БУ ВО

«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Ханты-Мансийского автономного округа – Югры»

Кафедра автоматизированных систем обработки информации и управления (АСОИУ)

ОТЧЁТ

по учебной практике, по получению первичных профессиональных умений и навыков

«Разработка инструмента мониторинга очереди задач и заданий СУД DIRECTUM. Методы нахождения причин повышенной очереди»

Выполнил студент гр. 606-71м

Боровков Дмитрий Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Руководитель практики:

профессор кафедры АСОИУ, д. т. н.

Бушмелева Кия Иннокентьевна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Сургут, 2018

СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Пользователи, работая в системе DIRECTUM, активно пользуются её главной функцией – документооборотом. Руководители, отправляя задачи по актам, планам, резолюциям своим подчиненным, освобождены от труда по доставке уведомлений подчиненным, по постоянному контролю хода выполнения задач. Все эти функции выполняет сама система, используя ресурсы серверов. В случае большого количества подобных запросов руководителей ресурсы серверов заполняются и может произойти зависание всей системы. Для минимизации или предотвращения подобных случаев необходимы инструменты мониторинга хотя бы основных показателей системы DIRECTUM. Одним из таких показателей является размер очереди задач и заданий системы DIRECTUM, находящихся в обработке.

В ходе производственной практики была поставлена задача: создать инструмент мониторинга значения данного параметра, а также изучить методы поиска причин роста очереди задач и заданий DIRECTUM.

ОПРЕДЕЛЕНИЯ

Система управления документами ‑ комплекс программных и технических средств, организационных и распорядительных документов, позволяющих автоматизировать процессы управления документами [1].

Задание ‑ объект, предназначенный для автоматизации деловых процессов. Задания создаются автоматически, в ходе обработки маршрутов задач. Одно задание всегда адресовано только одному пользователю. В задании можно посмотреть описание работ, заданное в задаче, и можно дописать свой текст, например, текст с описанием выполненных работ[2].

Задача ‑ объект, предназначенный для автоматизации деловых процессов. В задаче описываются работы, которые необходимо выполнить, задается порядок их выполнения, а также пользователи, которые будут выполнять эти работы. В рамках задачи может быть создано множество заданий [2].

Маршрут задачи – список исполнителей по задаче с указанием последовательности и сроков выполнения заданий[2].

Справочник ‑ компонента системы DIRECTUM, предназначенная для хранения структурированной информации. Справочник представляет собой список записей. В системе DIRECTUM имеется множество справочников, предназначенных для работы с разной информацией, например, «Организации», «Работники», «Подразделения», «Номенклатура дел». В разных справочниках разный состав полей [2].

Служба Workflow ‑ служба Windows «IS-Builder Workflow Processing», обеспечивающая обработку маршрутов задач [3].

Файловое хранилище ‑ хранилище текстов документов, у которого в справочнике «Хранилища текстов документов» поле «\*Тип» имеет значение «Файловое». Тексты документов, размещаемых в таких хранилищах, хранятся в файловой системе [3].

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1. О СУД DIRECTUM

### 1.1.1. Система управления документами

Эффективная организация во многом зависит от внедрения и поддержания системы управления, которая обеспечивает постоянное улучшение деятельности и отвечает потребности всех заинтересованных сторон. Системы управления обеспечивают методологию принятия решений и управления ресурсами для достижения целей организации.

Создание документов и управление ими является неотъемлемой частью любого вида деятельности или делового процесса организации, обеспечивая деловую результативность, подотчетность, управление рисками и непрерывность деловой деятельности, а также давая организации возможность извлекать пользу из обладающих ценностью информационных ресурсов.

Внедрение в организации СУД позволяет также обеспечить прозрачность и прослеживаемость решений, принятых руководством и подотчетность.

Цель внедрения СУД заключается в создании документов и управления ими систематическим и контролируемым образом для того, чтобы[1]:

1. Эффективно осуществлять деловую деятельность и предоставлять услуги.
2. Соответствовать законодательным и иным нормативным правовым актам, а также требованиям подотчетности.
3. Оптимизировать принятие решений, обеспечить согласованность и непрерывность деятельности организации.

Система управления документооборотом DIRECTUM поддерживает полный жизненный цикл управления документами, при этом традиционное бумажное делопроизводство органично вписывается в электронный документооборот.

На базе DIRECTUM разработан широкий набор бизнес-решений, направленный на решение бизнес-задач с определенным бизнес-эффектом.

### 1.1.2. Основные компоненты СУД DRIECTUM

На Рис. изображена общая схема СУД DIRECTUM.



Рис. 1. Общая схема СУД DIRECTUM

1. Клиентская часть DIRECTUM – набор программ и компонентов системы, выполняемых на компьютере пользователей и обеспечивающих доступ к данным системы DIRECTUM по локальной сети.
   1. SBRte.exe – среда исполнения IS-Builder System Runtime Environment, выполняет основные действия в системе, например, открытие проводника, выполнение сценариев.

Взаимодействие с другими компонентами:

* база данных ‑ запрашивает объекты системы;
* сервер сеансов ‑ проверка лицензий и установка блокировок на объекты системы;
* служба файлового хранилища ‑ получение прав на файлы документов;
* файловое хранилище ‑ получение содержимого документа;
* служба преобразования ‑ преобразование документов в форматы PDF, HTML и TIFF;
* служба ввода.

Клиентская часть инициирует и прекращает обработку маршрутов задач, помещая информацию о них в таблицу SBWorkflowProcessing базы данных.

* 1. SBLogon.exe – служба паролей, обеспечивающая аутентификацию пользователей при подключении к системе.

