

УДК 004 (575.2) (04)

СТРУКТУРА, АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕГРИРОВАННОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА В ВУЗЕ

Е.З. Гаврилец – канд. техн. наук, доцент,
О.А. Медведева – аспирант,
К.А. Скрынников, С.Д. Туровский – студенты

Problems of structure and algorithms development and their software implementation for computer-aided system, designed for automation of educational process organization at higher educational institutions are considered in the article.

Одним из направлений внедрения информационных технологий является организация образовательных процессов, в частности, в высших учебных заведениях. В статье рассматриваются вопросы разработки структуры, алгоритмического и программного обеспечения системы, предназначенной для автоматизации организации учебного процесса в вузе (рис. 1).

Верхний уровень иерархии указанной структуры занимает учебно-методическое управление (УМУ), на нижнем уровне распо-

ложены факультеты, объединяющие в своем составе кафедры, деканат и диспетчера, ответственного за составление расписания учебных занятий. УМУ осуществляет координацию взаимодействия между кафедрами с учетом существования трех типов кафедр: профилирующие (осуществляющие выпуск специалистов), непрофилирующие и смешанного типа (выпускающие специалистов и обслуживающие другие специальности).

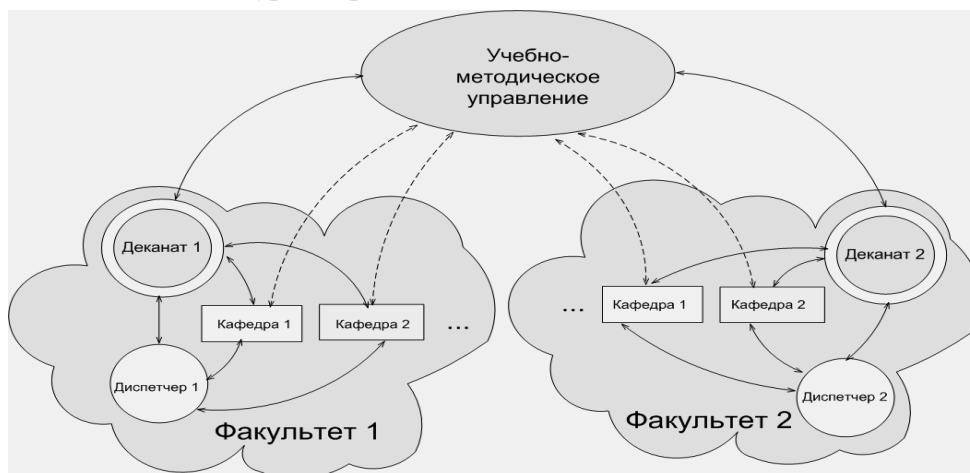


Рис. 1. Структура высшего учебного заведения.

Контекстная диаграмма потоков данных (DFD) интегрированной автоматизированной системы организации учебного процесса в вузе, имеющем указанную структуру (рис. 2), включает интегрированную автоматизированную систему и четыре взаимодействующие с ней внешние сущности: Учебно-методическое управление, кафедру, деканат и диспетчера. На контекстной диаграмме представлены основные потоки данных, которыми обмениваются внешние сущности с автоматизированной системой.

На основании функциональной модели в виде иерархии диаграмм потоков данных и словаря модели получена логическая модель данных интегрированной системы в виде диаграмм “сущность-связь” (ERD) в среде ErWin [1]. Для каждой сущности определены ключевые атрибуты, установлены отношения между сущностями на уровне логической модели.

Логические модели данных использованы при генерации физических моделей данных. Окончательная структура данных сформирована в результате экспорта физических моделей данных в среду MS SQL Server 2000.

Функции интегрированной автоматизированной системы

В соответствии с составом внешних сущностей функциональной модели в системе выделены четыре подсистемы, осуществляющие свои функции.

