

УЧЕБНО-МЕТОДИЧЕСКАЯ ТЕСТИРУЮЩЕ-ОЦЕНИВАЮЩАЯ ПРОГРАММНАЯ СИСТЕМА ПОДДЕРЖКИ УЧЕБНЫХ КУРСОВ ПО ТЕОРИИ ПРИНЯТИЯ РЕШЕНИЙ

Алексей Волошин, Даниил Ковалёв

Аннотация: Рассматривается программное обеспечение поддержки курсов по теории принятий решений в виде учебно-методической тестирующей-оценивающей системы. Описывается эволюция его разработки, структура, назначение и функционирование. Приводится пример использования.

Ключевые слова: информационные технологии в образовании, учебно-методические программные системы, MS Visual Studio, C #, командная разработка, система защиты авторских прав, теория принятия решений.

ACM Classification Keywords: D.2.13 reusable Software - Reuse models.

Введение

На конференции MeL-2006 [Волошин, 2006] впервые были представлены результаты по созданию учебно-методического программного продукта (ПП) для поддержки нормативного учебного курса «Системы и методы принятия решений» (СМПР), читаемого на факультете кибернетики Киевского национального университета имени Тараса Шевченко для студентов третьего курса специальности «информатика» направления «прикладная математика». Читаемый курс состоит из 34 лекционных часов, 17 часов лабораторных занятий и 57 часов самостоятельной работы, его содержание базируется на учебном пособии [Волошин, 2006а] (с 2010 года - [Волошин, 2010]). В течение семестра студент должен был выполнить 3 лабораторные работы (с учетом сложности задания), которые заключались в создании программных модулей, реализующих конкретные алгоритмы решения задач теории принятия решений (ТПР). Организация лабораторных работ была следующей: первое занятие – постановка задачи; второе (через 2 недели) – промежуточный контроль, уточнение задания, консультация; третье – прием задания. Выполнение лабораторной работы осуществлялось студентами за счет часов, выделяемых на самостоятельную работу (10-12 часов на одну лабораторную). В случае, когда по реализации программного модуля имелись замечания, окончательный прием лабораторной работы осуществлялся в течение двух недель в неаудиторное время. После приема первой лабораторной работы выдавалось задание по второй и т.д. Окончательный прием выполненного задания осуществлялся «комиссией», состоящей из преподавателя и двух студентов. Результирующая оценка по курсу СМПР (исходя из 100-бальной шкалы) определялась следующим образом: 40 баллов (экзамен), 20 баллов (2 контрольные работы), 30 баллов (лабораторные работы), 10 баллов (коллоквиум по теоретическим вопросам курса за 2-3 недели до окончания семестра).

Первая версия программной системы SMPR0 [Волошин, 2006] представляла собой набор отдельных программных модулей, реализующих некоторые алгоритмы решения задач ТПР определенного раздела курса в соответствии с [Волошин, 2006]. Программные модули функционировали независимо, реализовывались на различных языках программирования, имели произвольную структуру и внешнее представление и оформление.

В следующей версии системы SMPR1 [Волошин, 2008] была реализована возможность взаимодействия программных модулей путем формирования унифицированных требований к их программной реализации, осуществляемой на языке программирования C # с помощью платформы .NET, среды разработки MS Visual Studio и технологии командной разработки SVN [Гамма, 2007].

Разработка проекта SMPR1 вызвала необходимость организации коллективной разработки ПП. Интересно отметить, что во время разработки проекта SMPR1 принципы коллективного создания программных продуктов на факультете кибернетики не только не применялись, но и не преподавались (со слов студентов) и эта проблема (коллективная разработка программного обеспечения) «намного сложнее, чем кажется на первый взгляд» (подробнее см. [Волошин, 2008]). Реализация отдельных разделов курса СМПП осуществлялась группами студентов под руководством «координаторов». Систему SMPR1 по функционированию содержательно можно назвать «учебно-методической демонстрационно-тестирующей». Наряду с демонстрацией решения конкретной задачи ТПП пользователь мог проверить правильность решения задачи, указывая ответ и получая оценку «правильно-неправильно».

