструментов, используемых для обработки тех или иных категорий документов.

### 2.1. Форматы офисных документов

Форматы документов оказывают существенное влияние на возможности и способы их программной обработки. Редакторы офисных документов позволяют в интерактивном режиме создавать, редактировать и сохранять документы в определенных форматах. В плане программной обработки документов различают закрытые и открытые форматы (табл. 1).

*Закрытые форматы* доступны для программной обработки только с использованием специальных инструментов, поставляемых владельцами соответствующих редакторов. Следовательно, при совместной обработке разноформатных документов необходимо использовать несколько инструментов, что далеко не всегда удобно.

Так, текстовый процессор Microsoft Word поддерживает большое количество бинарных форматов, для программной обработки которых предусмотрены специальные формат-ориентированные средства. Аналогичным образом сохраняются закрытые бинарные форматы документов Microsoft Visio, Excel, PowerPoint. Бинарный формат считается устаревшим, неудобен в обработке, трудно переносим от одного текстового процессора к другому, не дает возможности создавать компактные файлы. Постепенно закрытые форматы выходят из употребления.

*Открытые форматы* допускают программную обработку общедоступными средствами.

Как правило, наряду с закрытыми форматами разработчиками предоставлялись и специальные открытые форматы на основе XML. Внутренне содержимое документов в этих форматах представляет собой смесь из XML- и бинарных данных. Такие документы должны корректно открываться и отображаться в редакторах других производителей.

В связи с критикой закрытых и специализированных XML-форматов был разработан стандарт Office Open XML (OOXML) документов [7–10], регламентирующий физическое представление документа в виде ZIP-архива, содержащего папки и XML-файлы отдельных фрагментов – данных, стилей, иллюстраций и др. Программная обработка таких форматов может выполняться единообразно с помощью стандартных средств обработки ZIP-архивов и файлов XML. Независимо от того, какой это документ (Microsoft Word, Visio, Excel, PowerPoint), все они по современным требованиям объединены стандартом Office Open XML (OOXML).

Похожим образом организованы форматы OASIS [11] – открытые форматы свободных, бесплатно распространяемых текстовых офисных процессоров. Так, формат OpenDocument Format (ODF) ориентируется на семейство редакторов OpenOffice. Он также используется в открытых бесплатных процессорах офисных документов LibreOffice Writer, LibreOffice Draw, LibreOffice Calc, LibreOffice Impress.

Таблица 1

**Основные форматы офисных документов**

Категория	Обозначение	Описание
Word-подобные текстовые документы	DOC	Закрытый бинарный формат текстовых документов Microsoft Word
	DOCX	Открытый сжатый OOXML-формат текстовых документов Microsoft Word – соответствует стандарту Office Open XML (OOXML)
	ODT	Открытый сжатый ODF-формат текстовых документов OpenOffice / LibreOffice Writer – соответствует стандарту OpenDocument Format (ODF)
Visio-подобные графические документы	VSD	Закрытый бинарный формат графики Microsoft Visio
	VDX	Открытый несжатый XML-формат графики Microsoft Visio
	VSDX	Открытый сжатый OOXML-формат графики Microsoft Visio
	ODG	Открытый сжатый ODF-формат графики OpenOffice / LibreOffice Draw
	FODG	Открытый несжатый XML-формат графики OpenOffice / LibreOffice Draw
Excel-подобные электронные таблицы	XLS	Закрытый бинарный формат табличных документов Microsoft Excel
	XLSX	Открытый сжатый OOXML-формат графических документов Microsoft Excel
	ODS	Открытый ODF-формат табличных документов OpenOffice / LibreOffice Calc
	FODS	Открытый несжатый XML-формат таблиц OpenOffice / LibreOffice Calc
PowerPoint-подобные презентации	PPT	Закрытый бинарный формат презентаций Microsoft PowerPoint
	PPTX	Открытый сжатый OOXML-формат презентаций Microsoft PowerPoint
	ODP	Открытый сжатый ODF-формат презентаций OpenOffice / LibreOffice Impress
	FODP	Открытый несжатый XML-формат презентаций OpenOffice / LibreOffice Impress

*Размеченный текст.* Важную категорию, не отраженную в табл. 1, составляют документы без стилового оформления в виде размеченного текста. К ним относятся размеченный текст на стандартизованных языках разметки XML и HTML, JSON, экранные формы с элементами управления на языке разметки HTML. Форматы размеченного текста также используются для представления данных при взаимодействии приложений в Сети, в электронных письмах и сообщениях мессенджеров.

Таким образом, имеет место технологическое разнообразие форматов обрабатываемых документов, обусловленное использованием на практике как новых, так и устаревающих стандартов форматов данных. Использование открытых форматов особенно привлекательно при совместной обработке документов, поскольку не требует использования специального инструментария.

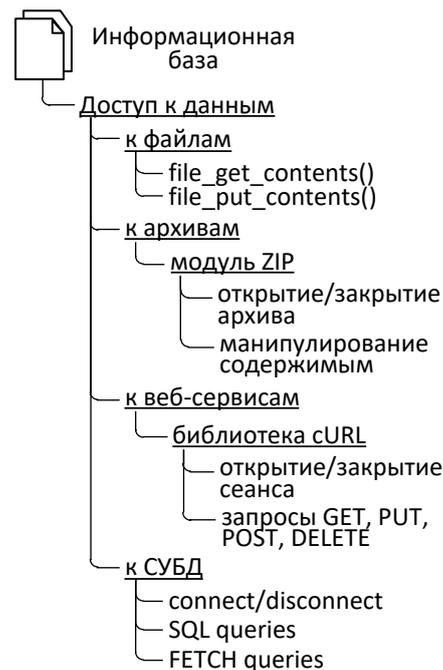
## 2.2. Источники и приемники обрабатываемых документов

Структура информационной базы требует отдельного рассмотрения в связи с разнообразием методов хранения источников и приемников обрабатываемых документов (рис. 4).

