

**Объектно-ориентированная модель конструирования распределенных баз данных
в условиях нечетких множеств**

¹ *Швачич Геннадий Григорьевич, доктор технических наук, профессор,
Национальная металлургическая академия Украины, г. Днепропетровск*

² *Холод Елена Григорьевна, кандидат технических наук, доцент,
Днепропетровский университет имени Альфреда Нобеля, г. Днепропетровск*

Аннотация: Рассматривается объектно-ориентированная модель системы, которая обеспечивает решение комплекса задач планирования и отчетности в условиях нечетких множеств на основе симметричных пар криптографических ключей. В отличие от классических реляционных баз данных, модель позволяет эффективно решать комплекс задач автоматических транзакций. К ним относятся: наследование параметров плана прошлого периода в плане нового периода, наследование ранее описанных структур, удаление и добавление в рабочий план нового периода новых структур, подлежащих интерактивному описанию, автоматическую проводку информации для требуемого периода времени и др. Объектно-ориентированная модель системы создана на платформе новейших SET-ориентированных (безъядерных) технологий и распределенных вычислений (MIDAS). Единая стандартизация задач и принятая открытая архитектура позволяют неограниченно развивать систему по мере информационной потребности и выполнять параллельные работы независимыми разработчиками в современных средах Delphi и C++Builder.

Ключевые слова: базы данных, нечеткие множества, криптографические ключи, процессор отношений, серверный модуль, безъядерные технологии, распределенные вычисления, транзакции.

Постановка проблемы исследований. Получившая в последнее время широкое распространение классическая модель построения реляционных баз данных, описывающая жесткие отношения между объектами, не в полной мере удовлетворяет ряду задач, таких, например, как динамические системы планирования, отчетности и ведения документации. Кажущаяся простота описания связей между объектами в простых

(прозрачных) системах трудно реализуема в сложных системах, требующих рекуррентных отношений.

Разработка единого информационного стандарта планирования учебного процесса в ВУЗах является актуальной и весьма трудоёмкой задачей. Это связано с часто меняющимися требованиями к ведению учебного процесса, наличием определенных традиций в ВУЗах, что приводит

к работе с неопределенной информацией, т.е. от разработчика требуется решение задач в условиях нечетких множеств [1]. Переход на кредитно-модульную форму обучения в определенной степени способствует стандартизации информации, особенно в части единства учебных планов и программ. Тем не менее, любая система должна обеспечивать достаточную гибкость, учет специфики ВУЗа, особенно в отношении существующих традиций и школ.

Анализ существующих систем и последних исследований в данной области. Примерами удачных решений рассматриваемого класса задач являются динамические индексируемые системы, обладающие большим жизненным циклом. Например, “1С: Бухгалтерия” и “1С: Предприятие”, а также менее распространенная программа “Project”. При этом приложения “1С” функционируют по детерминированным правилам, а их адаптация к изменяемым условиям выполняется разработчиком на основании изменений в нормативных документах. Также следует заметить, что классификаторы отношений указанных приложений заведомо известны.

С целью гибкого доступа к информации в базах применяется язык структурированных запросов SQL. Язык SQL позволяет создавать сложные динамические связи между неиндексированными полями таблиц на основе простых SQL-предложений. Однако, система, построенная на основе SQL-сервера, обладает высокой эксплуатационной стоимостью, что для ВУЗа зачастую является проблематичным. Кроме того, не всякий сотрудник, например учебного отдела, в со-

стоянии оперативно сконструировать SQL-запрос.

Таким образом, хотя язык SQL и обладает уникальной гибкостью, его интерактивное применение (скрытое допускается), скорее всего, неприемлемо.

Цели и задачи исследований. Проведенный анализ показал, что только математически обоснованная модель автоматической классификации информации позволяет достоверно описать информационный реестр, определить единый стандарт (системный интерфейс) для всех решаемых в системе задач и в значительной степени сократить время и ресурсы, необходимые для разработки системы в целом.

Отказ от традиционной реляционной модели построения баз данных и замена ее параллельным пакетом набора данных, поддерживаемых собственным процессором отношений, позволяет эффективно решать задачи с глубокими рекуррентными связями, как в монопольном режиме эксплуатации, так и в условиях распределенных систем.