В SMPR2 [Волошин, 2010а] была реализована функция оценивания знания студента-пользователя методов решения задач ТПП определенного раздела курса, расширен интерфейс системы, в частности, в использовании языков описания – украинский, русский, английский, китайский (отдельные модули). SMPR2, как приложение к учебному пособию [Волошин, 2010], была представлена участникам конференции MeL-2010 (Киев, сентябрь, 2010 г.). Система SMPR2 при чтении курсов по теории принятия решений уже используется в ряде вузов Украины (в частности, Киева, Одессы, Ужгорода и Черкасса).

Представляемую в данном докладе версию SMPR3 авторы считают «промышленным» вариантом системы. По сравнению с SMPR2 в SMPR3 добавлено две основные функции: 1) оценка знаний пользователя-студента по всему курсу СМПП с учетом всех разделов. Общая оценка выдается в нечеткой форме (например, «Ваши знания можно оценить между «удовлетворительно» и «хорошо», но ближе к оценке «удовлетворительно»; «Вам необходимо подучить такой-то раздел курса»); 2) вторая функция связана с системой защиты программной системы от несанкционированного использования (см. ниже). Версия SMPR3 будет прилагаться на отдельном диске к следующему изданию учебного пособия по ТПП.

В следующей версии SMPR4 авторами планируется осовременить систему, переведя ее в среду Интернет, т.е. создать сайт, содержащий полную функциональность системы и доступный любому пользователю в любое время.

Общая информация о программной системе SMPR

Система состоит из ядра и набора специализированных модулей, который отвечает за решение определенного класса задач. Ядро создает среду функционирования модулей с возможностью параллельного решения задач и обмена данными между модулями. Оно представляет собой общие интерфейсы, стандарты обмена данными, системы помощи и информации о модуле.

Ядро включает буфер - средство обмена данными между модулями. Буфер может манипулировать исходными данными разного рода, а также результатами их обработки. Он представляет собой "среду обитания" данных уровня системы, то есть данных, к которым может получить доступ любой модуль. На архитектурном уровне буфер представляет собой специальный класс, который предоставляет модулям интерфейсы загрузки, хранения и валидации данных. Ядро системы не является специализированным под конкретные алгоритмы конкретного учебного курса. Конкретику задач предоставляет только тот набор внешних модулей, который доступен ядру для использования. Именно это свойство ядра системы позволяет использовать его как основу для разработки учебно-методических систем для различных учебных курсов, имеющих структуру "класса задач - методы их решения" с возможностью обмена данными между классами. Структура модуля, отвечающего за конкретный класс задач, построена таким образом, что ядро может автоматически определять не только наличие модуля, но и некоторые его характеристики (время использования, количество попыток использования и т.п.).

Система спроектирована с учетом возможности расширения ее функциональности путем добавления новых классов задач, а также методов решения имеющихся задач. Для расширения перечня классов решаемых задач создается новый проект модуля, в котором реализуются заложенные на этапе проектирования абстрактные классы. Созданный проект компилируется в библиотеку, которая добавляется в папку плагинов (дополнений) системы. После запуска программного комплекса новая функциональность будет автоматически добавлена в главное меню проекта. Добавление новых методов решения задач реализуется несколько иначе. Оно требует ознакомления с абстрактным классом конкретного модуля. В каждом проекте модуля, а также в шаблоне проекта, находится специальная папка, в которой содержатся файлы классов методов решения задач данного модуля. Добавление нового метода заключается в добавлении в эту папку нового файла, в котором будет реализован класс метода, после чего необходима перекомпиляция всего проекта. Поскольку система состоит из ядра и множества модулей, представленных в форме плагинов, наличие или отсутствие отдельно взятого модуля не влияет на работоспособность других модулей системы в целом.

