

УДК 004.652:004.85



О.А. Мазурова, М.А. Семикин, М.С. Широкопетлева
 ХНУРЭ, г. Харьков, Украина, oksana.mazurova@nure.ua
 ХНУРЭ, г. Харьков, Украина, maksym.semikin@gmail.com
 ХНУРЭ, г. Харьков, Украина, marija.shirokopetleva@nure.ua

ОБУЧАЮЩАЯ СИСТЕМА ПОДДЕРЖКИ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

Работа посвящена разработке математической модели и обучающего веб-приложения для поддержки процесса проектирования баз данных. Предложено математическое описание процесса проектирования базы данных, которое моделирует взаимосвязь этапов анализа предметной области, концептуального, ER- и логического моделирования баз данных. Для моделирования связи между основными составляющими рассмотренных моделей базы данных использованы базовые понятия сущности и атрибута базы данных. На основании математического описания предложен алгоритм и разработана обучающая система поддержки проектирования баз данных.

БАЗА ДАННЫХ, КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ, ER-МОДЕЛИРОВАНИЕ, ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ, СУЩНОСТЬ, АТТРИБУТ, ОБУЧАЮЩАЯ СИСТЕМА

Введение

Одним из направлений повышения эффективности обучения на сегодня является разработка и использование компьютерных обучающих систем (особенно, обучающих веб-приложений), возможности которых позволяют персонализировать обучение и повысить к нему интерес.

Дисциплина «Базы данных» является одной из нормативных дисциплин цикла профессиональной подготовки бакалавров в области компьютерных наук и программной инженерии. Вопросы проектирования баз данных (БД) являются основой данной дисциплины и требуют не только теоретического, но и практического их освоения во время практических, лабораторных работ и курсового проектирования. Особое внимание при обучении вопросам проектирования БД необходимо уделить этапу концептуального моделирования (КМ) БД, который является достаточно творческим и существенно зависит от знаний и опыта разработчика.

Современные CASE-средства проектирования БД, такие как Vignale Paradigm, ERVisio, ERWin и другие, не поддерживают этап КМ, который традиционно относится к бумажной стадии проектирования БД. Такие профессиональные средства не могут эффективно использоваться на этапе обучения вопросам проектирования БД. Таким образом, существует потребность в разработке обучающего веб-приложения, которое поможет поддержать процесс обучения студентов всем этапам проектирования БД.

Концептуальное моделирование БД предполагает структурированное описание различных аспектов будущей БД и разрабатываемой информационной системы (ИС) [1, 2]. КМ базируется на результатах анализа предметной области (ПО) разработки, который чаще всего включает анализ разнообразной входной документации. Автоматизированная поддержка анализа такой документации [3] позволит выявить ключевые слова,

взаимосвязи и закономерности, которые могут быть учтены во время КМ и использованы для поддержки обучения. Для выявления ключевых слов целесообразно использовать подход на базе соединения статистического и лингвистического анализа текста [3], которые позволят учесть встречаемость слов в текстах, а также отношения между словами.

Современные обучающие системы, такие как Coursera, Edx, Udacity, TrainingWare, Moodle, охватывают различные аспекты обучения в привычной «школьной» парадигме «материал-упражнения» и ориентированы на широкую целевую аудиторию. Однако, актуальной является разработка систем, обучающих с учетом задачно-целевого или ситуативно-проблемного направлений. Например, обучающая система может быть построена для решения с ее помощью практической задачи проектирования БД в рамках курсового проектирования. В этом случае студенты вовлекаются в творческий процесс, в ходе которого, не только закрепляют полученную во время изучения дисциплины информацию, но и превращают ее в организованное знание.

1. Постановка задачи

В работе была поставлена задача разработать обучающее веб-приложение, которое позволит поддержать студента на основных этапах проектирования БД в рамках выполнения курсового проекта по дисциплине «Базы данных».

Для достижения поставленной цели необходимо:

- разработать математическую модель для описания процесса проектирования БД, которая моделирует связь основных этапов проектирования;
- разработать алгоритм использования модели для поддержки процесса проектирования БД;
- разработать обучающую систему, которая позволит поддержать начинающего разработчика БД на основных этапах проектирования БД.

2. Математическое описание процесса проектирования базы данных

Процесс проектирования БД можно представить как множество $DB = \{AN, CM, ER, LM, FM\}$ этапов анализа ПО AN . концептуального CM , ER (инфологического) ER , логического LM и физического FM моделирования.