Взаимодействие с другими компонентами:

При подключении пользователя к системе программа SBLogon.exe обращается к базе данных и проверяет, что пользователь зарегистрирован в системе.

1. Сервер сеансов – служба Windows.

Основные функции:

* учет свободных лицензий и подключение пользователей к системе DIRECTUM;
* блокировки объектов (Пока один пользователь редактирует объект, другой пользователь не сможет вносить свои изменения);
* управление кэшированием справочников.

Взаимодействие с другими компонентами:

* SBRte.exe и служба Workflow обращаются к серверу сеансов для установки блокировок на объекты системы в базе данных и файловом хранилище;
* сервер сеансов обращается к службе файловых хранилищ, чтобы она забрала права доступа у пользователя, когда снимается блокировка с документа.

1. Служба Workflow – служба Windows, обеспечивающая обработку маршрутов задач.

Взаимодействие с другими компонентами:

* служба Workflow запрашивает задачи, задания уведомления, вложения и объекты, используемые в типовых маршрутах, в базе данных;
* сервер сеансов ‑ установка блокировок на задачи, задания, уведомления, вложения и объекты, используемые в типовых маршрутах;
* служба файлового хранилища ‑ получение прав на файлы документов;
* файловое хранилище ‑ получение содержимого документа;
* служба преобразования ‑ преобразование документов в форматы PDF, HTML и TIFF;
* служба Workflow передает данные агенту служб взаимодействия систем при обработке задач для пользователей удаленных систем.

1. Службы обработки событий – службы Windows, которые обрабатывают серверные события.

Взаимодействие с другими компонентами:

* служба обработки событий запрашивает объекты системы в базе данных;
* сервер сеансов ‑ установка блокировок на объекты;
* служба файлового хранилища ‑ получение прав на файлы документов;
* файловое хранилище ‑ получение содержимого документа.

1. Файловые хранилища – папка в файловой системе, в которой хранятся файлы с текстами документов системы DIRECTUM.
2. Служба файловых хранилищ ‑ служба Windows, управляющая доступом к файлам хранилищ текстов документов системы DIRECTUM и устанавливаемая на каждый из компьютеров, на которых размещаются файловые хранилища.
3. Служба преобразования документов – служба Windows, предназначенная для преобразования документов из различных форматов в форматы PDF и HTML, например, при экспорте или импорте документа, создании версии документа.
4. Служба ввода документов – служба Windows, предназначенная для автоматического ввода документов, полученных из файловой системы, с факса или электронной почты в систему DIRECTUM, на почтовый сервер или произвольную программу. Позволяет преобразовывать документы в многостраничный TIFF-файл и извлекать из них штрих-коды при вводе документов.

## 1.2. Схема группы таблиц задач и заданий

В рамках выполнения плана работ, акта или др. документов создаются задачи. В задаче описываются работы, которые необходимо выполнить, задается порядок их выполнения, а также пользователи, которые будут выполнять эти работы. Задача отправляется по типовому маршруту: указывается, в каком порядке задача будет проходить по пользователям (исполнителям задачи). В рамках задачи создается несколько заданий, они создаются автоматически для исполнителей задачи.

На рис. Представлена схема групп таблиц задач и заданий в СУД DIRECTUM [4].

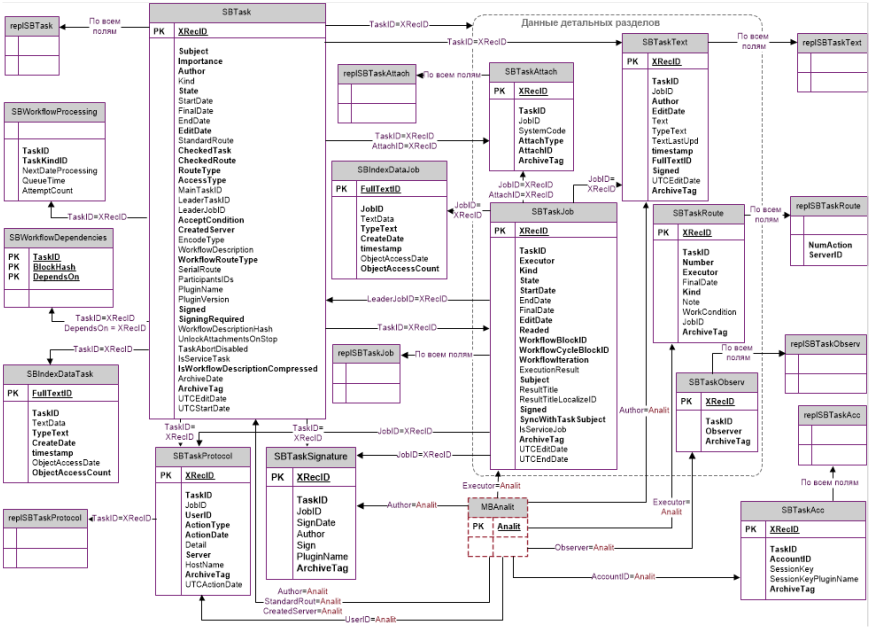


Рис. 3. Схема группы таблиц задач и заданий в СУД DIRECTUM

Основные параметры всех когда-либо созданных задач хранятся в таблице SBTask.

Таблица SBTask – базовая таблица проводника системы. Содержит данные о задачах. Обновляется при создании, старте, выполнении, прекращении, возобновлении и подписании задачи.

Далее идет описание таблиц с детальной информацией о задачах и заданиях: SBTaskJob, SBTaskAttach, SBTaskText, SBTaskRoute, SBTaskObserv.

Основные параметры всех когда-либо созданных заданий хранятся в таблице SBTaskJob.