Основу системы планирования определяет базовая структура справочной информации, которая является ядром информационно справочной системы (ИСС). Базовая структура определяет отношения (административное строение) Факультет → Кафедра, Факультет → Специальность, как отношения один ко многим и, независимо от факультетов, кафедр и специальностей, описывает дисциплины, преподаваемые в ВУЗе (рис. 1).

Ядро ИСС определяет заголовки, являющиеся общими для всех задач, решаемых в системе планирования и отчетности (рис. 2).



Рис. 1. Структурная схема ядра ИСС

Факультет	Кафедра		Специальность		Дисциплина
Код факультета	Код факультета	Код кафедры	Код факультета	Код специальности	Код дисциплины

Рис. 2. Структурная схема заголовков задач

Каждая из решаемых задач подключается к ядру параллельно и использует либо все поля заголовка, либо только те поля, которые определяются ее требованиями (рис. 3).

Из приведенной схемы видно, что она обеспечивает как отношение *Задача → Справочник*, так и отношение *Задача → Задача*. Например, задача формирования учебного плана опирается

не только на справочники, но и на рабочий план дисциплины, а также может использовать информацию о количестве студенческих групп и количестве студентов, обучающихся по данной специальности. На схеме не отражены константы планирования учебного процесса, так как они не формируют криптографические ключи.

В процессе разработки ИСС должны быть реализованы следующие основные задачи: синтез

криптографического ключа из содержательной информации, реализация основного свойства первичного ключа, поддержка целостности данных системы, обеспечение невозможности модификации данных (содержательной информации) и установленных правил создания справочников, разработка дружественного пользовательского интерфейса, получение твердых копий справочников.