*Локальные источники/приемники данных.* Локальный метод размещения источников данных обеспечивает хранение во внешней памяти сервера в той же инфраструктуре, где находится информационная система. Локально хранятся файлы документов, архивов, размечаемый текст в виде файлов и другие собственные источники данных. Также в инфраструктуре веб-приложения на сервере, как правило, установлена СУБД, где данные помещаются в базу данных или извлекаются из нее. Могут использоваться собственные данные XML или JSON путем создания локальных микросервисов с собственной информационной базой.



**Рис. 4.** Виды источников и приемников данных.



**Рис. 5.** Виды доступа к источникам и приемникам данных в PHP.

*Удаленные источники/приемники данных.* В современных условиях информационные системы не ограничиваются локальными источниками данных, задействуют также и внешние источники, располагающиеся удаленно, передача данных в этом случае происходит по каналам сети Интернет. Удаленно хранятся файлы, архивы, файлы с размеченным текстом, электронная почта, веб-сервисы содержат данные, предоставляемые по доступу через API с использованием ключа (токена доступа API KEY). Это обеспечивается за счёт использования сетевых протоколов HTTP, HTTPS, FTP, WebDAV, IMAP, POP, SMTP. Сохранение, извлечение, удаление и обновление данных в удаленных источниках и приемниках данных называется «отображение» (от англ. mapping) из информационной базы веб-приложения во внешнюю среду хранилищ данных.

*Доступ к источникам и приемникам.* Для каждого вида источника/приемника данных требуется определенное программное средство доступа к данным – эффективный и стабильный инструмент, позволяющий получать и отправлять данные. В качестве примера (рис. 5) рассмотрим средства, которые предоставляются популярной платформой PHP, ориентированной на программирование веб-приложений.

Доступ к данным файловых источников обеспечивается за счет использования функций `file_get_contents()` / `file_put_contents()` (если в настройках сервера включена директива `allow_url_fopen`). Файлы с использованием этих функций можно получать в обработку как в локальной файловой системе, так и в файловой системе удаленных серверов.

Для работы с архивами на платформе PHP есть специализированный модуль Zip, он позволяет открывать и считывать данные из разных архивов ZIP, TAR и др. Для открытия удаленных архивов есть возможность использовать оболочки (или как их ещё называют – «обертки»). Работа архивов здесь важна для обработки офисных документов, соответствующих новым стандартам, где документ является ZIP-архивом, и его содержимое включает иерархию папок с XML-файлами. Модуль Zip позволяет открывать такие архивы и манипулировать данными внутри архива.

Доступ к веб-сервисам помогает организовать расширение CURL, обеспечивающее возможность в рамках сеанса отправлять запросы по различным протоколам, таким как HTTP.

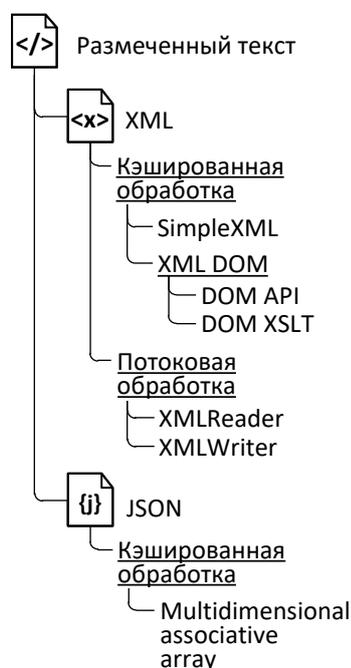
Предусмотрен стандартный набор HTTP-запросов GET, POST, PUT, DELETE, которых достаточно, чтобы обращаться к удаленным ресурсам (с использованием токенов доступа через API сервиса или без него, если доступ к ресурсу или сервису открыт).

Использование СУБД обеспечивается путем создания соединения (connection) и отправки через него запросов на языке SQL. Результаты запроса в системе PHP помещаются в массив с помощью функций FETCH. Соединение является сеансовым, постоянное соединение не поддерживается.

Таким образом, источники и приемники обрабатываемых документов характеризуются технологической разнородностью. Доступ к данным имеет существенные стилистические различия, что осложняет совместное использование на программном уровне.

### 2.3. Формат-ориентированные средства обработки разнородных данных

Разнородные данные во всем многообразии форматов требуется обрабатывать, для чего применяются специализированные средства обработки таких данных [12], специальными языками запросов для каждого формата. В данной работе эти возможности иллюстрируются на примере языка веб-программирования PHP.



**Рис. 6.** Средства обработки размеченного текста в PHP.

*Средства обработки размеченного текста.* На рис. 6 представлены средства PHP для обработки размеченного текста – XML и JSON.

Для стандартизованного XML предусмотрены два варианта обработки: кэшированная и потоковая.

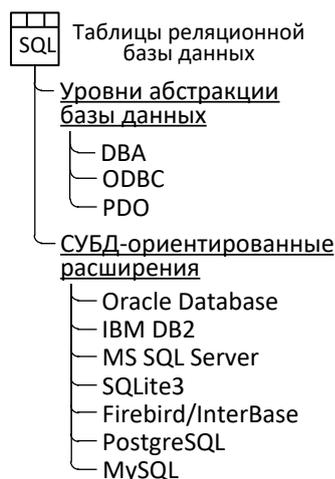
*Кэшированная XML-обработка.* При кэшированной обработке документ загружается в память целиком и затем обрабатывается в оперативной памяти с использованием объектной модели DOM (Document Object Model) [13]. Метод кэширования позволяет однократно создать или загрузить документ, а затем многократно выполнять навигацию в оперативной памяти с помощью команд DOM API. Для объектов DOM может применяться трансформация XSLT, которая позволяет преобразовывать загруженный документ в другой текстовый документ (не обязательно в XML-формате), используя стиливые XSL-шаблоны. Другой разновидностью кэшированной обработки является SimpleXML, где используются селекторы свойств объектов и итераторы массивов. В целом это тоже метод кэшированной обработки XML, отличающийся простотой в использовании. Кэшированная обработка показывает хорошие результаты при не очень объемных документах.