Анализ ПО AN проводится на основании результатов интервьюирования представителей заказчика и анализа входной документации Doc , к которой следует отнести спецификации требований, описания бизнес-процессов, корпоративные или отраслевые стандарты, и т.п.. Документация в ПО может быть представлена в виде $Doc = \{Doc_k(W, Wd)\}$, где W – набор слов, входящих в документ, Wd – зависимости или связи между словами.

Процесс КМ заключается в выделении концептуальных (понятийных) составляющих ПО создания БД: сущностей, их атрибутов, связей между ними, закономерностей и т.п. [1, 2]. Все составляющие КМ могут быть промоделированы с учетом базовых в области БД понятий сущности и атрибута:

- $A = \{a_{ij}\}$ – множество атрибутов сущностей;
- $E = \{e_i(\)\}$ ($i = \overline{1, n}$) – множество сущностей $S_i(\)$ ПО, которые характеризуются атрибутами из множества A .

Таким образом, концептуальная модель БД в общем виде может быть описана, как совокупность:

$$CM = \{FS, ID, SN, DM, IC, AR, LR, SR\},$$

где FS – описание функциональной структуры проектируемой системы; ID – описание информационных потребностей пользователей; SN – описание схемы взаимосвязи объектов ПО; DM – описание документооборота в системе; IC – описание ограничения целостности ПО; AR – описание алгоритмических зависимостей в ПО; LR – описание лингвистических зависимостей в ПО; SR – описание требования к ИС в целом.

С учетом основополагающей роли сущностей и атрибутов может быть предложено следующее описание составляющих КМ:

- описание функциональной структуры ИС $FS = \{Act, Pr\}$ может быть проведено в виде use-case диаграммы языка UML, как множество акторов $Act = \{Act_i\}_{i=1, n}$ и их прецедентов $Pr = \{Pr_j(Act_i)\}_{j=1, m; i=1, n}$;

- описание информационных потребностей $ID = \{ID_i\}_{i=1, n}$ пользователей, где $ID_i = \{SSF_i, Stat_i, Auto_i\}$ – информационные потребности актора Act_i , которые традиционно для ИС включают:

- а) $SSF_i = \{Sh_i, St_i, Fr_i\}$ – потребности в поиске $Sh_i = \{Sh_i^j(E, A)\}$, сортировке $St_i = \{St_i^k(E, A)\}$ и фильтрации $Fr_i = \{Fr_i^p(E, A)\}$ сущностей E по атрибутам A ;

- б) $Stat_i = \{Stat_i^k(E, A)\}$ – потребностей в получении статистики на базе атрибутов A сущностей E ;

- в) $Auto_i = \{Auto_m^i(E, A)\}$ – потребности в автоматизации, во время проведения которой используются те или иные атрибуты A сущностей E .

- схема взаимосвязи SN сущностей и атрибутов $SN = \{RE_{kl}\}$, где $RE_{kl} = \langle e_k, e_l, R_i \rangle$ – взаимосвязь между сущностями e_k и e_l , где R_i – тип связи между этими сущностями;

- описание документооборота системы $DM = \{Doc_i(E, A)\}$ может быть представлено как множество документов $Doc_i(E, A)$, которые содержат отчетную информацию на базе значений атрибутов A сущностей E ; поскольку такие документы могли выступать исходной документацией, на которой проводился анализ ПО, то $DM \cap Doc$;

- описание алгоритмических зависимостей $AR = \{AR_i(V, E, A)\}$ задает зависимости между некоторыми вычисляемыми атрибутами V и атрибутами A сущностей E , на основании которых они вычисляются.

В процессе концептуального проектирования БД могут быть сформулированы функциональные и ряд нефункциональных требований, которые в той или иной мере касаются объектов (сущностей) ПО. Таким образом, может быть предложен следующий минимальный набор требований:

$$SR = \{dV, dSh, dSt, dFr, dStat, dAuto, dDM, dAR, dMDB, dTR\},$$

где $dV = \{dV_m(E, A)\}$ – требование наличия функции отображения в ИС данных об основных сущностях E и их атрибутах A ;

- dSh, dSt, dFr – требования наличия функций поиска, сортировки и фильтрации сущностей по их атрибутам, причем $Sh_i \subset dSh, St_i \subset dSt, Fr_i \subset dFr$;

- $dStat$ – требование наличия функции получения статистик на базе атрибутов сущностей ($Stat_i \subset dStat$);

- $dAuto$ – требование наличия задачи автоматизации в ПО ($Avto_i \subset dAvto$);

- dDM – требование наличия функций формирования выходных отчетов ($dDM \subset DM$);

- dAR – требование наличия функции формирования вычисляемых полей на базе алгоритмических зависимостей ($dAR \subset AR$);

- $dMDB$ – требования к СУБД;

- dTR – требования к текстовым редакторам для вывода отчетных документов из DM .