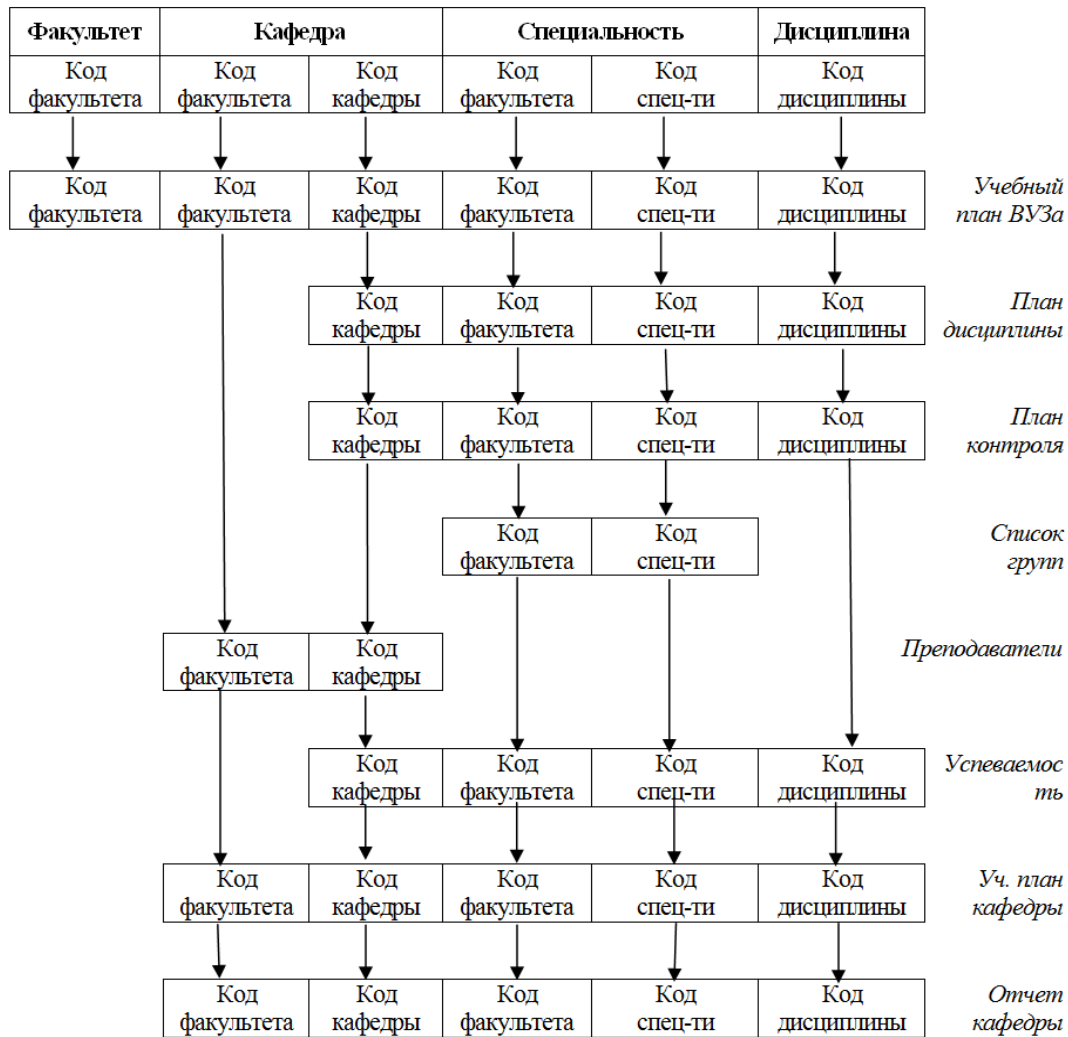


Рис. 3. Схема подключения задач к ядру ИСС

Изложение основного материала исследований

Выбор основных компонентов и структурирование данных. Описанные объекты реализуются с использованием компонентов удаленного доступа TClientDataSet как доменные структуры, позволяющие получать доступ к значениям атрибутов посредством имен доменов. Отношения между полями таблиц, описывающих объекты, реализуются непосредственно в процессе дизайна приложения посредством установки “жестких” связей по схеме Master – Detail отношений. Связь объекта TClientDataSet

с элементами управления осуществляется посредством связующего компонента TDataSource.

Криптография как средство идентификации информации. Все существующие стандартные и вновь определяемые классы или библиотеки классов регистрируются в файле реестра с помощью специальных ключей, обеспечивающих их однозначную идентификацию. Первый метод получения кода класса – генерирование последовательности случайных чисел по одному из рекуррентных алгоритмов (например, $g_i = ag_{i-1} + b(\text{mod } m)$, где g_i – i -й член последовательности псевдослучайных чисел, a, b ,

m и g_0 – ключевые параметры). Второй метод представляет собой криптографическую 32 или 64 разрядную свертку (XЭШ функцию), сжимающую содержательную часть текста – интерфейс модуля класса. Например, в ОС UNIX реализована функция Peter J. Weinberger, обладающая повышенной чувствительностью к равнозначным информационным блокам. Этот факт обеспечивает регистрацию класса только в случае его отсутствия. Важнейшим вопросом является проверка целостности данных в файлах (контрольная сумма), которая осуществляется функцией CRC32 по алгоритму:

METALLURGICAL	2049846875		HIGHER MATHEMATICS	1799696223
ELECTRO METALLURGICAL	1089707116	или	APPLIED MATHEMATICS	1611101285

Эту же функцию эффективно применять и для создания специальных ключей проверки возможной модификации информации и описания структурных связей.

Для идентификации динамических образований (лекционный поток, состав преподавателей и т.д.), а также наследования ранее описанных структур в процессе преобразования планов предыдущих периодов в планы новых периодов целесообразно использовать функцию Peter J. Weinberger: $X := \text{HESH}(TA \text{ as String} \parallel TB \text{ as String} \parallel TC \text{ as String} \parallel TD \text{ as String})$, где: TA, TB, TC, TD – значения числовых доменов, а \parallel – знак конкатенации.

Отношения и симметричные пары криптографических ключей. Административная структура ВУЗа описывается парными отношениями (один ко многим) вида: Факультет : Специальности; Факультет : Кафедры; Специальности : Группы; Кафедры : Преподаватели; Группы : Студенты; Дисциплина : Рабочая программа. Аналогичным образом определяются отношения между решаемыми задачами: Рабочая программа : Периоды выполнения; Дисциплина : Преподаватели; Лекционный поток : Группы; Дисциплина : Группа (успеваемость); и т.д.

F_CODE	S_CODE	C_CODE	D_CODE	C_ROW	C_COL	CRCF	NUM
--------	--------	--------	--------	-------	-------	------	-----

Рис. 4. Интерфейс задач

Здесь: F_CODE – код факультета; S_CODE – код специальности; C_CODE – код кафедры; D_CODE – код дисциплины; C_ROW (C_COL) – код свертки значимой информации строки (столбца); CRCF – контрольная сумма кортежа; NUM – код (номер) задачи.