*Потоковая XML-обработка.* Для случаев, когда документ занимает большой объем во внешней памяти, предусмотрена потоковая обработка.

В PHP есть специализированные инструменты XMLReader / XMLWriter для чтения/записи документа по частям. Недостатком данного метода является ограниченная навигация в документе из-за того, что документ загружается в оперативную память по частям.

*Кэшированная JSON-обработка.* Для обработки размеченного текста на языке JSON используется только кэшированная обработка, основанная на загрузке документа в ассоциативный массив. В PHP для работы с JSON предусмотрены функции загрузки и выгрузки – json\_encode() и json\_decode() [14].

*Средства доступа к реляционным базам данных.* Для работы с базами данных в PHP предусмотрены специализированные средства подключения и манипулирования данными (рис. 7) – реализованные в PHP уровни абстракции базы данных и СУБД-ориентированные расширения.



**Рис. 7.** Средства доступа к реляционным базам данных в PHP.

*Уровни абстракции баз данных* – это многопрофильные инструментари, помогающие подключаться к разнообразным базам данных стандартизованным методом. К таким средствам относятся средства DBA (Database Abstraction Layer), позволяющие подключаться к СУБД типа Berkley DB. Средства ODBC (Open Database Connectivity) позволяют работать с СУБД, которые имеют семантические средства ODBC API: Adabas D, IBM DB2, iODBC, Solid, Sybase SQL Anywhere и др. Средства PDO – это объекты данных PHP, которые реализуются в базах данных, все СУБД, которые их реализовали, могут быть подключены к PHP для обработки данных, удобное универсальное средство для работы с данными на языке PHP в объектно-ориентированном стиле [15].

*СУБД-ориентированные расширения* – это еще одно семейство средств доступа к различным СУБД. Они содержат модули и драйверы для конкретных СУБД, реализованные для работы только с ними. К наиболее популярным СУБД такого рода относятся Oracle Database, MS SQL Server, SQLite3, PostgreSQL, MySQL. Эти средств «заточены» под

конкретные СУБД, в том числе под используемый диалект языка SQL.

Таким образом, многообразие форматов данных порождает соответствующее многообразие формат-ориентированных средств обработки. Если приложение настроено на работу с источником одного типа, поддерживаемого средством обработки, то при подключении источника нового типа требуется заново реализовывать метод работы с этим источником. В связи с этим возникает сложность подключения в одном приложении многих источников и методов их обработки, интеграции данных источников разного типа.

#### 2.4. Сложности поддержки учебного ИТ-проектирования

Оценим влияние выявленного многообразия категорий обрабатываемых данных, функций обработки, форматов обрабатываемых данных, типов источников и приемников данных и инструментов доступа к ним на сложность поддержки учебного ИТ-проектирования в автоматизированной информационной системе.

*Многообразие категорий обрабатываемых данных.* Широкий охват возможностей ИТ-проектирования делает необходимым программную работу вплотную со всеми категориями офисных документов – Word-, Visio-, Excel-подобных. Это многообразие вызывает трудности в подключении документов, манипулировании данными внутри них и совместном сопоставлении данных из разнородных документов.

Персонализация документов для нужд обучающихся, когда персональные данные вносятся в шаблон конструкторской документации, обуславливает необходимость работать с текстовыми, графическими и табличными данными в составе конструкторской документации. Кроме офисных документов, необходимо работать с размеченным текстом в файлах документов с внутренней разметкой XML, JSON, HTML. Письма, сообщения, таблицы реляционных баз данных, сопровождающие учебный процесс, также несут важную информацию для обработки. Они отличаются по структуре и создают трудности с реализацией обработки документов.

Проблема упрощается в условиях стандартизации категорий документов, когда появляется возможность работы с внутренним содержанием как с XML-документами, однако элементы контента, данных, стилей все же отличаются с переходом от категории к категории. Поэтому требуется модель документа, включающая элементы данных с целью создания таких документов автоматически и единообразно для каждой из категорий.

*Многообразие функций программной обработки данных.* Данные сами по себе не дают возможности поддержать учебный процесс. Они создаются и преобразовываются (служат инфор-

мационным обеспечением) в ходе выполнения функций учебного процесса. Функции ИТ-проектирования сопровождаются преобразованием данных в конструкторских документах (например, генерированием ТЗ, шаблонов КД, отчетов и экранных форм). Возможны и обратные функции по извлечению данных из КД, выдаче заданий, проверке КД и т. п. Все это обуславливает сложности, порождаемые многообразием функций поддержки учебного проектирования.

*Многообразие форматов обрабатываемых данных.* У каждой категории офисных документов прослеживается историческая линия эволюции форматов документов. Форматы документов Microsoft с начала своего пути были закрытыми, в то время как бесплатные офисные пакеты OpenOffice, LibreOffice были созданы с самого начала открытыми. Постепенно с течением времени стандарты документов не были так развиты, но развитие смещалось в сторону взаимного проникновения через файлы документов. Приходилось сталкиваться с проблемами совместимости документов разными офисными пакетами, появлялись смешанные форматы, где часть являлась бинарной, а другая часть открытой на основе XML. В настоящее время используются открытые стандарты Office Open XML (OOXML) и OpenDocument Format (ODF) на основе XML, но тем не менее разнообразие форматов документов сохраняется. Приходится работать с каждым из них, что порождает эту проблему разнообразия форматов обрабатываемых документов.