Функции, реализуемые подсистемой “Учебно-методическое управление”:

- учет данных о профессорско-преподавательском составе (ППС) и студентах университета;
- учет данных о параметрах учебных планов и составе общих блоков планов для всех специальностей университета;
- учет и контроль данных об учебных планах по специальностям;
- учет и контроль данных о плановой учебной нагрузке преподавателей и о фактическом выполнении этой нагрузки;
- формирование отчетной документации УМУ;
- организация обмена данными между учебно-методическим управлением и кафедрами университета.

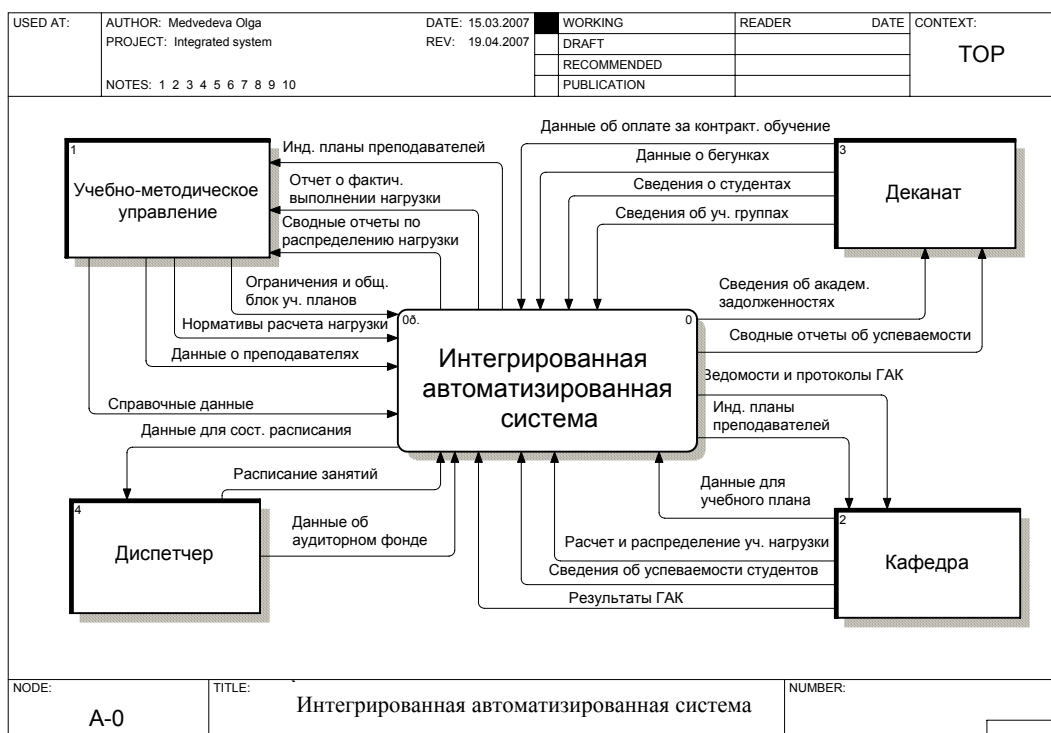


Рис. 2. Контекстная диаграмма потоков данных.

Функции, реализуемые подсистемой “Кафедра”:

- ✓ автоматизация составления учебных планов для выпускающих кафедр в соответствии со стандартами по специальности;
- ✓ автоматизация распределения годовой учебной нагрузки профессорско-преподавательского состава;
- ✓ автоматизация формирования индивидуальных планов преподавателей и сведений, предоставляемых учебно-методическому отделу и диспетчеру факультета;
- ✓ автоматизация учета успеваемости студентов;
- ✓ автоматизация формирования документации для проведения заседаний государственных аттестационных комиссий (ГАК) и итоговых отчетов ГАК.

Функции, реализуемые подсистемой “Деканат”:

- ↔ ввод и корректировка данных об учебных группах;
- ↔ ввод и корректировка данных о студентах;
- ↔ формирование зачетно-экзаменационных ведомостей;
- ↔ учет успеваемости студентов;
- ↔ формирование отчетной документации.