$RC := ((CRC \text{ SHR } 8) \text{ AND } \$FFFFFF) \text{ XOR } CRCTbl[(CRC \text{ XOR } Source[I] \text{ AND } \$FF)]$,

где CRC SHR 8 – значение контрольной суммы, сдвинутой на 8 разрядов; CRCTbl – таблица опорных констант; CRC XOR Source[I] – значение контрольной суммы исключительно блока информации.

Для описания информационного реестра ВУЗа особый интерес представляет функция вычисления контрольной суммы, обеспечивающая однозначный код свертки смысловой информации и неповторяемость значений в пределах 2^{32} , например:

Каждая категория идентифицируется собственным криптографическим ключом и отношения устанавливаются между ключами [2]. Так как криптографический ключ является сверткой смысловой информации, то нет необходимости в разработке специальных классификаторов, что приводит к сокращению объемов работ и сроков разработки системы, а также повышает её надежность в целом.

Информационный реестр системы. Единый стандарт задач. Информационный реестр системы представлен четырьмя ключевыми полями, идентифицирующими отношения между главными объектами административной структуры ВУЗа и ключевым полем справочника преподаваемых дисциплин. Ключевые поля определяют парные отношения вида: *Факультет : Специальность и Факультет : Кафедра*. Ключевое поле категории *Дисциплина* является независимым. Три служебных поля обеспечивают идентификацию динамических образований (криптографических свертки значимой информации столбцов и строк) и контрольную сумму кортежа. Четвертое служебное поле идентифицирует решаемую задачу. Таким образом, восемь полей (рис. 4) определяют заголовок для всех задач системы.

Процессор отношений. Каждая задача изначально представляет собой неупорядоченный набор данных, отношения между которыми реализует процессор отношений. Процессор отношений является внутренним ядром системы и представляет собой набор (матрицу 4x4) экземпляров класса доступа к данным TDataSet. Логи-

ческие отношения между компонентами устанавливаются между справочной информацией и

```
DataSetG.Filter := 'Problem = ' + 'value' + ' and ' + 'F_CODE = ' +
QuotedStr(DataSetAF_CODE.Value) + ' and ' + 'S_CODE = ' +
QuotedStr(DataSetBS_CODE.Value) + 'C_CODE = ' +
QuotedStr(DataSetCC_CODE.Value) + ' and ' + 'D_CODE = ' +
QuotedStr(DataSetDD_CODE.Value);
```

а для справочников:

```
DataSetB.Filter := 'F_CODE = ' + QuotedStr(DataSetAF_CODE.Value);
DataSetC.Filter := 'F_CODE = ' + QuotedStr(DataSetAF_CODE.Value);
```

При этом данные выборок упорядочиваются в их виртуальных отражениях путем комплексной индексации полей. Процессор отношений реализован в виде отдельного DataModule, интерфейс которого содержит набор Public процедур и функций, его архитектура позволяет решать одновременно восемь прямых и обратных задач. С точки зрения пользователя, процессор отношений является набором команд, реализующих конкретные задачи.

Архитектура системы. Каждый пользователь имеет экземпляр системы, размещенной на съёмном носителе, либо прозрачный доступ к экземпляру приложения. Пользователи получают доступ к данным в объеме, определенном лицензией, и могут осуществлять обмен данными между собой и с главной базой данных в пакетном режиме. Главная база данных представляет собой отдельный файл (пакет файлов), располо-

женными решаемых задач. Например, для задачи выборки данных имеем:

женный на выделенном компьютере, либо является расширенным представителем любого экземпляра приложения. Таким образом, принятая архитектура полностью соответствует концепции распределенных систем.

По умолчанию, каждый экземпляр приложения является локальной СУБД. После выполнения команды регистрации класса процессора отношений, приложение приобретает свойство сервера, предоставляющего доступ к своим ресурсам всем клиентам сети.

Опытный образец системы. В соответствии с предлагаемой концепцией был разработан опытный образец системы. Одна из основных реализованных задач – формирование общего плана ВУЗа либо плана кафедры (рис. 5) на основе справочно-нормативной информации (рис. 6), рабочих программ и контингента студентов.