Сущности и атрибуты являются наиболее важными ключевыми понятиями КМ, т.к. служат основой для описания практически всех ее составляющих. Поэтому поддержка на этапе выбора этих ключевых понятий позволит облегчить весь процесс КМ. Для поиска слов, которые можно рекомендовать в качестве сущностей или атрибутов БД, можно использовать статистический подход (метод подсчета TF-индекса) [3, 4] в соединении с методами синтаксического анализа текстов для

моделирования связи между сущностями и атрибутами. Показатель TF (англ. *term frequency* – частота слова) – статистическая мера, которую можно использовать для оценки важности слова в контексте документа $Doc_k(W, Wd)$. Она определяется как отношение числа входжений n_i некоторого слова W_i к общему количеству слов $\sum_k n_k$ документа $tf(W_i, Doc_k) = \frac{n_i}{\sum_k n_k}$.

Таким образом, множество сущностей E может быть выбрано из множества лексем W документа $Doc_k(W, Wd)$ с учетом их принадлежности множеству существительных Wn ($Wn \subset W$, $E \subset Wn$). Итак, лексема $W_i \in W$ может быть рекомендована в качестве сущности $e_j \in E$, в том случае, если она является существительным $e_j \in Wn$ и $tf(W_i, Doc_k) = tf(e_j) \geq tfe_{\min}$, где tfe_{\min} – минимальная частота встречаемости сущностей в текстах (tfe_{\min} может быть получена опытным путем).

Множество атрибутов A также может быть сформировано на базе документа $Doc_k(W, Wd)$ как конечное множество слов, которые принадлежат существительным, а также связаны с сущностями словами-связями из RE , которые являются глаголами Wv ($Wv \subset W$, $RE \subset Wv$). Т.е. лексема $W_i \in W$ может быть рекомендована в качестве атрибута $a_{ij} \in A$ сущности $e_j \in E$, в том случае, если она является существительным $a_{ij} \in Wn$, связана с сущностью e_j словом-связью $RE(e_j, a_{ij})$, который является глаголом $RE(e_j, a_{ij}) \in Wv$, и $tf(W_i, Doc_k) = tf(a_{ij}) \geq tfa_{\min}$, где tfa_{\min} – минимальная частота встречаемости атрибутов в текстах.

Процесс инфологического (ER) моделирования является необязательным этапом проектирования БД, но достаточно наглядным и полезным этапом в ходе обучения этому процессу. Сущности, атрибуты и взаимосвязи между сущностями в ER-модели могут быть описаны на основании соответствующих понятий из ранее построенной концептуальной модели CM . Таким образом, ER-модель БД в общем виде может быть описана, как:

$$ER = \{E', A', SN'\},$$

где $E' = \{e'_i(A')\}$ ($i = \overline{1, m}$) – множество сущностей $e'_i(A')$, которые характеризуются атрибутами из множества A' (причем $E \subset E'$); $A' = \{a'_{ij}\}$ – множество атрибутов сущностей (причем $A \subset A'$); схема взаимосвязи $SN' = \{RE'_{kl}\}$ сущностей, где $RE'_{kl} = \langle e'_k, e'_l, R'_i \rangle$ – взаимосвязь между сущностями e'_k и e'_l , R'_i – тип связи между этими сущностями (причем $SN \subset SN'$).

В ходе логического моделирования реляционной БД отношения, их атрибуты и взаимосвязи между реляционными отношениями могут быть

получены на основании соответствующих понятий из ранее построенной ER-модели ER . При этом новые отношения, атрибуты и связи могут возникнуть в результате проведения нормализации логической модели. В случае, если процессу нормализации подверглась ER-модель, то в ходе логического моделирования новые отношения не появятся.

Логическая модель БД в общем виде может быть описана, как множество:

$$LM = \{T, A'', SN''\},$$

где $T = \{t_i(A'')\}$ ($i = \overline{1, p}$) – множество реляционных отношений $t_i(A'')$, которые определены на атрибутах из множества A'' (причем $E' \subseteq T$); $A'' = \{a''_{ij}\}$ – множество атрибутов ($A' \subseteq A''$); схема взаимосвязи $SN'' = \{RT_{kl}\}$ отношений, где $RT_{kl} = \langle t_k, t_l, R'_i \rangle$ – взаимосвязь между отношениями t_k и t_l , R'_i – тип связи между этими отношениями (причем $SN' \subseteq SN''$).