Функции, реализуемые подсистемой “Диспетчер”:

- ввод и корректировка исходных данных (данные о распределении нагрузки, об аудиторном фонде и т.д.);
- автоматизация составления расписания;
- автоматизация формирования итоговых документов.

Алгоритмы функционирования интегрированной автоматизированной системы

Этапы взаимодействия подсистем “Учебно-методическое управление” (УМУ) и “Кафедра”:

Этап 1. Подсистема УМУ формирует и экспортирует профилирующим кафедрам стандарты по специальностям, сведения по общему блоку дисциплин, а также параметры ограничений, в условиях которых должно осуществляться формирование учебных планов.

Этап 2. На профилирующих кафедрах с использованием соответствующей подсистемы разрабатываются учебные планы по специальностям на основе образовательных стандартов и с учетом данных, полученных из подсистемы УМУ на предыдущем этапе.

Этап 3. Подсистема УМУ формирует и экспортирует всем кафедрам нормативы для расчета учебной нагрузки ППС, а также сведения о нагрузке для обслуживающих или смешанного типа кафедр, сформированные при анализе данных учебных планов профилирующих кафедр.

Этап 4. Кафедры с использованием соответствующей подсистемы осуществляют расчет и распределение учебной нагрузки между преподавателями и экспорт этих сведений в подсистему УМУ, где они проверяются на соответствие нормативам.

Этап 5. Подсистема УМУ на основе полученных от кафедр на предыдущем этапе данных формирует сводные документы по расчету и распределению учебной нагрузки по факультетам и в целом по вузу.

Этап 6. Кафедры в конце учебного года с использованием соответствующей подсистемы формируют и осуществляют экспорт в подсистему УМУ данных о фактическом выполнении учебной нагрузки.

Этап 7. Подсистема УМУ формирует сводные отчетные документы о фактическом выполнении нагрузки факультетам и в целом по вузу.

Алгоритм формирования учебного плана

Алгоритм формирования учебного плана на базе образовательного стандарта по специальности [2] состоит из следующих этапов:

1) для каждой из обязательных (установленных стандартом) дисциплин определяется общее количество часов и устанавливается принадлежность дисциплины блоку и компоненту учебного плана;

2) для каждой обязательной дисциплины учебного плана задается распределение часов аудиторной нагрузки (лекции, семинарские и лабораторные занятия), а также формы отчетности (экзамен, зачет, курсовая работа и т.д.) по семестрам;

3) для неустановленных стандартом дисциплин задается принадлежность к блоку и компоненту учебного плана и распределение часов аудиторной нагрузки и форм отчетности по семестрам;

4) проверяется выполнение следующих ограничений:

а. каждая дисциплина должна быть включена только в один компонент одного блока учебного плана;

б. сумма часов дисциплин каждого компонента каждого блока учебного плана не должна превышать количества часов, заданного стандартом;

с. доля аудиторной нагрузки по дисциплине не должна превышать максимально допустимую;

д. недельная аудиторная нагрузка студентов (в часах) не должна превышать максимально допустимую;

е. количество отчетностей каждой формы в семестре не должно превышать максимально допустимое.

Если хотя бы одно из ограничений не выполняется, осуществляется переход к этапу 2. При выполнении ограничений процесс формирования учебного плана завершается.

Алгоритмы распределения годовой учебной нагрузки ППС

В интегрированной автоматизированной системе предусмотрена возможность использования для распределения годовой учебной нагрузки ППС кафедры как оптимизационного, так и неоптимизационного алгоритмов.

Формализация задачи распределения годовой учебной нагрузки ППС кафедры вуза как задачи целочисленного нелинейного программирования приведена в [3], там же предложен вариант сведения этой задачи к серии задач целочисленного линейного программирования и описан оптимизационный алгоритм распределения годовой учебной нагрузки.

Неоптимизационный алгоритм распределения годовой учебной нагрузки ППС кафедры содержит следующие этапы.

Этап 1. Фиксация количества ставок преподавателей.