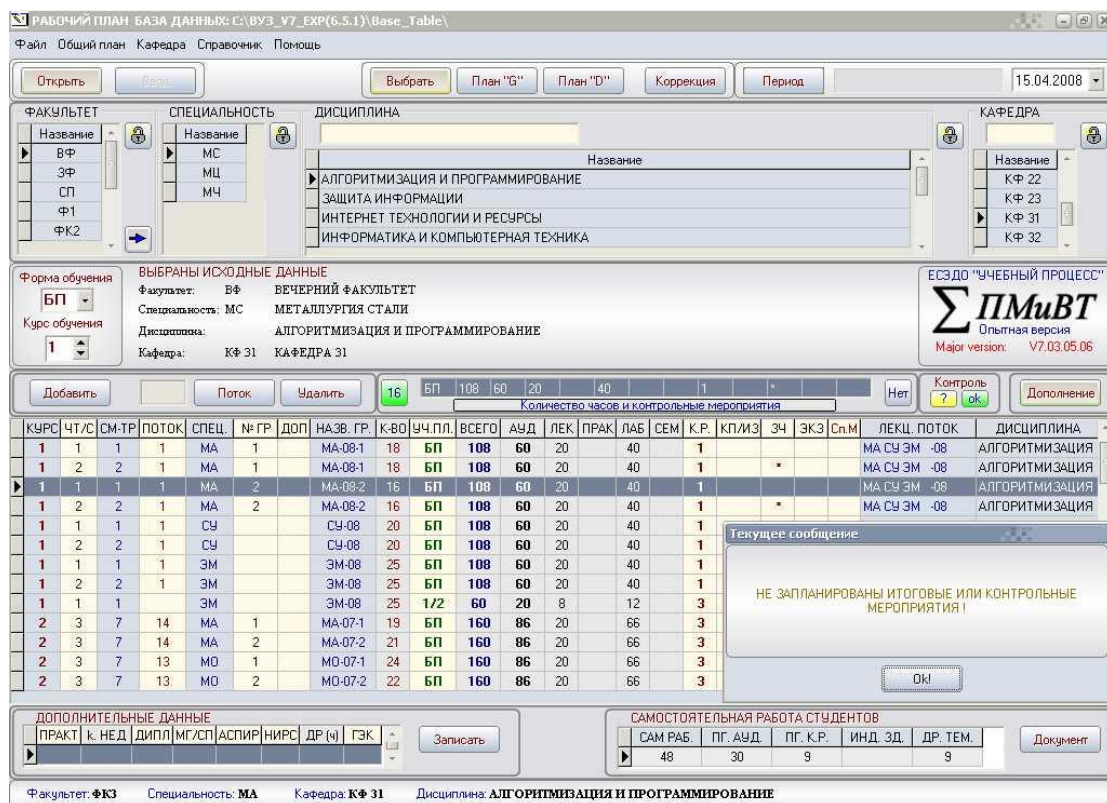


Рис. 5. Главное окно системы

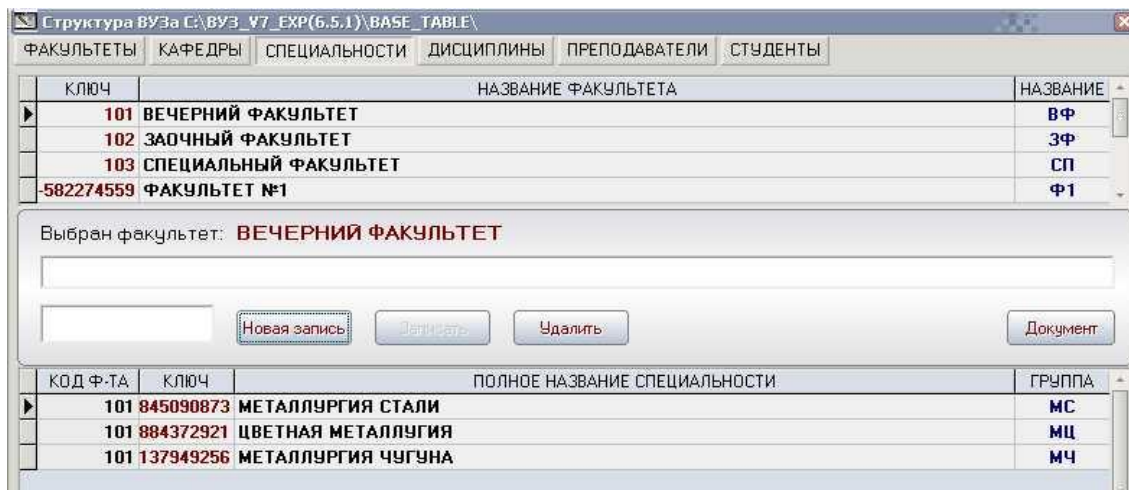


Рис. 6. Окно описания реестра системы

Интерактивный ввод первичного плана поддерживается комплексом мероприятий, облегчающих действия пользователя по формированию записей путем простой подстановки данных, автоматическим формированием атрибутов групп и потоков, а также выводом сообщений, носящих предупредительный и рекомендательный