Физическая FM модель БД может быть построена с учетом особенностей СУБД, выбранной согласно требованиям $dMDB$, на основании модели LM .

3. Описание алгоритма поддержки обучения проектированию БД

На основании приведенного описания DB , которое учитывает основные этапы проектирования БД, может быть предложен следующий алгоритм поддержки обучения проектированию БД.

1 этап: Поддержка этапа анализа ПО на основании входных документов $Doc_k(W, Wd)$. В ходе анализа студенту может быть рекомендовано обратить внимание на некоторые ключевые слова $Wk \subset W$, которые потенциально могут описывать сущности ПО и их атрибуты. Кандидаты в ключевые слова $W_i \in Wk$ выбираются среди слов, которые не являются стоп-словами, т.е. не представляют ценности при данном виде обработки: предлоги, союзы и т.п. [4]. Далее проводится расчет частоты встречаемости $tf(W_i, Doc_k)$ в документе $Doc_k(W, Wd)$ ключевых слов $W_i \in Wk$. С учетом $tf(W_i, Doc_k)$ слово может быть отнесено к:

- предварительному множеству сущностей $E^* = \{W_i\}$, если оно существительное и $tf(W_i, Doc_k) \geq tfe_{\min}$;

- предварительному множеству атрибутов $A^* = \{W_i\}$, если оно существительное и $tfe_{\min} \geq tf(W_i, Doc_k) \geq tfa_{\min}$.

2-й этап: поддержка на этапе КМ заключается в предоставлении дополнительной информации на базе результатов анализа ПО или ранее определенных составляющих КМ:

- во время описания функциональной структуры ИС FS может быть предложен предварительный список сущностей E^* для выбора из него

множества акторов $Act = \{Act_i\}_{i=1,n}$ и построения use-case диаграммы;

– во время построения схема взаимосвязи $SN = \{E, A, RE\}$ для формирования множества сущностей $E = \{e_j\}$ могут быть предложены слова $W_i \in E^*$ из предварительного списка сущностей E^* ($W_i = e_i \in E$); для формирования множества атрибутов $A = \{a_{ij}\}$ могут быть предложены слова из предварительного множества атрибутов $W_j \in A^*$. Для определения принадлежности существительного к множеству атрибутов для предложений из текста исходных документов строятся синтаксические деревья и определяются взаимосвязи $RE(e_i, W_j)$ возможного атрибута W_j с сущностями $e_i \in E$, которые встречаются в соответствующих предложениях. Если выявлено слово-связь $RE(e_i, W_j)$, которое принадлежит множеству глаголов Wv , слово W_j может быть предложено в качестве атрибута сущности $e_i(a_j)$ $W_j = a_j \in A$;

– во время описания документооборота системы $DM = \{Doc_i(E, A)\}$ разработчику может быть предложено множество атрибутов A сущностей E , на основании которых он может сформулировать содержимое выходного документа;

– во время описания алгоритмических зависимостей $AR = \{AR_i(V, E, A)\}$ разработчику может быть предложено множество атрибутов A сущностей E , на основании которых вычисляются вычисляемые атрибуты V ;

– во время описания информационных потребностей $ID = \{ID_i\}_{i=1,n}$ пользователей для каждого пользователя из множества акторов Act_i разработчику может быть предложено множество атрибутов A сущностей E , что позволит описать потребности в поиске $Sh_i = \{Sh_i^j(E, A)\}$, сортировке $St_i = \{St_i^k(E, A)\}$, фильтрации $Fr_i = \{Fr_i^p(E, A)\}$ и получении статистики $Stat_i = \{Stat_i^k(E, A)\}$; потребности в проведении автоматизации $Auto_i = \{Auto_m^i(E, A)\}$;

– на базе описанных выше составляющих КМ может быть предложена помощь в формулировании набора требований к созданию ИС $SR = \{dV, dSh, dSt, dFr, dStat, sAuto, dDM, dAR, dMDB, dTR\}$.

3-й этап: поддержка на этапе построения ER-модели $ER = \{E', A', SN'\}$ заключается в предоставлении дополнительной информации с этапа КМ, а именно: множество сущностей $E' = \{e'_i(A')\}$ может быть сформировано на базе сущностей КМ $E \subset E'$; множество атрибутов сущностей $A' = \{a'_{ij}\}$ – на базе атрибутов КМ $A \subset A'$; схема взаимосвязи $SN' = \{RE'_{kl}\}$ сущностей – на базе схемы взаимосвязи $SN \subset SN'$.