Таблица SBTaskJob –. содержит данные о заданиях и уведомлениях. Обновляется при создании, изменении, прекращении, выполнении и подписании заданий.

Связь между задачей и заданием на уровне базы данных осуществляется через поле LeaderJobID таблицы SBTask и ключевое поле XRecID таблицы SBTaskJob, отношение один ко многим соответственно.

XRecID – идентификатор записи в таблице. LeaderJobID – идентификатор ведущего задания.

Таблица SBTaskAttach –. содержит данные о вложениях в созданные задачи или задания. Обновляется при добавлении/удалении вложений задач или заданий в компоненте.

Таблица SBTaskText – содержит данные о текстах задач и заданий. Обновляется при редактировании и подписании текста в задачах и заданиях системы.

Таблица SBTaskRoute –.содержит данные о каждом этапе маршрута задач. Обновляется при редактировании данных маршрута задач.

Таблица SBTaskObserv – содержит данные о наблюдателях задач. Обновляется при определении наблюдателей во время создания задачи.

Таблица SBIndexDataTask –содержит слепки задач (изображения), используемые для общего поиска задач по их карточке. Слепки задач включают в себя значения реквизитов Тема задачи, Типовой маршрут, Вид задачи. В таблице первичным ключом является поле FullTextID – идентификатор для полнотекстового поиска, по которому находится слепок задачи.

Таблица SBIndexDataJob –содержит слепки заданий, используемые для общего поиска заданий по их карточке. Аналогия с SBIndexDataTask.

Таблица SBWorkflowDependencies – содержит список зависимых друг от друга задач по блоку мониторинг. Новые записи в таблицу добавляются при обработке задачи, зависимой по блоку мониторинг от незавершенных задач. Записи удаляются из таблицы при завершении, прекращении, удалении задачи с блоком мониторинг и задач, от которых она зависит.

Блок «Мониторинг» является блоком в алгоритме типового маршрута, который указывает задаче ожидать до завершения некоторого действия, например завершения другой задачи.

Таблица SBTaskAcc – содержит данные о правах доступа к задачам. Обновляется при изменении прав доступа в карточке задач.

Таблица SBTaskProtocol – содержит протокол работы с задачами и заданиями. Обновляется при любых действиях над задачами и заданиями.

Таблица SBTaskSignature – содержит данные о подписании задач и заданий (данные о задачах/заданиях хранятся соответственно в таблицах SBTask и SBTaskJob). Обновляется при подписании задач и заданий.

Таблица SBWorkflowProcessing – таблица предназначена для хранения информации о задачах, принятых к обработке и об обрабатываемых в текущий момент службой IS-Builder Workflow Processing. Информация о задаче в таблицу заносится при старте задачи, при выполнении задания по задаче и при завершении работы задачи, от которой зависит выполнение других задач. Удаляется при завершении очередного этапа обработки маршрута и при прекращении или удалении задачи. Связан напрямую с таблицей SBTask через ключ TaskID таблицы SBWorkflowProcessing и ключ XRecID таблицы SBTask.

Таблица MBAnalit – содержит значения реквизитов раздела «Карточка» всех справочников. Такие данные как автор задачи, задания хранятся в справочнике Работники.

Таблицы, где в начале есть “repl” являются копиями таблицы без “repl”. Например, таблица replSBTask является копией таблицы SBTask. Таблица repl\* необходима для репликации базы данных на другой сервер. Например, у предприятия есть два филиала в разных городах. Первый город имеет главную базу данных, другой город вторичную, то есть реплицируемую базу данных. Интернет-соединение в одном из городов низкое, тогда, чтобы не отправлять запрос на открытие, добавление и т. Д. какого-нибудь документа в главный город, в филиале есть своя база данных, в которой ведется ее редактирование, и все изменения затем отправляются на главный сервер в конце дня. Таким образом, поддерживается возможность использования копии базы данных.

## 1.3. Важность мониторинга очереди задач и заданий DIRECTUM

В первую очередь служба очереди задач и заданий (служба Workflow) является основным компонентом системы DIRECTUM наряду с сервером сеансов, службой обработки событий, службой файловых хранилищ, базой данных, службой ввода и преобразования документов.

Служба Workflow обрабатывает маршруты задач по типовым маршрутам. Таблица, непосредственно связанная с данной службой (SBWorkflowProcesing) содержит все задачи, которые обрабатываются в данный момент или только что принятые к обработке.

Уже по количеству обрабатываемых задач одновременно можно судить о загруженности системы. Так как для каждой задачи необходимо создать задания для каждого исполнителя, направить уведомления исполнителю о получении задания, то даже незначительное количество задач, обрабатываемых службой Workflow, могут сильно нагрузить систему вплоть до неработоспособного состояния. Так как служба Workflow является службой Windows, которая работает под специальным техническим пользователем, служба может занимать неограниченный объем оперативной памяти, нагружать процессор до 100%, соответственно при захвате всей оперативной памяти сервер зависнет и служба, соответственно, тоже.

# 2. ПОСТАВНОВКА ЦЕЛИ И ЗАДАЧ

Цель: Разработать инструмент мониторинга очереди задач и заданий DIRECTUM и изучить метод поиска причин повышенной очереди задач и заданий.

Задачи:

1. Изучить общие сведения и структуру СУД DIRECTUM.
2. Изучить один из основных компонентов системы: служба Workflow.
3. Задать требования к входной, выходной информации инструмента.
4. Разработать инструмент мониторинга очереди задач и заданий DIRECTUM.
5. Изучить методы поиска причин повышенной очереди задач и заданий DIRECTUM с помощью инструмента мониторинга очереди задач и заданий DIRECTUM.