характер. Расчет рабочих планов кафедр (общего рабочего плана) опирается на пять базовых методик, описание которых сводится к вводу формул в стандартном виде (рис. 7). Селектор выборки данных позволяет осуществлять не только поэтапное представление плана, но и различного рода выборки.

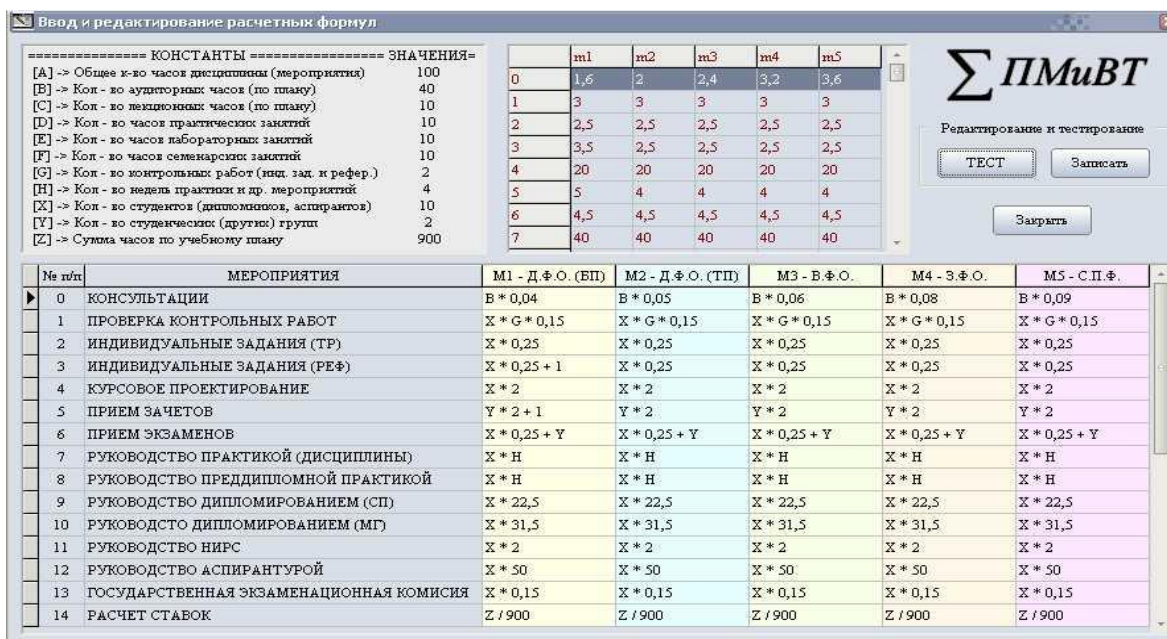


Рис. 7. Окно описания методик

Одной из наиболее сложных задач является планирование работы преподавателя. Её решение реализовано с применением алгоритмов искусственного интеллекта. Суть проблемы сводится не только к одноразовому описанию структуры *Дисциплина* → *Лектор* → *Ассистенты* (условные категории), а и к сохранению описанных структур в последующих планах. Заметим, что все связанные с этим интерактивные операции приведены к виду простых подстановок и команд контекстного меню соответствующих элементов управления системы (рис. 8).

В рамках принятой концепции приложение, реализующее систему планирования (безъядерная СУБД), обеспечивает функционирование системы на съемном носителе (JetFlash) небольшой емкости без каких бы то ни было специальных установок и драйверов поддержки. Этот факт обеспечивает в перспективе весьма эффективную распределенную многопользовательскую систему, позволяющую осуществлять обмен информацией между участниками сети без установки дополнительных элементов системы на компьютерах.