*Многообразие типов источников документов.* В современных условиях документы хранятся не только локально, но и удаленно с применением сетевых технологий доступа к ним. Невозможно представить современную АИС без собственной реляционной базы данных, где данные хранятся в таблицах, и доступ осуществляется с использованием языка запросов SQL. На первый план выходят RESTful-сервисы, цифровые удаленные архивы, микросервисы с собственной реляционной базой данных, облачные хранилища как разновидность интернет-сервиса, почтовые службы. В учебном ИТ-проектировании также используются сервисы видеоконтента с учебными видео. Все это многообразие источников документов активным образом задействовано и будет расти, что, в свою очередь, будет углублять возникающую проблему интеграции многообразия источников документов.

*Многообразие инструментов доступа к данным.* В части разработки средств доступа к данным нет недостатка инструментариев. С выходом новых стандартов и программных продуктов публикуются программный код и паттерны проектирования для работы с данными. Если взять платформу PHP, на которой разрабатывается АИС поддержки учебного проектирования, то для одних только баз данных существуют 3 вида абстракций доступа к реляционным системам, а специализированных средств доступа еще больше. Помимо этого, существуют расширения для удаленного взаимодействия с веб-сервисами. Для обработки размеченного текста есть инструментарии потоковой и кешированной обработки. Здесь усматривается многообразие инструментариев доступа к источникам данных и влечет за собой проблему сложности разработки АИС поддержки учебного проектирования.

*Пути снижения трудоемкости разработки АИС поддержки учебного проектирования.* Рассматривая функции, для реализации которых потребуется привлечь различные форматы из категорий обрабатываемых данных, складывается сложная картина разработки АИС поддержки учебного проектирования. Почти все функции завязаны на обработке различных источников данных с гетерогенными данными, что говорит о необходимости привлечения разнородных инструментов доступа. В этих условиях возникает необходимость разработки концептуальной модели, обеспечивающей возможности интеграции гетерогенных данных на высоком уровне абстракции. На сегодняшний день требуются специализированные средства, позволяющие задавать источники данных и методы их обработки в модели.

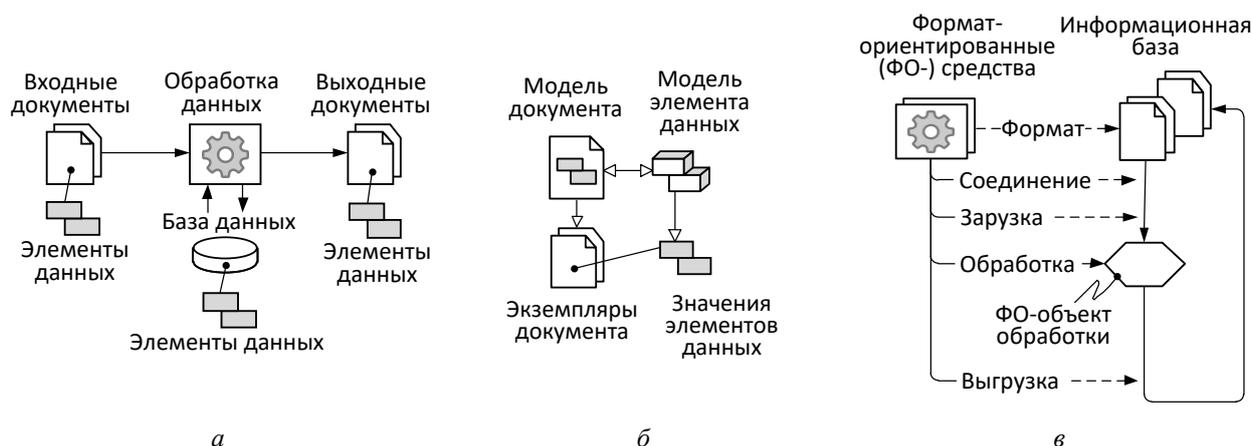
### 3. ОБОБЩЕННАЯ МОДЕЛЬ ОБРАБОТКИ РАЗНОРОДНЫХ ДАННЫХ

Рассмотренные выше многочисленные особенности обработки разнородных документов требуют единого подхода в плане совместного использования в учебном ИТ-проектировании.

Представим их в виде обобщённой модели, позволяющей единообразно рассматривать эти процессы.

### 3.1. Модели обработки данных

Ситуационно-ориентированным подходом предлагается динамическая модель обработки данных, где в каждой ситуации есть ассоциированные элементы данных, доступных средствам обработки [16]. На рис. 8 представлены модели обработки разнородных данных [17] и операции с использованием формат-ориентированных средств обработки.



**Рис. 8.** Обобщенные модели: процесса обработки разнородных данных (а); документа с обрабатываемыми элементами данных (б); типовых операций обработки (в).

*Модель процесса обработки разнородных данных.* Укрупненно модель обработки представлена на рис. 8 а, где входными и выходными данными являются документы источников данных, а также данные в базе данных. Входные и выходные документы содержат элементы данных. Элементы данных, ассоциированные с элементами документов, могут содержаться также и в базе данных. Значения элементов вводятся, изменяются или извлекаются в ходе обработки [18].