Требования к входным данным:

1. Система должна обеспечить возможность выполнения SQL-запросов к базе данных Directum.

Требования к выходным данным:

1. Цветовое кодирование числа, показывающего общее количество задач и заданий.
2. Группировка задач и заданий по видам.
3. Представление количества задач и заданий в виде графика в разрезе времени.

Функциональные требования:

1. Возможность масштабирования шкалы времени.
2. Автообновление информации об очереди каждые 20 секунд.
3. Проверка учетной записи на возможность просмотра инструмента.

Нефункциональные требования:

1. Инструмент мониторинга не должен загружать оперативную память сервера.
2. Производительность сервера не должна зависеть от количества пользователей, одновременно использующих инструмент мониторинга.
3. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ
   1. Контекстная диаграмма

На рис. 4. Представлена контекстная диаграмма разрабатываемого Инструмента мониторинга по методологии IDEF0.

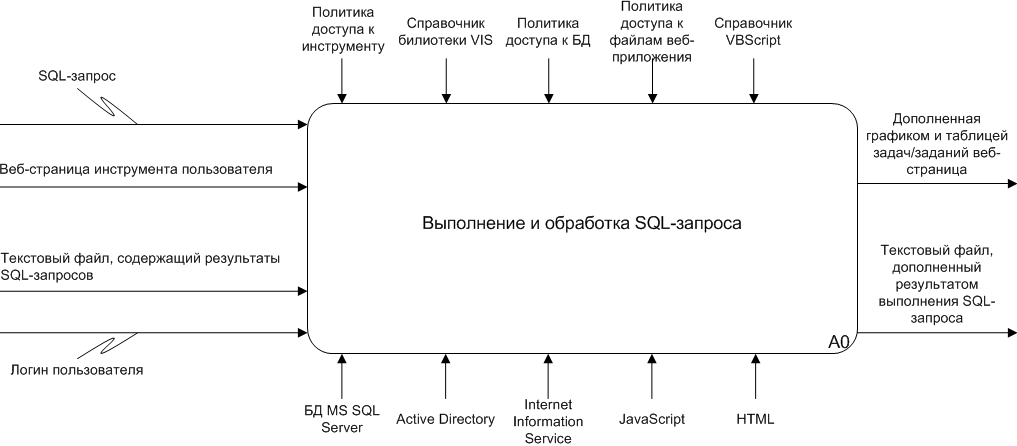


Рис. 4. Контекстная диаграмма A0

Главной функцией Инструмента мониторинга является выполнение запросов к БД и обработка их результатов.

Изначально предполагается, что есть текстовый файл, который после каждого запроса к БД будет дополняться новой информацией из БД об очереди задач и заданий. Информация берется с помощью SQL-запроса.

Например, каждые 20 секунд формируется новый SQL-запрос, который берет новую информацию об очереди и добавляет к файлу.

Пользователь, который желает получить новую информацию об очереди из файла или впервые открыть веб-страницу Инструмента должен пройти этап проверки доступа к Инструменту. Специальный метод проверки доступа использует имя пользователя для сравнения с именами членов группы пользователей, созданной для ограничения доступа к Инструменту, в Active Directory.

В случае успеха проверки, пользователь отправляет запрос серверу на получение новой информации об очереди из файла, ответ сервера – HTML-вставки в веб-страницу клиента, формирующие график и таблицу.

Скрипты на стороне клиента написаны на JavaScript, на стороне сервера – VisualBasic.

Сам сайт обслуживается программой IIS.

Входные данные:

1. SQL-запрос – запрос на языке SQL для получения информации об очереди задач и заданий.
2. Веб-страница Инструмента пользователя – веб-страница пользователя, в котором выполняется скрипт-запрос на получение новой информации об очереди. Если клиент впервые открывает веб-страницу, то дополнительно осуществляется проверка полномочий пользователя.
3. Текстовый файл, содержащий результаты SQL-запросов – файл, в котором после каждого SQL-запроса Инструмента добавляется информация об очереди. Клиенты достают информацию об очереди из этого файла.
4. Логин пользователя – логин клиента.

Управление:

1. Политика доступа к Инструменту – обязательность разделения пользователей на тех, кто имеет доступ к Инструменту и на тех, кто не имеет.
2. Справочник библиотеки VIS – библиотека, которая используется для создания графика очереди задач и заданий, которая также дает возможность масштабирования графика по времени.
3. Политика доступа к БД – так как, веб-приложение обслуживается IIS, то все приложение можно представить как процесс в ОС Windows, который работает от какого-то пользователя. Этот пользователь должен обладать правами на чтение данных из БД SQL и Active Directory.
4. Политика доступа к файлам веб-приложения – доступа к файлам должен иметь лишь локальный администратор сервера.
5. Справочник VBScript – скрипт проверки полномочий написан на VB Script.

Механизмы:

1. БД MS SQL Server – БД СУД Directum, из которой извлекается информация об очереди.
2. Active Directory – из её БД извлекается информация о членах группы пользователей, созданной для ограничения доступа к Инструменту.
3. Internet Information Service – программа, обслуживающая веб-приложение, принимающая запросы клиентов и отправляющая им ответы.
4. JavaScript – скрипты на стороне клиента написаны на этом языке.
5. HTML – язык, задающий структуру и содержание веб-страницы.

Выходная информация:

1. Дополненная графиком и таблицей задач и заданий веб-страница клиента – в случае, если веб-приложение обслужило запрос клиента.
2. Текстовый файл, дополненный результатом выполнения SQL-запроса – в случае, если веб-приложение отправило запрос на получение информации об очереди к БД СУД Directum.
   1. Декомпозиция контекстной диаграммы

На рис. 5. Представлена декомпозиция контекстной диаграммы.

