одно значение, но это далеко не так, например, исторический тезаурус чувашского языка çу распознает минимум 5 значений слова-формы, и поэтому текущая технология поиска означает, что пользователь должен часто просеивать результаты, которые включают все эти возможности. Хотя есть возможность фильтрации на основе других слов в контексте поиска. По-прежнему существует большая потребность в ручной проверке и фильтрации, а поисковому устройству препятствует их частая невозможность указать грамматическую часть речи слова, означающую, что все 5 значений «набора» по всем частям речи могут быть возвращены при поиске.

Поскольку поиск является все более важной частью эффективного использования информации, содержащейся в быстро растущих наборах текстовых данных, таких как оцифрованные книги, интернет-новости и контент в социальных сетях, эта проблема словоформ и многозначности наносит вред бизнесу и широкой общественности, а также научным кругам. Анализ текстовых данных позволяет компаниям усовершенствовать свои бизнес-стратегии, пользователям интернета, чтобы быстрее находить требуемую им информацию, а исследователям - выявлять шаблоны в больших наборах данных. Таким образом, вместо использования этих словообразований ключевым фактором в решении проблемы поиска является разработка возможности запуска семантических поисков, в которых «значения» слов являются первичными, а не их прокси-словом.

С этой целью проект «Национальный корпус чувашского языка» направлен на использование всеобъемлющей лексической и семантической базы данных языка. Эта система использует тезаурус на основе базы словаря чувашского языка Н. И. Ашмарина, который в настоящее время содержит более 31 000 чувашских слов, переводов на русский язык, примеров использования в художественной литературе и в повседневной речи, синонимов и лексико-семантических помет.

Этап семантического анализа ориентирован на смысловую интерпретацию текста, которая имеет направленность на внутреннее представление объекта, в то время как лексический и синтаксический анализ ориентирован на внешние конструкции языка.

Для поисковой системы с использованием семантического анализа выделяются несколько этапов:

1. Запрос пользователя.
2. Запрос системой с помощью диалоговых форм уточняющих моментов у пользователя.
3. Преобразование системой запроса пользователя на язык поисковой системы.
4. Поиск с учетом семантического анализа.
5. Поиск синонимов и связанных слов.

Семантический анализ не имеет формальных методов и средств, то есть разрабатывается с помощью уникальных методов с учетом частных решений и рекомендаций. Таким образом, семантический анализатор текста скорее можно отнести к области искусственного интеллекта.

Таланова М.Б., Романова А.А.

**Разработка информационной системы по распределению дополнительных работ
на сельскохозяйственном производстве**

*Нижегородский государственный инженерно-экономический университет
(Россия, Княгинино)*

doi:10.18411/lj-05-2018-81

idsp: 000001:lj-05-2018-81

Аннотация

В данной статье описана информационная система документооборота для повышения качества работы сельскохозяйственного предприятия. Информационная система распределяет дополнительные работы на сельскохозяйственном производстве и отслеживает ход их выполнения.

Ключевые слова: база данных, документооборот, информационная система, клиент-серверное приложение, сельскохозяйственное предприятие

Сельское хозяйство важная отрасль экономики страны, которая обеспечивает человека продуктами питания и сырьем для изготовления предметов потребления. Сельскохозяйственные предприятия для повышения качества своей деятельности постоянно развиваются, что требует более эффективных методов организации и управления деятельностью предприятий с использованием современных технологий.

Объектом рассмотрения является информационная система электронного документооборота по распределению дополнительных работ на сельскохозяйственном производстве и отслеживание хода их выполнения.

Электронный документ – это любая текстовая, звуковая, графическая и другая неструктурированная информация, созданная с помощью средств компьютерной обработки информации и сохраненная на машинном носителе [1, с. 4].

Основная масса документов сельскохозяйственных предприятий выполняется и хранится на бумажных носителях. Процесс заполнения документов требует значительных затрат времени.

Любое предприятие осуществляет документооборот, по которому можно судить о его деятельности. Не везде ведется электронный документооборот, на многих сельскохозяйственных предприятиях всю информацию хранят и заполняют по старинке в бумажном виде. Это влияет на оперативность получения необходимой информации.

На сегодняшний день есть эффективные разработки для ведения документооборота, такие как «ЕФРАТ-Документооборот», СЭД Docsvision, FossDoc. У перечисленных систем есть существенный недостаток – стоимость. Не каждое предприятие может приобрести данный продукт.

Представленная система была разработана в ГБОУ ВО НГИЭУ по заказу ОАО «Суворовское».

В наше время каждое более или менее крупное предприятия имеет достаточно определенный штат сотрудников, которые выполняют те или иные обязанности и получают определенный оклад. Время от времени возникает потребность в выполнении некоторой дополнительной работы, не входящей в круг основных должностных обязанностей сотрудников, в частности сезонной. Для наведения порядка в этой сфере деятельности предприятия были проклассифицированы все виды дополнительных работ, определены суммы оплаты по факту их выполнения. При изучении документов в организации ОАО «Суворовское» была проведена работа по созданию программы для распределения и учета видов работ сотрудников предприятия. Для этого была собрана вся необходимая информация о сотрудниках предприятия и обо всех видах работ, в которых они заняты.

На основании полученных данных создана программа с клиент-серверным приложением имеющей под собой основу реляционной модели управления данными (сокращенно называемой СУРБД) [2,3].

Информационная система состоит из следующих таблиц: «Работники», «Виды работ», «Отделы», «Работы».

При запуске клиента появляется стартовое окно программы (рис. 1). Данные в нем не отображаются, так как нет подключения к серверу. Для подключения вводится IP адрес сервера, и выполняется команда «Соединиться».