Этап 2. Для видов учебной нагрузки, зависящих от дисциплин (лекции, семинарские, лабораторные занятия, экзамены, зачеты и т.д.), определяется количество часов (как плановой нагрузки, так и оформленных на условиях почасовой оплаты) в семестр для каждого преподавателя по читаемым им дисциплинам в каждом потоке. В качестве потока рассматриваются как лекционные потоки, так и учебные группы и подгруппы.

Этап 3. Определение количества часов в семестр для каждого преподавателя по видам нагрузки, не зависящим от дисциплин (руководство практиками, руководство аспирантами, заведование кафедрой, участие в ГАК и т.д.).

Этапы 2 и 3 осуществляются последовательно для каждого семестра.

Этап 4. Определенный при выполнении предыдущих этапов вариант распределения учебной нагрузки проверяется на выполнение следующих ограничений:

а. для каждого преподавателя количество часов, оформленных на условиях почасовой оплаты, не должно превышать максимально допустимое;

б. доля в общей годовой нагрузке кафедры часов, оформленных на условиях почасовой оплаты, не должна превышать максимально допустимую;

с. плановая (без учета почасовой) нагрузка преподавателя должна принадлежать интервалу, границы которого определяются должностью и количеством ставок;

д. количество преподавателей, выполняющих один вид нагрузки в одном потоке, не должно превышать максимально допустимое.

Если ограничения выполняются (вариант решения найден), то работа алгоритма распределения учебной нагрузки прекращается. В противном случае последовательность действий необходимо повторить, начиная с этапа 1.

Алгоритмы составления расписания учебных занятий

В интегрированной автоматизированной системе предусмотрена возможность использования для составления расписания учебных занятий как неоптимизационного, так и оптимизационного алгоритмов [4].

Этапы неоптимизационного алгоритма, который может применяться для составления расписания на уровне кафедры и на уровне факультета в целом, следующие:

Этап 1. Ввод данных о распределении учебной нагрузки ППС и об аудиторном фонде.

Этап 2. Ввод данных о расписании занятий, установленных на уровне факультета (занятия иностранным языком, физвоспитание и т.д.).

Этап 3. Ввод данных о разрешенном времени занятий для учебных групп по дням недели.

Этап 4. Ввод данных о допустимом времени занятий для каждого преподавателя по дням недели.

Этап 5. Формирование варианта расписания учебных занятий.

Этап 6. Проверка сформированного на предыдущем этапе варианта расписания на выполнение следующих ограничений:

- а. в одной аудитории в один день недели в одну учебную пару может быть только одно занятие;
- б. в одну учебную пару одного дня недели у преподавателя может быть только одно занятие;
- с. в один день недели в одну учебную пару у подгруппы может быть только одно занятие;
- д. количество пар в день у подгруппы не превышает заданное;
- е. количество пар в день у преподавателя не превышает заданное;
- ф. аудиторная нагрузка по дисциплине должна быть выполнена полностью.

Если хотя бы одно из ограничений не выполняется, осуществляется переход к этапу 3. При выполнении ограничений процесс составления расписания учебных занятий завершается.

Оптимизационный алгоритм базируется на том, что задача составления расписания может быть формализована как задача целочисленного нелинейного программирования, а с учетом ряда предположений сведена к задаче целочисленного линейного программирования (ЦЛП). Однако применение оптимизационного алгоритма для составления расписания учеб-

ных занятий на уровне факультета в целом в силу большой размерности этой задачи представляется нецелесообразным. В то же время размерность задачи составления расписания на уровне кафедры позволяет применять для ее решения оптимизационный алгоритм. Первые четыре этапа этого алгоритма совпадают с соответствующими этапами неоптимизационного алгоритма. На пятом этапе формируются параметры задачи ЦЛП, причем, в качестве критерия минимизации принимается суммарное количество “окон” у преподавателей и студенческих групп. Для решения сформированной задачи ЦЛП могут быть использованы существующие пакеты прикладных программ, в частности, пакет Lp_solve [4]. Если решение задачи ЦЛП не найдено или решение существует, но оно по каким-либо причинам не устраивает соответствующее должностное лицо, то последовательность действий необходимо повторить, начиная с этапа 3. В противном случае работа алгоритма завершается.