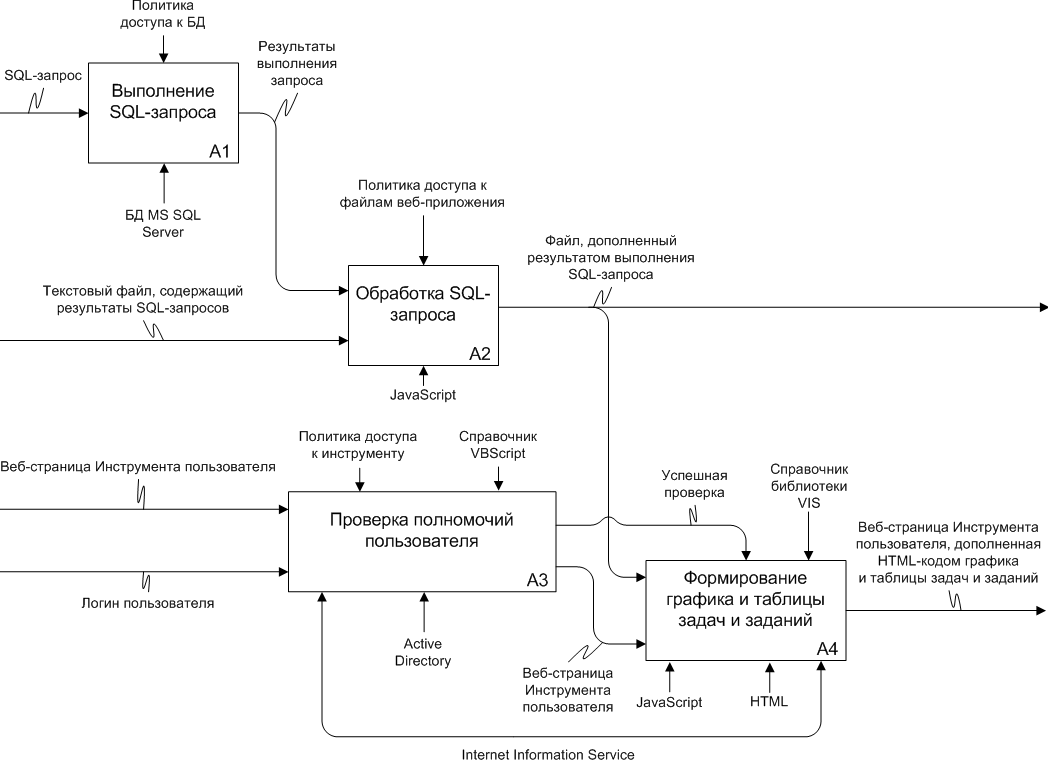


Рис. 5. Декомпозиция контекстной диаграммы

Блоки A1 и А3 могут выполняться параллельно, так как запрос веб-приложение к БД и запрос клиента к веб-приложению не зависят друг от друга.

Блок А1 выполняет SQL запрос к БД СУД Directum, блок А2 обрабатывает ответ, то есть записывает ответ в файл.

Код метода выполнения SQL-запроса и записи результата в файл объединен один.

* + 1. Описание выполнения блоков А1 и А2.

Информация обо всех задачах и заданиях, находящихся в данный момент в обработке, хранится в БД СУД Directum. Для получение данной информации необходим SQL-запрос. Текст SQL-запроса представлен в приложении 1.

Для получения ID всех задач (TaskID) необходим запрос к таблице SBWorkflowProcessing – задачи и задания, находящиеся в обработке. С помощью функции COUNT по ID задач считается количество записей таблицы. Для получения данных и видах задач необходимо связать задачи таблицы SBWorkflowProcessing с записями таблица MBAnalit. Это можно сделать через таблицу SBTask. Таблица SBWorkflowProcessing связывается с таблицей SBTask через ИД задач. Таблица SBTask связывается с таблицей MBAnalit через ИД типового маршрута задачи таблицы SBTask (StandardRoute) с ключом таблицы MBAnalit (Analit). Имя типового маршрута хранится в поле NameAn таблицы Analit.

В приложении 2 представлен текст метода, выполняющего формирование и запуск SQL-запроса и запись результатов в файл.

Переменная objConnection – переменная-объект подключения к БД.

Метод Open открывает подключение к БД. Его свойства:

1. Provider – определяет драйвер, который будет использован для подключения к БД.
2. Integrated Security – равен SSPI, если БД поддерживает проверку подлинности Windows, чтобы не указывать в ConnectionString имя пользователя и пароль.
3. Persist Security Info – необходимость использования шифрованного канала.
4. User ID – имя пользователя БД, от которого будет выполняться запрос.
5. Initial Catalog – имя БД
6. Data Source – имя сервера БД.

objRS – переменная, содержащая результат выполнения SQL-запроса.

FSO – переменная, содержит объект для работы с файловой системой.

Textstream – поток файла, открытого с помощью метода FSO – OpenTextFile.

Вначале идет запись текущей даты и времени, для этого используются зарезервированные переменные: date, time.

Далее циклом производится запись каждой строки запроса – первый столбец строки – наименование вида задач и заданий, второй – количество зада и заданий данного вида.

После записи, закрывается соединения с БД и закрывается поток записи файла.

* + 1. Описание блока А3

Блок А3 предназначен для проверки полномочий пользователя перед выполнением его запроса.