*Модель документа с обрабатываемыми элементами данных.* Сущность модель-ориентированной технологии обработки документов поясняется на рис. 8 б. Модель документа соответствует некоторому классу однородных документов. Экземпляр документа – это конкретный документ, принадлежащий данному классу. Аналогичным образом модель элемента данных соответствует некоторому классу атомарных или составных данных. Экземпляр элемента данных – это конкретное значение в экземпляре документа или в базе данных. Модель документа включает сведения об элементах данных, содержащихся в документах этого класса. Эти сведения дают возможность доступа к экземплярам элементов данных в экземплярах документа при обработке. В свою очередь, модель элемента данных задает свойства данных этого класса (тип, размещение в базе данных и др.), необходимые для доступа и обработки значений. В процессе обработки экземпляра документа, основываясь на сведениях из моделей, отыскиваются нужные элементы данных, в которые помещаются или из которых извлекаются соответствующие значения [19].

*Модель типовых операций обработки документа.* Обработка документов, несмотря на их разнородность, в обобщенном виде может быть представлена набором типовых операций (см. рис. 8 в): 1) соединение с источником данных (локальным или удаленным); 2) загрузка данных из источника в объект обработки (например, в DOM-объект); 3) обработка данных в объекте (например, трансформация документа целиком или вставка/извлечение отдельных значений); 4) выгрузка результата в приемник данных. Источники и приемники данных,

вообще говоря, разнородны по используемым форматам данных, что обуславливает использование формат-ориентированных средств обработки [20].

### 3.2. Концептуальная ER-модель обработки данных

Построим модель «сущность – связь» (ER-модель – Entity-Relationship Model), отражающую на концептуально-онтологическом уровне абстракции обобщенные классы сущностей и межклассовых отношений, которые участвуют в совместной обработке разнородных документов.

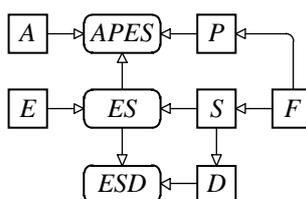
Выделенные классы сущностей представлены в табл. 2.

Таблица 2

Базовые классы сущностей

Обозначение и название	Пояснение	Пример экземпляра	Пример атрибута
$D$ – Документ	Класс конкретных документов. Экземпляр $d \in D$ соответствует некоторому конкретному документу в информационной базе. Атрибуты задают существенные свойства, относящиеся к документу в целом	Техническое задание, сгенерированное для конкретного студента	Идентификатор документа
$F$ – Формат	Класс форматов документов. Экземпляр $f \in F$ соответствует конкретному формату. Атрибуты задают свойства формата в целом, необходимые для обработки документов этого формата	Документ Word DOCX	Код формата (напр., «DOCX»)
$S$ – Схема	Класс схем – моделей структур для категорий документов, задающих компоненты документов и их связи. Экземпляр $s \in S$ представляет собой конкретную схему. Атрибуты задают свойства схемы в целом, необходимые для применения схемы при обработке документов	Шаблон технического задания курсового проекта	Идентификатор шаблона технического задания
$E$ – Элемент	Класс моделей элементов данных, задающих компоненты документов и их связи. Экземпляр $s \in S$ представляет собой конкретную модель элемента данных. Атрибуты задают свойства элемента вообще, независимо от его присутствия в том или ином документе	Идентификатор студента – исполнителя курсового проекта	Тип данных и размер идентификатора студента
$P$ – Процессор	Класс формат-ориентированных средств обработки, используемых для выполнения CRUD-операций с элементами данных документа. Экземпляр $p \in P$ соответствует конкретному процессору. Атрибуты задают свойства процессора.	PHP-модуль MySQLi для доступа к реляционным базам данных	Строка соединения базы данных с СУБД MySQL
$A$ – Алгоритм	Класс алгоритмов обработки документов. Экземпляр $a \in A$ соответствует конкретному алгоритму решения задачи обработки документов. Атрибуты задают свойства алгоритма в целом.	Алгоритм генерации технического задания курсового проекта	Идентификатор алгоритма генерации технического задания

Приведенные выше базовые классы связаны друг с другом определенными отношениями. На рис. 9 базовые сущности представлены в виде прямоугольников, а отношения между ними – с помощью бинарных связей типа «один-ко-многим», отображаемых стрелками « $\leftarrow$ » в направлении от сущности-родителя к сущности-ребенку. Некоторые сущности связаны отношениями типа «многие-ко-многим». Для их представления с помощью связей типа «один-ко-многим» введены сущности-пересечения, соответствующие парам или тройкам родительских сущностей. На ER-диаграмме сущности-пересечения представлены прямоугольниками со скругленными углами. Межклассовые отношения, заданные в ER-модели, поясняются в табл. 3.