Программная реализация

Интегрированная автоматизированная система разработана на базе двухуровневой модели технологии клиент-сервер с реализацией сервера баз данных в MS SQL Server 2000 [5]. На сервере расположена база данных,

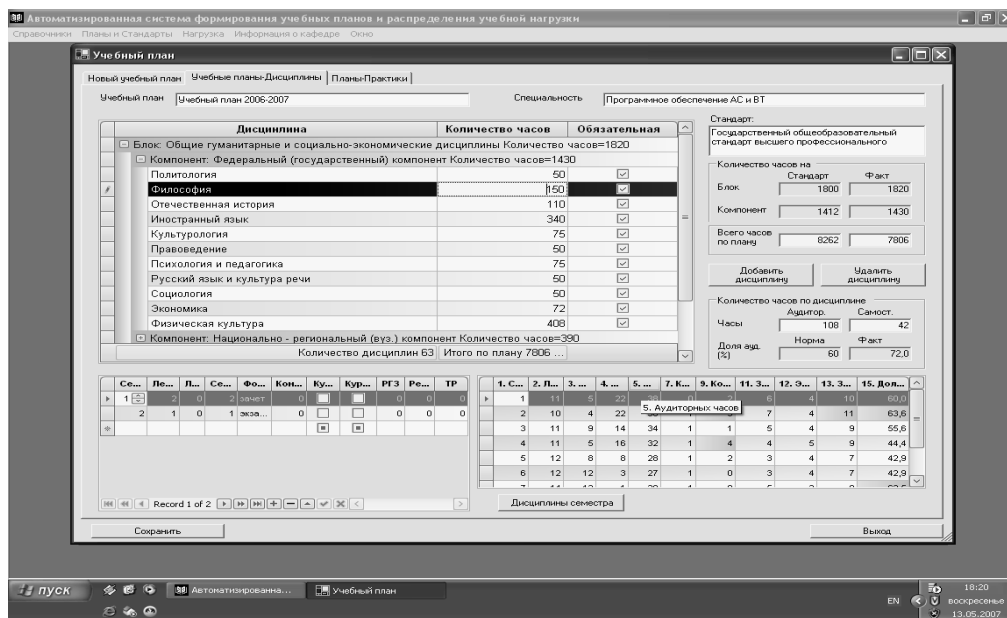


Рис. 3. Фрагмент интерфейса пользователя (формирование учебного плана в подсистеме “Кафедра”).