Проверка полномочий пользователей

В целях выполнения политики безопасности круг пользователей, имеющих доступ к Инструменту мониторинга, должен быть ограничен.

Разрабатываемый инструмент должен иметь метод идентификации пославшего запрос пользователя, метод аутентификации (проверка пользователя на включение его в определенную группу Active Directory), авторизации (предоставление пользователю прав на использование Инструмента).

В приложении 3 представлен код на языке Visual Basic Script, который реализует методы идентификации, аутентификации и авторизации.

objRootDSE – объект RootDSE, предоставляет различную информацию о возможностях LDAP-сервера (сервера Active Directory).

strDNSDomain – переменная, содержащая наименование домена, которому принадлежит компьютер (сервер), на котором располагается Инструмент.

StrBase – добавляет к началу переменной протокол LDAP (по аналогии с HTTP).

Для выполнения SQL команд необходимо использовать объект Command.

adoCommand – хранит объект Command. Его свойства:

* ActiveConnection – аналогичен свойству Open объекта ADODB.Connection.
* Page Size – задает число записей на странице.
* Timeout – число секунд ожидания выполнения команды.
* CommandText – текст команды.

Следующее, что нужно сделать – назначить объекту Command объект подключения – ADODB.Connection.

adoConnection – хранит объект подключения. Его свойства:

* Provider – определяет драйвер, который будет использован для подключения к базе данных.
* Open – connection string, строка, определяющая параметры подключение к БД.

strFilter – устанавливает фильтр для поиска группы, в которой содержаться пользователя, которым разрешен доступ к Инструменту.

strQuery – завершающий запрос, проставляется имя домена, фильтры для поиска группы, кого выбрать из этой группы и выбирать ли членов групп, которые являются членами целевой группы.

adoRecordSet – переменная результаты выполнения команды в БД Active Directory.

intCount – переменная, равная нулю в случае, если пользователь, отправивший запрос, не найден, единице – в другом случае.

Далее начинается цикл, который работает, пока не будут пройдены все строки запроса.

Следующий цикл for рассматривает поля objField для каждой строки adoRecordSet.

Условие в цикле for проверяет, является ли текущее поле текущей строки массивом. Если это так, то проверяется каждое значение массива на схожесть с именем пользователя. В случае успеха переменная intCount увеличивается на единицу и все циклы и метод завершают работу. Метод же передает значение переменной intCount в качестве результата, что говорит о том, что пользователь прошел идентификацию, аутентификацию и авторизацию.

* + 1. Описание блока А4

Блок А4 предназначен для формирования графика и таблиц задач и заданий.

Для полного понимания принципа работы блока А4 необходимо дать описание работы веб-страницы на стороне клиента.

Клиентская часть инструмента представлена в приложении 4.

Для сокращения объема данного отчета, в приложении 4 представлен только скрипт клиентской веб-страницы.

Переменная I – идентификатор для включения и остановки бесконечного повторения выполнения функции Update.

Переменная dataset – массив точек для графика очереди.

Главная функция, выполняющаяся в скрипте ‑ &(function(){}). Данная функция запускает функцию Update в первый раз при запросе веб-страницы клиентом.

Функция ClearInterval останавливает бесконечное выполнение данной функции, чтобы не случилось лавинообразное выполнение функции, так как чтение из файла очереди может занять больше времени, чем интервал между запусками данной функции.

Функция post загружает данные с сервера, а именно HTML-код таблицы задач и заданий. Его аргументы:

1. URL – адрес, на который отправляется запрос.
2. Data – данные, отправляемые на сервер в формате «Ключ-Значение».
3. Функция, вызываемая в случае успешного выполнения (data3).

Согласно передающимся аргументам функции, на сервере выполнится скрипт Table.asp.

В функции data3 результат выполнения POST запроса вставляется в элемент с идентификатором «content».

Следующие две функции append добавляют контент внутрь элементов с идентификаторами tg.

Веб-страница делится на два поля: левая – для графика, правая для таблицы. Каждые 20 секунд обновляются таблица и график с помощью подачи веб-страницей клиента запроса на сервер. Кривая выходит через создание точек, где по Х – текущее системное время, по Y – общее количество задач. Изображение графика создается при помощи открытой библиотеки vis.js и vis.css. Таблица создается при помощи функции Table.asp. Из ячейки таблицы «Итого без ТМ:» выбирается значение для добавления точки к графику. Не используется значение ячейки «Итого:» так как они учитывают все задачи, даже те, которые не загружают систему.

Программой задано цветовое оповещение: при значении больше 200 фон ячейки значения «Итого без ТМ:» становится оранжевым. При значении больше 300 – красным.

Переменная graph2d содержит объект vis – график.

Переменна options параметры графика.