4-й этап: поддержка на этапе построения логической модели $LM = \{T, A'', SN''\}$ заключается

в предоставлении дополнительной информации с этапа ER-моделирования, а именно: множество реляционных отношений $T = \{t_i(A'')\}$ может быть сформировано на базе сущностей ER-модели $E' \subseteq T$; множество атрибутов $A'' = \{a''_{ij}\}$ – на базе атрибутов ER-модели $A' \subseteq A''$; множество связей $SN'' = \{RT_{kl}\}$ между отношениями – на базе взаимосвязей из ER-модели $SN' \subseteq SN''$.

4. Описание обучающего веб-приложения

На основании предложенной мат. модели и алгоритма разработано обучающее веб-приложение, которое обеспечивает поддержку начинающему разработчику на основных этапах проектирования БД, а именно:

– поддерживает работу с исходными документами на этапе анализа ПО разработки;

– поддерживает структуризацию и пошаговое прохождение этапа концептуального моделирования, обеспечивая связь этого этапа с этапом анализа;

– поддерживает прохождение этапа ER-моделирования (без проведения нормализации модели), обеспечивая связь этого этапа с этапом концептуального моделирования;

– поддерживает прохождение этапа логического моделирования (без проведения нормализации), обеспечивая связь этого этапа с этапом ER-моделирования;

– поддерживает формирование проектной документации в виде:

а) описания основных составляющих КМ, которые могут выступить в качестве основы для пояснительной записки по курсовому проектированию по дисциплине «Базы данных»;

б) постановки задачи на разработку ИС в виде функциональных и нефункциональных требований к разрабатываемой информационной системе;

в) схем и диаграмм с этапов, концептуального, ER- и логического моделирования.

Созданное средство решает проблему интерактивного обучения «бумажным» этапам проектирования, а именно анализу и концептуальному моделированию БД. Оно делает работу визуально наглядной и быстрой, что также увеличивает интерес к обучению вопросам проектирования баз данных.

Система представляет собой разделенное веб-приложение и работает во всех современных браузерах на всех платформах.

В качестве серверной части выступает RESTful API, написанное с помощью фреймворка Spring Boot, в качестве СУБД используется MariaDB. В качестве объектного интерфейса к базе данных используется Hibernate. Миграции базы данных реализованы с помощью библиотеки Flyway. В сборке проекта и менеджменте зависимостей серверной части задействован Maven.

Клиентская часть представляет собой одностраничное динамическое веб-приложение, написанное на React JS. Манипуляции клиентским состоянием выполнены по архитектуре Flux, с помощью конкретной его реализации — Redux. Обработка многоэтапных или асинхронных операций на клиентской части выполняются согласно паттерну Saga с использованием библиотеки Redux Saga. Общение между клиентом и сервером осуществляется с помощью JSON AJAX сообщений.

Выводы

В работе было предложено математическое описание процесса проектирования БД как совокупности взаимосвязанных этапов, начиная с традиционно бумажных этапов анализа и концептуального моделирования. Основой для описания составляющих на основных этапах проектирования выступают понятия сущности и атрибута, что позволяет промоделировать некоторые достаточно творческие моменты проектирования БД. На основании предложенного математического описания разработан алгоритм, который обеспечивает информационную и методологическую поддержку обучаемого в процессе проектирования БД. Разработанная модель и алгоритм положены в

основу обучающего веб-приложения, которое может быть использовано для получения практических навыков проектирования БД, в том числе для поддержки выполнения курсового проекта в рамках дисциплины «Базы данных». Разработанное приложение внедрено в учебный процесс для направления подготовки 121- Программная инженерия ХНУРЭ для уровня «бакалавр» в рамках дисциплины «Базы данных».

Список литературы:

1. *Дуго С.М.* Базы данных: проектирование и использование: Учебник. - М.: Финансы и статистика, 2005. - 592 с: ил.;
2. Arash Termehchy Cost-Effective Conceptual Design for Information Extraction / Arash Termehchy, Ali Vakilian, Yodsawalai Chodpathumwan, Marianne Winslett. - ACM Transactions on Database Systems, 2015, Volume 40, p. 12:1-12:39;
3. *Кобзарева Т. Ю.* В поисках синтаксической структуры: автоматический анализ русского предложения с опорой на сегментацию / Т. Ю. Кобзарева — М.: РГГУ, 2015. - 371 с.;
4. *Я. О. Вакуленко.* Застосування методів аналізу текстів для підтримки концептуального моделювання баз даних/ Вакуленко Я. О., Мазурова О. О. — Системи обробки інформації. Збірник наукових праць, 2017, випуск 1 (147), с. 127-13.

Поступила в редколлегию 20.06.2017