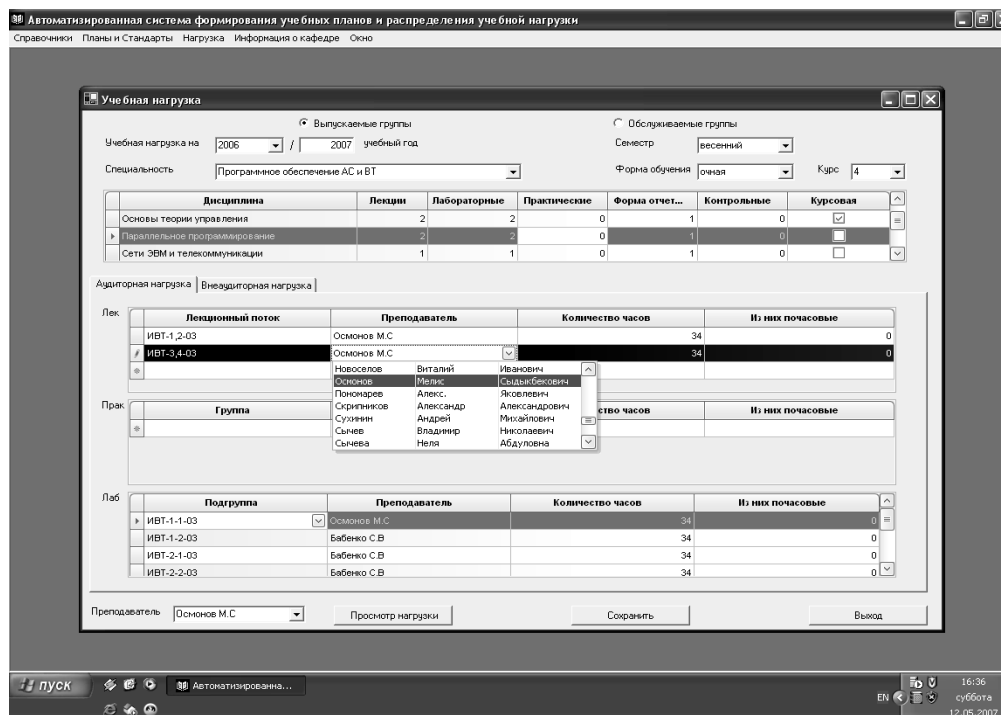


Рис. 4. Фрагмент интерфейса пользователя (распределение учебной нагрузки в подсистеме “Кафедра”).

а также набор хранимых процедур и функций пользователя, реализующих бизнес-логику работы системы. Система построена на единой базе данных, реплицируемой на локальные копии каждой из подсистем через коммуникационную сеть или при помощи переноса данных на электронных носителях информации.

Клиентское программное обеспечение разработано в среде MS Visual Studio 2003 [6]. Фрагменты интерфейса пользователя приведены на рис. 3 и 4.

Состав выходных документов интегрированной автоматизированной системы

Выходные документы подсистемы “Учебно-методическое управление”:

- список кафедр и перечень выпускаемых специальностей;
- сведения об обеспеченности образовательного процесса вуза профессорско-преподавательским составом (итоговые данные);
- отчет о вакансии по кафедрам вуза;
- сводный отчет о фактическом выполнении учебной нагрузки кафедрами вуза.

Выходные документы подсистемы “Кафедра”:

- ✚ учебный план;
- ✚ расчет и распределение учебной нагрузки ППС кафедры;
- ✚ сводная ведомость по распределению нагрузки ППС кафедры;
- ✚ сведения для индивидуальных планов преподавателей;
- ✚ сведения, подаваемые диспетчеру факультета для составления расписания;
- ✚ итоговый отчет кафедры о фактическом выполнении учебной нагрузки;
- ✚ сведения об успеваемости студентов кафедры;
- ✚ рабочие карты председателей и членов ГАК;
- ✚ протоколы и ведомости заседаний государственных аттестационных комиссий.

Выходные документы подсистемы “Деканат”:

- ✓ сведения о студенческом составе учебных групп;
- ✓ зачетно-экзаменационные ведомости;

✓ сводные отчеты об успеваемости студентов факультета.

Выходные документы подсистемы “Диспетчер”:

- расписание занятий по учебным группам;
- расписание занятий по преподавателям;
- расписание занятий по аудиториям.

Интегрированная автоматизированная система организации учебного процесса разработана применительно к Кыргызско-Российскому Славянскому университету, но может быть легко тиражируема для других вузов, имеющих аналогичную организационную структуру.

Литература

1. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – <http://www.citforum.ru/database/case/index.shtml>.
2. Гаврилец Е.З., Медведева О.А. Автоматизированная система формирования учебных планов и распределения учебной нагрузки преподавателей кафедры вуза // Современные наукоемкие технологии. – 2007. – №2.
3. Гаврилец Е.З., Медведева О.А. Оптимизация распределения годовой учебной нагрузки профессорско-преподавательского состава кафедры вуза // Проблемы автоматизации и управления. – 2006.
4. Использование пакетов прикладных программ для решения оптимизационных задач. – <http://csi.ucoz.ru/Pakety.htm>.
5. Kriegel A., Trukhnov B.M. SQL Bible. – Indianapolis, 2003.
6. Balena F. Programming Microsoft Visual Basic.NET. Version 2003. – Microsoft Press, 2004.