**Рис. 9.** ER-модель обработки данных.

Таблица 3

**Связи между классами сущностей**

Обозначение связи	Пояснение
$F \rightarrow S$	Бинарная связь типа «один-ко-многим» между классами Формат («родитель») и Схема («ребенок»). Каждому экземпляру Формата может соответствовать несколько экземпляров Схемы, но каждый экземпляр Схемы принадлежит одному экземпляру Формата
$S \rightarrow D$	Бинарная связь типа «один-ко-многим» между классами Схема («родитель») и Документ («ребенок»). Каждому экземпляру Схемы может соответствовать несколько экземпляров Документа, но каждый экземпляр Документа принадлежит одному экземпляру Схемы
$F \rightarrow P$	Бинарная связь типа «один-ко-многим» между классами Формат («родитель») и Процессор («ребенок»). Каждому экземпляру Формата может соответствовать несколько экземпляров Процессора, но каждый экземпляр Процессора принадлежит одному экземпляру Формата
$E \rightarrow ES \leftarrow S$	Бинарная связь типа «многие-ко-многим» между классами-родителями Элемент и Схема. Связь-сущность <i>ES</i> («ребенок») соответствует парам экземпляров Элемента и Схемы, отражая присутствие некоторого элемента в некоторой схеме. Каждому экземпляру Элемента, как и каждому экземпляру Схемы, может соответствовать несколько экземпляров <i>ES</i> , но каждый дочерний экземпляр <i>ES</i> принадлежит одному родительскому экземпляру Элемента и одному родительскому экземпляру Схемы. Атрибуты связь-сущности <i>ES</i> задают доступ и другие специфические особенности присутствия конкретных элементов данных в конкретных схемах. Отметим, что экземпляры <i>ES</i> представляют собой мультимножество, поскольку один элемент данных может присутствовать в одной схеме несколько раз
$ES \rightarrow ESD \leftarrow D$	Бинарная связь типа «многие-ко-многим» между классами-родителями <i>ES</i> и Документ. Связь-сущность <i>ESD</i> («ребенок») соответствует парам экземпляров <i>ES</i> и Документ. Каждому родительскому экземпляру <i>ES</i> или Документа могут соответствовать несколько дочерних экземпляров <i>ESD</i> , но каждый экземпляр-ребенок <i>ESD</i> принадлежит одному экземпляру каждого из родителей. Атрибуты связь-сущности <i>ESD</i> задают, прежде всего, значение соответствующего элемента данных в конкретном документе. Принимая во внимание определение самого класса <i>ES</i> можно утверждать, что каждый экземпляр <i>ESD</i> соответствует тройке экземпляров Элемент–Схема–Документ, то есть, по существу, представляет собой тернарную связь – отражает значение некоторого элемента данных в некотором документе, который соответствует некоторой схеме
$A \rightarrow APES \leftarrow P$ $\uparrow$ $ES$	Тернарная связь типа «многие-ко-многим» между классами-родителями Алгоритм, Процессор и <i>ES</i> . Связь-сущность <i>APES</i> («ребенок») соответствует тройкам экземпляров родителей и, с учетом определения самого класса <i>ES</i> , отражает обработку некоторого элемента данных некоторой схемы с помощью некоторого процессора. Каждому родительскому экземпляру Алгоритма, Процессора или <i>ES</i> может соответствовать несколько дочерних экземпляров <i>APES</i> , но каждый экземпляр-ребенок <i>APES</i> принадлежит одному экземпляру каждого из родителей. Атрибуты связь-сущности <i>APES</i> задают свойства связи «многие-ко-многим» и другие специфические особенности

### 3.3. О целесообразности нового уровня абстракции

При создании веб-приложений разработчик на практике сталкивается с многообразием обрабатываемых данных, привязанных к функциям их программной обработки. Это создает сложности в силу разветвления видов форматов обрабатываемых данных, выходом новых стандартов в дополнение уже имеющихся. Следом для разнообразных типов источников программируются и публикуются инструментарии обработки размеченных и реляционных данных. Встраивание в приложение многообразия инструментариев, источников, форматов напрямую не дает широких возможностей интеграции данных и ведет разработчиков веб-приложений по трудоемкому пути охвата всей технологической сложности современных программных средств обработки данных. Такой подход решает задачу интеграции данных, но не позволяет преодолевать проблему технологического разнообразия.

Напрашивается вывод, что разработчику веб-приложений необходим новый подход к встраиванию разнообразных форматов источников гетерогенных данных и методов их обработки. Нужен выход на новый уровень абстракции, основанный на понятии виртуального документа – абстрактного документа, который может отображаться на разнородные реальные данные и обеспечивать единообразный подход к их обработке. Такой уровень абстракции может быть построен на базе иерархической ситуационной модели и динамических объектах обработки данных. Это становится возможным благодаря использованию открытых форматов офисных документов на основе XML-разметки. Подход, предлагающий новый уровень абстракции, снижает трудоемкость, что делает его применение целесообразным в условиях разработки веб-приложений.

### ЗАКЛЮЧЕНИЕ

1. Совместная деятельность преподавателей (в роли методистов-разработчиков и консультантов) со студентами (в роли проектантов-исполнителей) в ходе учебного ИТ-проектирования связана с обработкой разнородных конструкторских документов в условиях большого разнообразия данных, что порождает необходимость совместной программной обработки офисных документов, размеченного текста, писем и сообщений, баз данных. При этом наблюдается большое разнообразие функций программной обработки – заполнение заготовок конструкторских документов, извлечение данных из них, персонализация документов, взаимодействие с базами данных, файловыми хранилищами, веб- и микро-сервисами и др.

2. Рассмотрение технологических аспектов программной обработки разнородных данных выявило большое разнообразие как открытых, так и закрытых форматов офисных документов, при этом большинство открытых форматов основываются на XML-разметке текста. При этом требуется доступ как к локальным, так и к удаленным источникам и приемникам данных. Методы доступа к файлам, архивам, веб-сервисам и СУБД также существенно различаются в плане концепций и способов применения. Сама обработка требует использования инструментов из большого ассортимента формат-ориентированных средств. Все это обуславливает технологическую сложность разработки программного обеспечения АИС.

3. Для выработки единого подхода к задаче построена обобщенная модель обработки разнородных данных, основанная на моделях конструкторских документов со встроенными элементами данных. Значения элементов данных извлекаются из документов, сохраняются в базе данных, вставляются в заготовки документов. Собственно обработка данных выполняется формат-ориентированными средствами с помощью формат-ориентированных объектов, в которых документы загружаются, трансформируются и выгружаются во внешнюю среду. Разработанная ER-модель данных задает обобщенную концептуально-онтологическую схему классов и межклассовых отношений (связей) сущностей, участвующих в обработке.

4. На основании проведенного исследования сделан вывод о целесообразности нового уровня абстракции – виртуальных документов. В основу этого уровня абстракции следует положить разработанные обобщенную модель обработки разнородных данных и концептуально-онтологическую схему, а также открытые форматы офисных документов на основе

XML-разметки. Абстракция виртуальных документов позволит разработчику единообразно программировать совместную обработку в условиях многочисленных и разнообразных форматов, средств обработки, методов доступа, что, как ожидается, приведет к снижению трудоемкости разработки программного обеспечения АИС.

#### СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. УУНИТ. Курсовой проект «Базы данных». Кафедра автоматизированных систем управления: сайт [Электронный ресурс]. URL: <http://hsm.ugatu.su/artem/dbproj/> (дата обращения: 11.07.2024). [[UUNIT. Course project "Databases". Department of Automated Control Systems: website [Online]. Available: <http://hsm.ugatu.su/artem/dbproj/> (Accessed July, 11.2024) (In Russian).]]
2. Климова А. В., Ковтуненко А. С., Парфенов Д. В. Формирование учебного плана с использованием технологии RDF-хранилищ // СИИТ. 2022. Т. 4. № 2 (9). С. 49-55. EDN: QMYQDT. [[Klimova A. V., Kovtunenka A. S., Parfenov D. V. "Curriculum development using RDF-storage technology" // SIIT, 2022, vol. 4, no. 2 (9), pp. 49-55. (In Russian). EDN: QMYQDT.]]
3. Гусаренко А. С. Ситуационно-ориентированные базы данных: обработка гетерогенных документов микросервисов в документо-ориентированном хранилище // МОИТ. 2022. Т. 10. № 4. С.1-16. EDN: XPOGWA. DOI: 10.26102/2310-6018/2022.39.4.003. [[Gusarenko A. S. "Situation-oriented databases: processing heterogeneous documents of microservices in a document based storage" // MOIT, 2022, vol. 10, no. 4, pp. 1-16. (In Russian). EDN: XPOGWA. DOI: 10.26102/2310-6018/2022.39.4.003.]]
4. Shoufan A. Estimating the cognitive value of YouTube's educational videos: A learning analytics approach // Computers in Human Behavior. 2019. Vol. 92. Pp. 450-458. DOI: 10.1016/j.chb.2018.03.036.
5. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Ситуационно-ориентированные базы данных: обработка офисных документов // МОИТ. 2022. Т. 10. № 2. С. 1-16. EDN: WXTJMP. [[Mironov V. V., Gusarenko A. S., Yusupova N. I. "Situation-oriented databases: processing office documents" // MOIT, 2022, vol. 10, no. 2, pp. 1-16. (In Russian). EDN: WXTJMP.]]
6. Миронов В. В., Гусаренко А. С., Тугузбаев Г. А. Персонализированные шаблоны конструкторских документов в учебном проектировании: онтологические аспекты и ситуационно-ориентированная реализация // Онтология проектирования. 2023. Т. 13. № 3(49). С. 333-351. EDN: GEPSYX. [[Mironov V. V., Gusarenko A. S., Tuguzbaev G. A. "Personalization of graphic design documents in educational design: functional model of the conceptual level" // Ontology of Designing, 2023, vol. 13, no. 3(49), pp. 333-351. (In Russian). EDN: GEPSYX.]]
7. ECMA-376 [Online]. Available: <https://ecma-international.org/publications-and-standards/standards/ecma-376/> (Accessed July, 11.2024).
8. Mironov V., Gusarenko A., Yusupova N. RESTful web services development in situation-oriented databases // Proc. 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS'2020). Atlantis Press. 2020. Pp. 1-6. DOI: 10.2991/assehr.k.201029.001.
9. Гайнуллин Р. Ф., Брагин Д. Г. Анализатор диаграммных языков для Microsoft Visio // Информационные технологии. 2013. № 6. С. 18-21. EDN: OLSXSFV. [[Gajnullin R. F., Bragin D. G. "Diagramming Language Analyzer for Microsoft Visio" // Information Technologies, no. 6. 2013. pp. 18-21. (In Russian). EDN: OLSXSFV.]]
10. Guo W. et al. Information hiding in OOXML format data based on the splitting of text elements // Proc. 2019 IEEE International Conference on Intelligence and Security Informatics. 2019. Pp. 188-190. DOI: 10.1109/ISI.2019.8823564.
11. Ensign C. OpenDocument V1.3 OASIS Standard published [Online]. Available: <https://www.oasis-open.org/2021/06/16/opendocument-v1-3-oasis-standard-published/> (Accessed July, 11.2024).
12. PHP: Database Extensions – Manual [Online]. URL: <https://www.php.net/manual/en/refs.database.php> (Accessed July, 11.2024).
13. Document Object Model (DOM) Level 1 Specification [Online]. Available: <https://www.w3.org/TR/REC-DOM-Level-1/> (Accessed May, 17.2020).
14. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Мониторинг просмотров видео YouTube в образовательной среде на основе ситуационно-ориентированной базы данных и веб-сервисов RESTful // СИИТ. 2021. Т. 3. № 1(5). С. 39-49. EDN: JWDKZZ. [[Mironov V. V., Gusarenko A. S., Yusupova N. I., "Monitoring youtube video views in the educational environment based on situation-oriented database and RESTful web services" // SIIT, 2021, vol. 3, no. 1 (5), pp. 39-49. (In Russian). EDN: JWDKZZ.]]
15. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Отображение виртуальных XML-документов на таблицы MySQL в ситуационно-ориентированных базах данных: «распределенный» подход // Информационные технологии и вычислительные системы. 2017. № 1. С. 77-89. EDN: YKJYOD. [[Mironov V. V., Gusarenko A. S., Yusupova N. I. "Displaying virtual XML-documents on MySQL tables in the situation-oriented databases, "distributed" approach" // Journal of Information Technologies and Computing Systems, 2022, vol. 10, no. 2, pp. 77-89. (In Russian). EDN: YKJYOD.]]
16. Hou X. et al. Comparison of word processing document format in OOXML and ODF // Proc. 6th International Conference on Semantics, Knowledge and Grid. 2010. Pp. 297-300. DOI: 10.1109/SKG.2010.44.
17. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Структурирование виртуальных мультидокументов в ситуационно-ориентированных базах данных с помощью entry-элементов // Труды СПИИРАН. 2017. № 4 (53). С. 225-240. EDN: YUJGDH. [[Mironov V. V., Gusarenko A. S., Yusupova N. I. "Structuring virtual multi-documents in situationally-oriented databases by means of entry-elements" // Trudy SPIIRAN, 2017, no. 4 (53), pp. 225-240. (In Russian). EDN: YUJGDH.]]
18. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Ситуационно-ориентированные базы данных: интеграция XML-данных с реляционной СУБД // Системы управления и информационные технологии. 2016. № 3(65). С. 48-56. EDN: WITPMN. [[Mironov