После соединения с сервером в таблицах появятся данные, полученные от сервера:

Рисунок 1 – Стартовое окно

В каждой из вкладок, внизу окна имеются интерфейсы для редактирования записей в базе данных. Некоторые данные вводятся, а некоторые выбираются из выпадающего списка.

В случае таблицы «Работы» – дату начала и дату окончания можно выбрать любую. А вот сотрудник и тип работы выбираются из выпадающего списка (берутся из других таблиц), так как таблица «Сотрудники» содержит только идентификаторы записей сотрудника и типа работы (рис. 2).

При нажатии на какую-либо запись в отображающемся списке – данные из неё автоматически занесутся в форму для редактирования.

После этого можно:

- Удалить запись, нажав на кнопку «Удалить».
- Изменить запись, отредактировав данные и нажав «Изменить»
- Добавить новую запись. Даже если не поменять данные – добавится точно такая же запись.

Рисунок 2 – Выбор данных для редактирования таблицы «Работы»

Аналогичным образом происходит редактирование остальных таблиц. Например редактирование записей таблицы «Виды работ» (рис. 3).

Рисунок 3 – Редактирование записей таблицы «Виды работ»

В работе было реализовано клиент серверное приложение для работы с базой данных состоящее из двух отдельных модулей, которые могут находиться как на одной машине, так и на разных.

Клиент был реализован на языке C++, в среде разработки Borland C++ Builder. Эта среда выбрана в связи с удобной работой с визуальным интерфейсом. Для взаимодействия со вторых модулем был написан класс `command_sender`, который хранит в себе функции для отправки команд на сервер, а так же получения с него нужных данных. Клиент не содержит в себе практически никакой работы с данными. При необходимости получить какие-либо данные он отправляет нужную команду серверу и получает данные в уже обработанном виде. Это позволяет не задумываться о нагрузке на клиентские станции.

Сервер реализован на языке C++, в среде разработки Visual Studio. Является консольным приложением и работает в 2 потока.

Первый поток занимается подключением клиентов. Для этого реализованы классы `client` и `clients`. Данный поток постоянно ждет подключения новых пользователей и при наличии нового подключения заносит его в класс `clients`. Одновременно может поддерживаться до 1000 клиентов. Для ожидания подключений используется слушающий сокет. При появлении нового клиента ему передается в пользование свободный сокет, а поток возвращается в режим ожидания подключений на слушающий сокет.

Второй поток постоянно ждет от подключенных пользователей команды. При получении команды от одного из пользователей, он выполняет ее и возвращается к ожиданию новых команд.

Для выполнения команд реализован класс `common`, который содержит в себе функции для получения исходных данных с клиента и передачи готовых данных. Напрямую с базой данных он не работает.

Для работы с базой данных используется класс `db_interface`. Он содержит в себе функции для подключения к базе данных, а так же для выполнения запросов и обработки полученных данных.

Разработанная информационная система позволяет эффективно хранить документацию, затрачивая наименьшее время на ее оформление. Достоинства разработанной информационной системы:

- минимальные затраты времени на оформление документации;
- классификация всех видов дополнительных работ;
- расширяемость системы (возможность ее доработки в случае повышения требований к информационной системе);
- удобный пользовательский интерфейс.

Разработанное приложение может использоваться на предприятии, упрощая работу отдела кадров по распределению дополнительных работ.

1. Хасбиева А.Д., Шмидт А.Б. Документооборот в дипломном проектировании: методические рекомендации. СПб.: Санкт-Петербургский государственный архитектурно-строительный университет, 2013. – 40 с.
2. Голицына, О.Л. Базы данных: учеб. Пособие./ О.Л. Голицына, Н.В. Максимов, И.И. Попов – М.: Форум: ИНФРА-М, 2007. – 400 с.
3. Избачков, В. Информационные системы 3-е издание./ В. Избачков, А. Петров, И. Васильев. – СПб.: Питер, 2010. – 544 с.

Харченко В.Ю., Иконников П.Е., Канавалов И.А.

Математическая модель твердения бетона в условия тепловой обработки

*Тамбовский государственный технический университет
(Россия, Тамбов)*

doi:10.18411/lj-05-2018-82

idsp: 000001:lj-05-2018-82

Сейчас, при бурно развивающемся строительстве, очень высок спрос на стройматериалы, которые удовлетворяют прочностным характеристикам при этом имеют не высокую рыночную стоимость. Поэтому основной является задача производства ЖБИ заданного качества с низкой себестоимостью. Достижение этих характеристик невозможно без глубоких исследований в области математической модели и управления процессом производства ЖБИ. Основными свойствами, характеризующими бетон в строительстве, являются прочность, долговечность и морозостойкость. Прочность бетона во многом зависит от структуры цементного камня, формирующегося в процессе твердения.

Скорость нарастания структурной прочности цементного камня, как и скорость любой химической реакции, может быть резко увеличена с повышением температуры среды. Поэтому для ускорения твердения бетона в производстве строительных материалов применяют тепловую обработку в специальных установках, которая значительно увеличивает оборачиваемость форм и увеличивает выпуск готовых бетонных и железобетонных изделий.

Одним из основных качественных показателей при решении задач моделирования процессом тепловлажностной обработки (ТВО) железобетонных изделий (ЖБИ) являются данные о влиянии основных факторов на формирование прочностных свойств бетона.

Свойства бетона зависят более чем от 50 факторов, о чем свидетельствуют результаты специальных исследований и производственный опыт, которые влияют на прочностные характеристики при сжатии. Создание математической модели с учетом всех прочностных свойств бетона не предоставляется возможным, было решено объединить их в отдельные группы, характеризующимися общими признаками и изучить влияние этих групп на прочность бетона, независимыми от других факторов.