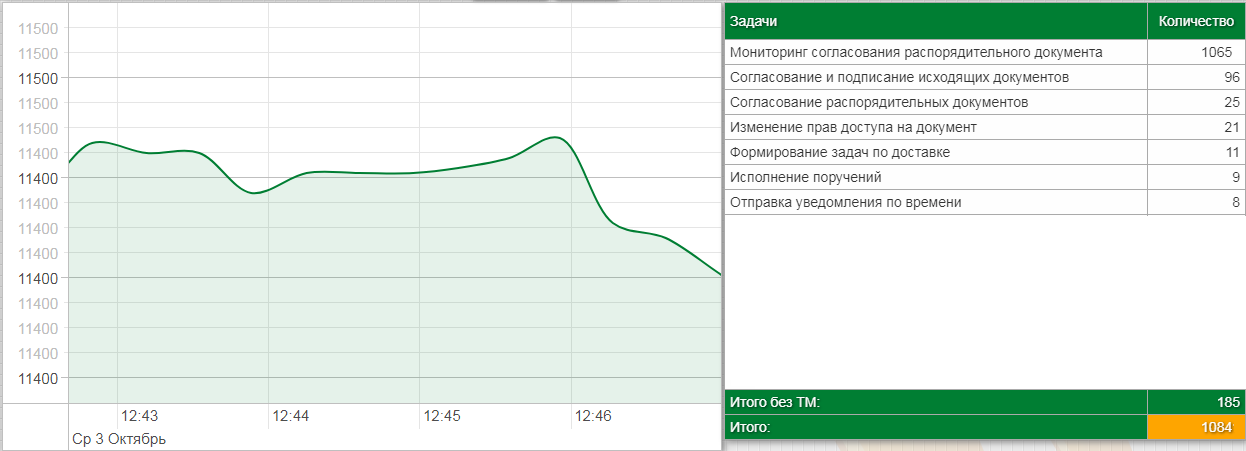


Рис. Пользовательский интерфейс инструмента мониторинга задач и заданий DIRECTUM

Таблица справа на рис. Позволяет понять, какие виды задач загружают систему в данный момент.

# 4. МЕТОДЫ ПОИСКА ПРИЧИН ПОВЫШЕННОЙ ОЧЕРЕДИ ЗАДАЧ И ЗАДАНИЙ DIRECTUM

В случае если суммарное количество задач превысило некоторый порог (порог устанавливается только опытным путем, так как сильно зависит от технических характеристик сервера), то стоит обратить внимание на виды задач и их количество в данный момент (рис. 5):



Рис. 5. Таблица инструмента мониторинга, показывающая количество выполняемых задач и заданий в разрезе вида задачи

Есть несколько видов задач, которых может быть много, например, «Согласование и подписание исходящих документов», это значит, что, например, руководитель выполнил рассылку резолюций подчиненным и др.

Если наблюдается большое количество задач различных видов, то проблема связана с недоступностью одного из основных компонентов, с которым Workflow взаимодействует (файловое хранилище, база данных, сервер сеансов) или с самой службой Workflow.

По всем основным компонентам системы ведут отчеты (логи) по возникающим ошибкам (рис. 6). Существует программа, которая агрегирует все виды ошибок файловых хранилищ и службы Workflow. В случае остановки работы какого-либо основного компонента ошибок в отчете будет заметно больше, чем при стационарной работы компонента. Поэтому по агрегированному отчету можно выяснить тип ошибки и компонент системы:

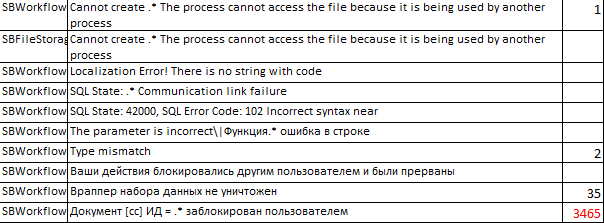


Рис. 6. Таблица, показывающая количество ошибок, зафиксированных компонентом workflow

На рис. Количество ошибок типа «Документ заблокирован пользователем» компонента Workflow заметно выше, значит ошибка возникла в Workflow из-за массовой блокировки документов.

Иногда высокая очередь задач и заданий говорит о нарушении работы другого компонента (рис. 7).

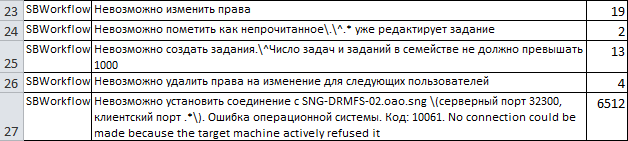


Рис. 7. Таблица, показывающая количество ошибок, зафиксированных компонентом workflow

В данном случае проблемы с доступом к файловому хранилищу, из-за чего пользователи не могут получить доступ к необходимым объектам и накапливается очередь Workflow.

# ЗАКЛЮЧЕНИЕ

За время прохождения производственной практики изучена общая схема СУД DIRECTUM и схема группы таблиц задач и заданий. Реализован механизм мониторинга количества задач, находящихся в обработке, в разрезе видов задач. Описаны методы, по которым возможно отследить причины высокой очереди задач.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Информация и документация. Системы управления документами. Основные положения и словарь : ГОСТ Р ИСО 30300-2015 . - Издание официальное. -М. : ИПК Издательство стандартов, 2016. -19 с.
2. DIRECTUM 5.6 Руководство пользователя [Электронный ресурс] [2018] – Режим доступа: http://download.directum.ru/110996ghj42wdfgt6489/DIRECTUM%205%206%20%20%d0%a4%d0%b8%d0%b7%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b0%d1%8f%20%d1%81%d1%82%d1%80%d1%83%d0%ba%d1%82%d1%83%d1%80%d0%b0%20%d0%b4%d0%b0%d0%bd%d0%bd%d1%8b%d1%85.pdf
3. DIRECTUM 5.6 Руководство администратора [Электронный ресурс] [2018] – Режим доступа: http://download.directum.ru/110996ghj42wdfgt6489/DIRECTUM%205%206%20%20%D0%A0%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE%20%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0.pdf
4. DIRECTUM 5.6 Физическая структура данных [Электронный ресурс] [2018] – Режим доступа: http://download.directum.ru/110996ghj42wdfgt6489/DIRECTUM%205%206%20%20%D0%A0%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.pdf