V. V., Gusarenko A. S., Yusupova N. I. "Situation-oriented databases: integration of XML data with a relational DBMS" // Control Systems and Information Technology, no. 1(151), 2017, pp. 48–56. (In Russian). EDN: WITPMN.]]

19. Миронов В. В., Гусаренко А. С., Юсупова Н. И. Инвариантность виртуальных данных в ситуационно-ориентированной базе данных при отображении на разнородные хранилища // Вестник компьютерных и информационных технологий. 2017. № 1(151). С. 29-36. EDN: XQQRHR. [[Mironov V. V., Gusarenko A. S., Yusupova N. I. "Invariance of the virtual data" // Herald of Computer and Information Technologies, no. 1(151), 2017, pp. 29–36. (In Russian). EDN: XQQRHR.]]

20. Миронов В. В., Гусаренко А. С., Тугузбаев Г. А. Извлечение семантической информации из графических схем // Информатика и автоматизация. 2021. Т. 20. № 4. С. 940-970. EDN: YNTPOZ. [[Mironov V. V., Gusarenko A. S., Tuguzbaev G. A. // Informatika i Avtomatizatsiya, 2021, vol. 20, no. 4, pp. 940-970. (In Russian). EDN: YNTPOZ.]]

*Поступила в редакцию 12 июля 2024 г.*

#### МЕТАДАННЫЕ / METADATA

**Title:** Joint program processing of different design documents in ais supporting educational IT design.

**Abstract:** The article discusses the issues of processing heterogeneous design documents in an automated design system in the field of information technology. Aspects of collaboration between teachers and consultants and students in the role of performers are discussed. The main categories of processed data are identified, such as office documents, marked-up text, letters and messages, and databases. The features of data processing for typical functions of educational design are analyzed. Technological aspects of processing heterogeneous data are considered: the variety of document formats, local and remote sources and receivers of data, format-oriented processing tools, methods of access to files, archives, web services and DBMS are analyzed. A generalized model of document processing is constructed using format-oriented objects, as well as an ER model of the data involved in processing. It is concluded that to reduce labor intensity, the developer needs a new level of data abstraction, called virtual documents.

**Key words:** course design; external data model; concept; conceptual model; situation-oriented databases; integration of heterogeneous sources; docx; xlsx; vsdx; NoSQL, SQL

**Язык статьи / Language:** Русский / Russian.

#### Об авторах / About the authors:

##### ГУСАРЕНКО Артем Сергеевич

Уфимский университет науки и технологий, Россия.  
Доцент. каф. автоматизированных систем управления. Дипл. информатик-экономист (Уфимск. гос. авиац. техн. ун-т, 2010). Канд. техн. наук по мат. и прогр. обеспечению выч. машин, комплексов и комп. сетей (там же, 2013). Иссл. в обл. иерархич. моделей и ситуац. управления.  
E-mail: gusarenko@ugatu.su  
ORCID: <https://orcid.org/0000-0003-4132-6106>  
URL: [https://elibrary.ru/author\\_profile.asp?authorid=692073](https://elibrary.ru/author_profile.asp?authorid=692073)

##### МИРОНОВ Валерий Викторович

Уфимский университет науки и технологий, Россия.  
Проф. каф. автоматизированных систем управления. Дипл. радиофизик (Воронежск. гос. ун-т, 1975). Д-р техн. наук по упр. в техн. системах (Уфимск. гос. авиац. техн. ун-т, 1995). Иссл. в обл. иерархических моделей и ситуац. управления.  
E-mail: mironov@list.ru  
ORCID: <https://orcid.org/0000-0002-0550-4676>  
URL: [https://elibrary.ru/author\\_profile.asp?authorid=691759](https://elibrary.ru/author_profile.asp?authorid=691759)

##### GUSARENKO Artem Sergeevich

Ufa University of Science and Technology, Russia.  
Docent, Dept. of Automated Control Systems. Dipl. Informatic economist (Ufa State Aviation Technical University, 2010). Cand. of Tech. Sci. on mathematical and software support for computers, complexes, and computer networks (ibid, 2013). Research in the field of hierarchical models and situational management.  
E-mail: gusarenko@ugatu.su  
ORCID: <https://orcid.org/0000-0003-4132-6106>  
URL: [https://elibrary.ru/author\\_profile.asp?authorid=692073](https://elibrary.ru/author_profile.asp?authorid=692073)

##### MIRONOV Valeriy Viktorovich

Ufa University of Science and Technology, Russia.  
Prof., Dept. of Automated Control Systems. Dipl. Radio physicist (Voronezh State Univ., 1975). Dr. of Tech. Sci. (Ufa State Aviation Technical University, 1995). Research in the field of hierarchical models and situational management.  
E-mail: mironov@list.ru  
ORCID: <https://orcid.org/0000-0002-0550-4676>  
URL: [https://elibrary.ru/author\\_profile.asp?authorid=691759](https://elibrary.ru/author_profile.asp?authorid=691759)