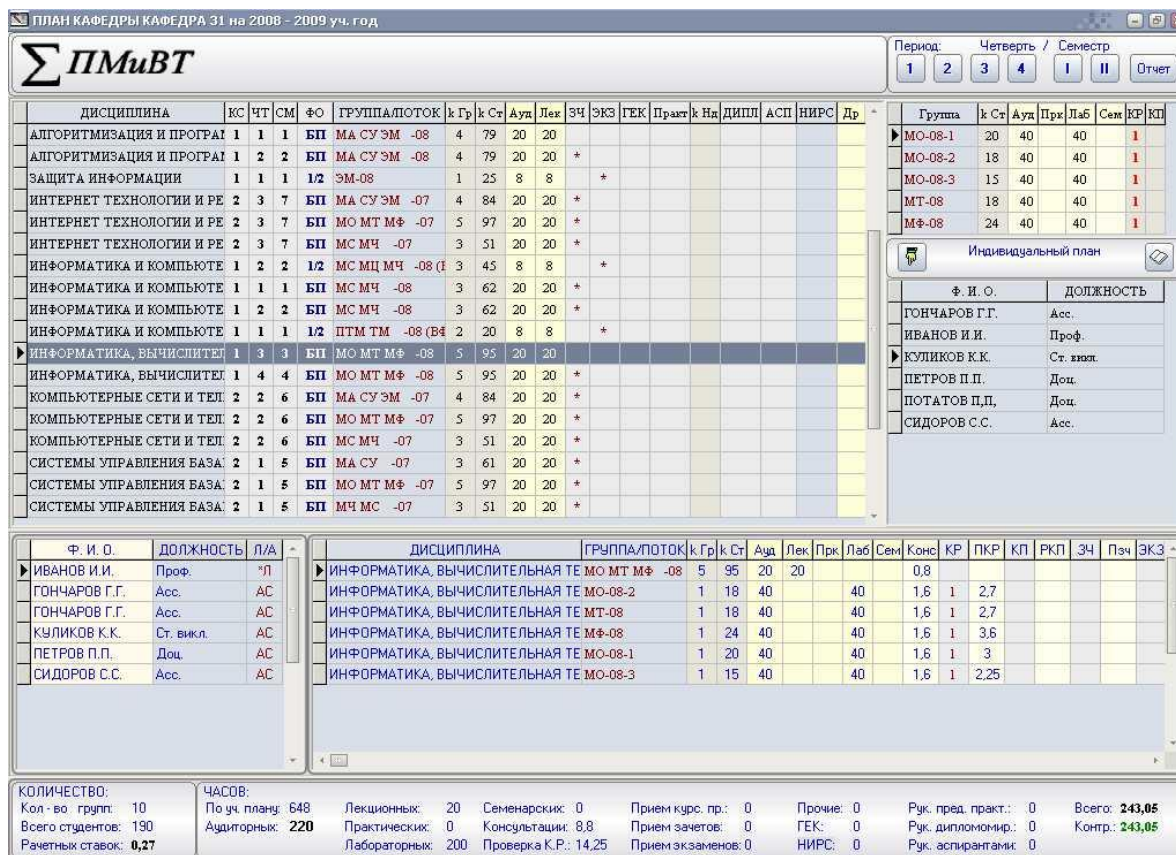


Рис. 8. Контекстное меню элементов управления системы

Целостность данных, надежность и информационная стойкость. Предложенная концепция распределенной безъядерной объектно-ориентированной системы автоматически снимает проблему поддержки индексной целостности, так как ключевые поля и индексы, в их общепринятом понятии, в ней вообще отсутствуют. За целостность отношений отвечает процессор отношений, инкапсулированный в клиентское приложение. Целостность отношений определяется контрольными суммами (свертками) строк и блоков (наборов строк). Вследствие чего такие операции, как транзакции периодов являются не только не опасными, а наоборот весьма полезными. Их рекомендуется выполнять как при изменении периодов планирования, также и после ввода некоторого объема информации. Данные операции выполняют повторное обновление виртуальных связей, удаляют ошибочные записи (возможные при сбоях электропитания) и другие операции, подобные дефрагментации файлов операционной системы, а также сжимают файлы и повторно сохраняют их на диске. В локальной системе защита данных от случайной и умышленной фальсификации достигается параллельным использованием одних и тех же данных, в основном виртуальных (существующих только в период работы приложения), для всех задач. Таким образом, ни какое изменение