Схема работы отдельного модуля представлена на рис.1.

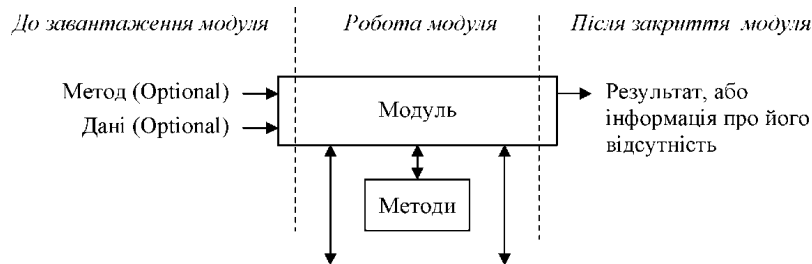


Рис.1. Схема работы модуля

Каждая тема представлена отдельным модулем в отдельном окне. Данные могут быть введены как с клавиатуры, так и из буфера программы. Каждый модуль имеет систему тестирования, предназначенную для проверки знаний.

Вид и функциональность основного окна системы системы

«Шапка» программы (1) включает в себя закладки:

1) «Файл», которая позволяет быстро закрыть программу; 2) «Модули», с которой можно запустить любой модуль; 3) «Язык», которая позволяет переключить интерфейс программы на один из четырех языков (украинский, русский, английский, китайский); 4) «Помощь». Закладка «Помощь» имеет два подраздела:

3.1) «О программе», который вызывает окно с полной информацией о разработчиках проекта и их контактами; 3.2) «Модули», который открывает дополнительное меню, привязанное к каждому из подключенных модулей и содержащее информацию по использованию модуля и теоретический материал.

На рис. 2 представлен основной интерфейс системы, в котором описаны разделы курса ТПР (содержание окон в данном случае не принципиально, далее содержание окна будет описано на примере одного из разделов).

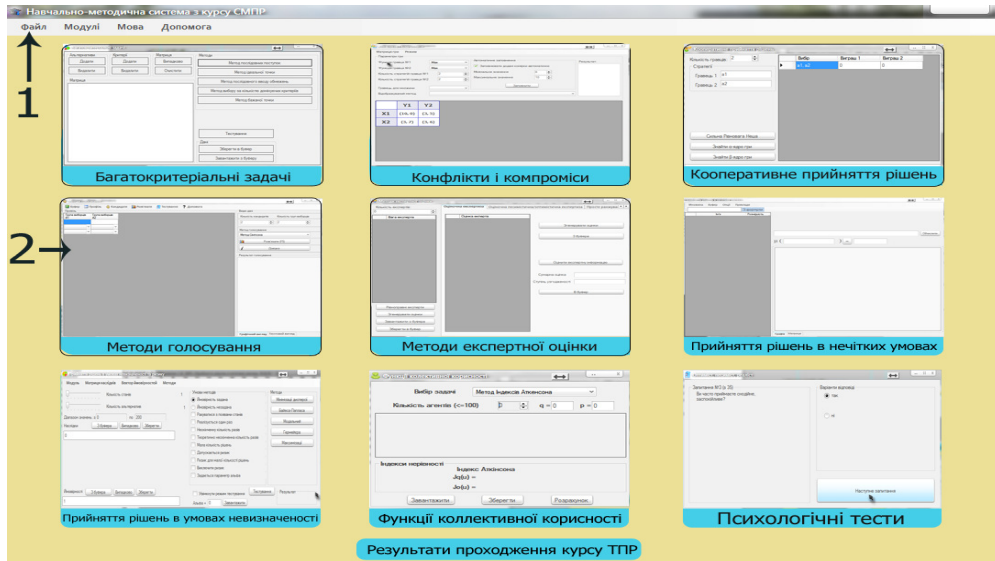


Рис.2. Основной интерфейс программы

Пример работы модуля «Конфликты и компромиссы»

Рассмотрим функционирование и внешний вид модуля «Конфликты и компромиссы» (рис.3). С помощью опций можно задать условия решаемой задачи. Входная матрица может быть задана вручную, загружена из буфера обмена или сгенерирована автоматически, если необходимо.