ПРИЛОЖЕНИЯ

Приложение 1

Текст SQL-запроса

|  |
| --- |
| with t as  (select sr.Analit,  isnull(sr.NameAn, 'NoStandart'),  COUNT(wf.TaskID) TaskSum  from SBWorkflowProcessing (nolock) wf  left join SBTask (nolock) t on t.XRecID = wf.TaskID  left join MBAnalitSpr sr on t.StandardRoute = sr.Analit  group by sr.Analit, sr.NameAn)  select \* from t order by TaskSum |

Приложение 2

Текст программы формирования и запуска SQL-запроса, и записи результатов в файл

|  |
| --- |
| Set objConnection = Server.CreateObject("ADODB.Connection")  objConnection.Open "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;User ID=Domain\UserForAccessToSQL;Initial Catalog=DIRECTUM;Data Source=sql-server"  Set objRS = objConnection.Execute("with t as (select sr.Analit idtm, isnull(sr.NameAn, 'NoStandart') TMname, COUNT(wf.TaskID) TaskSum from SBWorkflowProcessing (nolock) wf left join SBTask (nolock) t on t.XRecID = wf.TaskID left join MBAnalitSpr sr on t.StandardRoute = sr.Analit group by sr.Analit, sr.NameAn) select \* from t order by TaskSum desc")  set FSO = CreateObject("Scripting.FileSystemObject")  set textstream = FSO.OpenTextFile("queue.txt", 8, true)  textstream.write date & " " & time  If Not objRS.EOF Then  Do While Not objRS.EOF  textstream.write vbTab & objRS.Fields(1) & vbTab & objRS.Fields(2)  objRS.MoveNext  Loop  textstream.writeline  End If  Set objRS = Nothing  objConnection.Close  Set objConnection = Nothing  textstream.close |

Приложение 3

Метод проверки полномочий пользователя на использование Инструмента мониторинга

|  |
| --- |
| Function CheckAccess(ByVal UserName)  dim objRootDSE  dim strDNSDomain  dim adoCommand  dim adoConnection  dim strBase  dim strAttributes  set objRootDSE = getobject("LDAP://RootDSE")  strDNSDomain = objRootDSE.Get("DefaultNamingContext")  set adoCommand = createobject("ADODB.Command")  set adoConnection = createobject("ADODB.Connection")  adoConnection.Provider = "ADsDSOObject"  adoConnection.Open = "Active Directory Provider"  adoCommand.ActiveConnection = adoConnection  adoCommand.Properties("Page Size") = 100  adoCommand.Properties("Timeout") = 30  strBase = "<LDAP://" & strDNSDomain & ">"  strAttributes = "member"  dim strFilter  dim strQuery  dim adoRecordset  dim strDN  dim intCount  dim objField  strFilter = "(&(ObjectCategory=group)(ObjectClass=group)(sAMAccountName=group-monitor))"  strQuery = strBase & ";" & strFilter & ";" & strAttributes & ";range=0-\*;subtree"  adoCommand.CommandText = strQuery  set adoRecordset = adoCommand.Execute  intCount = 0  do Until adoRecordset.EOF  for each objField In adoRecordset.Fields  if (vartype(objField) = (vbArray + vbVariant)) \_  then  for each strDN In objField.Value  Set user = GetObject("LDAP://" & strDN)  if "OAO\" & Trim(UCase(user.sAMAccountName)) = Trim(UCase(UserName)) then  intCount = intCount + 1  exit  end if  next  end if  next  adoRecordset.MoveNext  loop  adoRecordset.close  set adoCommand = nothing  set adoConnection = nothing  CheckAccess = intCount  end function |

Приложение 4

Клиентская часть Инструмента мониторинга

|  |
| --- |
| <script>  var I;  var graph2d;  var dataset;  var options;  //----------------------------------------------------------------  function Update() {  clearInterval( I );  $.post( "Table.asp?rand=" + Math.random(), {}, function( data3 ) {  $( "#content" ).html( data3 );  $( ".process" ).remove();  $( "#tg" ).append( "<tr><td style='text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.77); background-color: #007E33;color: #fff;'>Итого без ТМ:</td><td id='sum\_wo\_tm' style='text-align:right;text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.77); background-color: #007E33;color: #fff;'><div></div></td></tr>" );  $( "#tg" ).append( "<tr><td style='text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.77); background-color: #007E33;color: #fff;'>Итого:</td><td id='sum' style='text-align:right;text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.77); background-color: #007E33;color: #fff;'></td></tr>" );  var out = 0;  $( ".wf:not(#sum)" ).each( function() {  tm = $( this ).prev().html();  if (tm != "Мониторинг согласования распорядительного документа") {  out += parseInt( $( this ).html() ); }  } );  $( "#sum" ).html( out );  var out\_wo\_tm = 0;  $( ".wf:not(#sum)" ).each( function() {  tm = $( this ).prev().html();  if ( ( tm !== "Универсальный ТМ по работе с НД" ) && ( tm !== "Актуализация операционной инструкции" ) && (tm != "Мониторинг согласования распорядительного документа")) {  out\_wo\_tm += parseInt( $( this ).html() );  }  } );  $( "#sum\_wo\_tm" ).html( out\_wo\_tm );  if ( out\_wo\_tm >= 200 ) $( "#sum\_wo\_tm" ).css( "background-color", "red" );  else $( "#sum\_wo\_tm" ).css( "background-color", "#007E33" );  if ( out >= 300 ) $( "#sum" ).css( "background-color", "orange" );  else $( "#sum" ).css( "background-color", "#007E33" );  var now = vis.moment();  try {  var obj = {};  obj.x = now;  obj.y = parseInt( out );  dataset.add( obj );  } catch ( e )  } );  c = true;  I = setInterval( Update, 20000 );  }  //----------------------------------------------------------------  $( function() {Update();} );  </script> |