данных в промежуточных представлениях невозможно. С другой стороны, изменение данных в исходном (контролируемом) плане немедленно отражается во всех создаваемых документах. Более того, в каждом документе выводятся две контрольных суммы, определяющие целостность данных. **Выводы и перспективы дальнейших исследований.** Предлагаемая объектно-ориентированная модель системы обеспечивает решение комплекса задач планирования и отчетности в условиях нечетких множеств на основе симметричных пар криптографических ключей. Идентификаторы объектов (криптографические ключи), описывающие структуру ВУЗа и процесс обучения, автоматически генерируются на основе полных названий структурных подразделений, специальностей, дисциплин и т.д. Для генерации ключей используется односторонняя ХЭШ функция CRC32 (стандарт NIST). Совокупность криптографических ключей является заголовком решаемой задачи. Процессор отношений описывается макрофункциями и представляет собой серверный модуль, обеспечивающий обработку задач. С точки зрения процессора отношений все задачи являются параллельными. Отсюда следует, что задачи могут подключаться к системе по мере возникновения в их потребности.

Каждая задача описывается линейной информационной таблицей в формате CDS (бинарном представлении формата XML), обеспечивающем компактные файлы безъядерной базы данных.

В отличие от классических реляционных баз данных, предлагаемая модель позволяет эффективно решать комплекс задач автоматических транзакций, таких как: наследование параметров плана прошлого периода в плане нового периода, наследование ранее описанных структур, удаление и добавление в рабочий план нового периода новых структур, подлежащих интерактивному описанию, автоматическую проводку студентов во всем периоде их обучения и др.

Гибкое описание структуры подразделений ВУЗа, учебных планов и программ, специально-

стей, состава студенческих групп обеспечивает одновременное изменение данных во всех задачах, решаемых системой.

Следует отметить, что объектно-ориентированная модель системы, построенная на платформе новейших SET ориентированных (безъядерных) технологий и распределенных вычислений (MIDAS), единая стандартизация задач и принятая открытая архитектура системы открывают возможности к дальнейшему её развитию по мере информационной потребности и выполнению параллельных работ независимыми разработчиками в современных средах Delphi и C++Builder.

Литература

1. Иващенко В.П. Объектно-аналитическая модель распределенных баз данных в условиях нечетких множеств / В.П. Иващенко, Г.Г. Швачич, А.В. Овсянников // Матеріали VI Міжнародної науково-технічної конференції «КОМТЕХБУД 2008». – Київ-Севастополь, 2008. – С. 70-71.
2. Иващенко В.П. О некоторых аспектах идентификации информации при помощи симметричных пар криптографических ключей / В.П. Иващенко, Г.Г. Швачич, А.В. Овсянников // Матеріали VI Міжнародної науково-технічної конференції «КОМТЕХБУД 2008». – Київ-Севастополь, 2008. – С. 105-106.

Shvachych G.G., Kholod E.G.

Object oriented model for distributed database engineering in conditions of fuzzy sets

Abstract: The proposed work considers object oriented model of the system that provides solution for enterprise planning and reporting task complex in conditions of fuzzy sets based on symmetric pairs of cryptographic keys. Object identifiers (cryptographic keys) describing the enterprise structure are generated automatically based on full (standardized) names of structural units. To generate the keys CRC32 one-way hash function (NIST standard) is used making possible creation of 2^{32} no repetitive keys. The population of cryptographic keys is the title of the problem being solved. The problem solution is made by relations processor described with macrofunctions. Relations processor represents a server module providing both single and batch problem processing. From the standpoint of relations processor all tasks are concurrent. Thus the key feature of the developed system lies in the fact that tasks may be connected to the system when they are necessary. In this connection each task is described by CDS linear information table (binary representation of XML format) providing compact files of non-kernel data base. Flexible description of the enterprise structure permits concurrent change of data in all tasks being solved by the system. In contrast to traditional relational data bases the object oriented model makes possible effective solving for a complex of automatic transaction tasks such as inheritance of past period design parameters in the parameters of new period, inheritance of previously described structures, removal and addition of new structures subject to interactive description to the new period detailed design, automatic information posting for the required time period, etc. The system model is powered by the latest SET-oriented (non-kernel) technology and grid computing (MIDAS). Unified task standardization and accepted open architecture make possible unlimited system developing depending on informational needs and performing parallel tasks by independent architects in the state-of-art Delphi and C++Builder environment.

Keywords: data bases, fuzzy sets, cryptographic keys, relations processor, server module, kernel-free technology, grid computing, transactions.