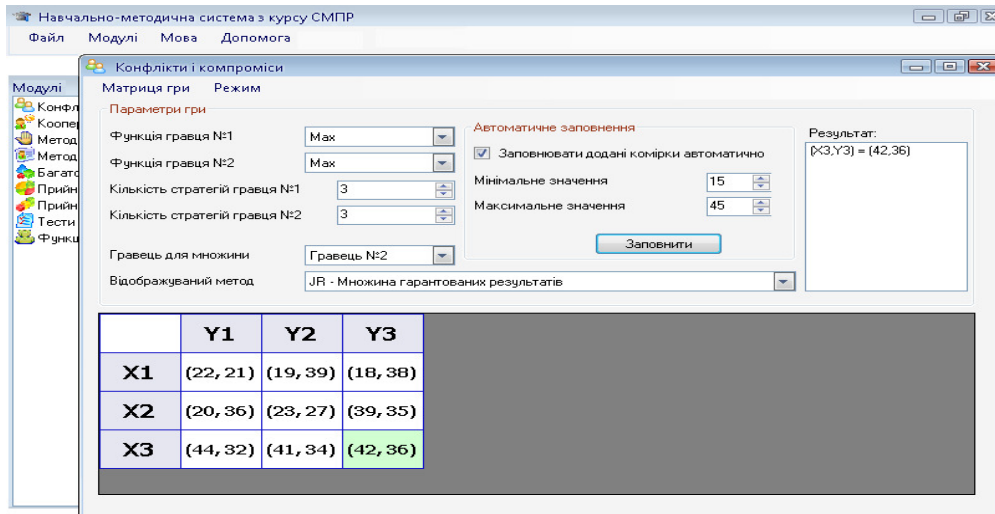


Рис.3. Структура модуля «Конфликты и компромиссы»

В модуле содержится 10 методов, каждый из которых может быть использован. Решение подсвечено зеленым во входной матрице и выводится в отдельном поле для удобства. У каждого модуля есть также второй режим - режим тестирования и имеется возможность самостоятельно выбирать методы для тестирования знаний пользователя. Для введения ответа достаточно «щелкнуть» курсором мыши по полю, где содержится ответ. Кнопка «Проверить» укажет правильный результат и выведет количество заработанных баллов. В дальнейшем этот результат автоматически сохраняется в ядре системы для демонстрации, по требованию пользователя, уровня его знаний.

Во всех модулях имеется функция случайного заполнения данными для быстрой демонстрации или проверки знаний. С помощью удобных элементов интерфейса можно задать собственные значения и мгновенно получить решение. Как уже было сказано, каждый модуль может взять значение из буфера программы, который сохраняет произвольное количество значений скалярного, векторного и матричного типа. На закладке «Помощь» имеется вся теоретическая информация (постановка задачи, методы решения и т.п.), необходимая для усвоения материала, использованного в модуле.

Система контроля знаний

В каждом из модулей программной системы реализованы методы, которые предлагают студенту постановку задачи и дают определенное задание: например, нахождение оптимальной, по некоторым критериям, стратегии; определение необходимого множества состояний системы и др. Задачи генерируются автоматически, во избежание повторений, и могут иметь различную сложность (размерность задачи, количество состояний или участников и т.п.).

Вся информация об оценках, полученных во время прохождения тестов, хранится в системе и может быть получена пользователем в любое время в виде окна с общими результатами тестирования по всем разделам курса и рекомендации по тем разделам, которые необходимо изучить лучше.

Система защиты

Система защиты ограничивает возможности несанкционированного копирования программы и позволяет «внимательнее относиться» к пользователям программы.

Специфика системы защиты заключается в том, что во время первого запуска программа просит прислать разработчику уникальный идентификатор, на основе которого будет создан ключ для программы (Рис.4). Этот ключ будет работать только для компьютера, на котором был получен идентификатор.

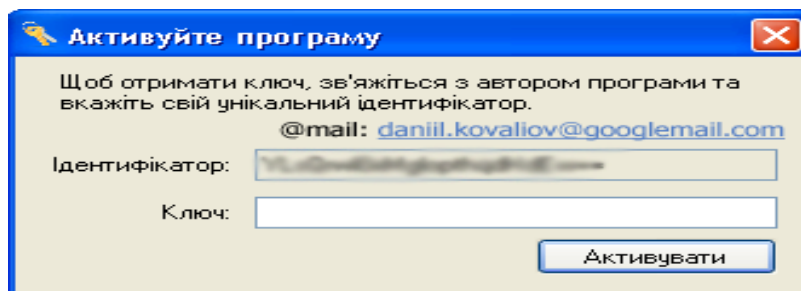


Рис. 4. Система активации

Этот принцип реализуется путем считывания серийного номера жесткого диска и шифрования его в уникальный код. Администратор получает этот код от пользователя, например, по электронной почте, и на его основе, с помощью специальной программы, создает ключ для активации. Таким образом, администратор имеет базу данных пользователей программы. Это позволяет предоставлять им обновления и в любой момент оказывать помощь по использованию программы.

Заключение

Представляемая в докладе учебно-методическая тестирующе-оценивающая программная система SMPR3 поддержки учебного курса «Системы и методы принятия решений» является полноценным программным продуктом для проведения лекций и лабораторных работ по теории принятия решений, тестирования и оценки знаний студентов. Авторы рекомендуют использовать ее в других вузах, а также

использовать ее инструментальные средства (ядро) для разработки программных продуктов поддержки иных учебных курсов.

В дальнейшем авторами планируется перевести систему в среду Интернет, создать сайт, который будет иметь полную функциональность системы и будет доступен любому пользователю в любое время.

Благодарности

Работа опубликована при финансовой поддержке проекта ITHEA XXI Института информационных теорий и приложений FOI ITHEA Болгария www.ithea.org и Ассоциации создателей и пользователей интеллектуальных систем ADUIS Украина www.aduis.com.ua.

Список литературы

- [Волошин, 2006] А. Волошин, К. Березовский., И. Дроздов. В создание коллективных учебно-методических программных продуктов по курсу «Теория принятия решений» // Труды конф. «MeI-2006», София, 2006. - С.67-70.
- [Волошин, 2006а] А.Ф. Волошин, С.А. Мащенко. Теория принятия решений. Учебное пособие. - Киев: Издательско-полиграфический центр «Киевский университет», 2006. - 304 с. (Укр.яз.).
- [Волошин, 2010] А.Ф. Волошин, С.А. Мащенко. Модели и методы принятия решений. Учебное пособие. - Киев: Издательско-полиграфический центр «Киевский университет», 2010. - 336 с. (Укр.яз.).
- [Волошин, 2008] Об опыте коллективной разработки учебно-методических программных систем, труды конференции «MeI-2008». - София, 2008.
- [Гамма, 2007] Э.Гамма, Р. Хелм, Дж. Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. - М.: Издательство «Питер», 2006.
- [Волошин, 2010а] О.Ф. Волошин. Про досвід колективної розробки навчально-методичних програмних систем. Ужгород: Праці V Міжнародної школи-семінару «Теорія прийняття рішень», Ужгород, 2010.-274 с.
-

Информация об авторах

Алексей Волошин – профессор Киевского национального университета имени Тараса Шевченко, факультет кибернетики; e-mail: olvoloshyn@ukr.net

Даниил Ковалев – студент КНУ им. Т. Шевченко; e-mail: daniil.kovaliov@gmail.com