

УЧЕБНЫЙ КУРС «ИНТЕРФЕЙСЫ ИНФОРМАЦИОННЫХ СИСТЕМ» В ОБУЧЕНИИ СПЕЦИАЛИСТОВ ПО ИНФОРМАЦИОННЫМ И КОММУНИКАЦИОННЫМ ТЕХНОЛОГИЯМ

Д. И. Черемисинов

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: cher@newman.bas-net.by

Обосновывается целесообразность преподавания курса «Интерфейсы информационных систем» в обучении специалистов по информационным и коммуникационным технологиям как средства формирования объема знаний о взаимосвязи ключевых концепции и принципов программирования.

ВВЕДЕНИЕ

Дисциплина «Интерфейсы информационных систем» читается в БГУИР кафедрой инженерной психологии и эргономики с 2010 г. Предметная область дисциплины является важной составляющей компетенции выпускника по специальности 1-58 01 01 «Инженерно-психологическое обеспечение информационных технологий». Дисциплина читалась во втором семестре четвертого курса.

Область знаний, связанная с информационными технологиями, принято разделять на четыре основные дисциплины – информатика (computer science), программная инженерия (software engineering), проектирование аппаратных платформ (hardware engineering) и информационные системы (information systems) [1,2]. Профессиональными объединениями разработаны рекомендации по преподаванию программирования и информатики в университетах [1,2], которым должны следовать учебные программы вузов, чтобы квалификация выпускаемых ими специалистов соответствовала международным требованиям. В рекомендациях [1] различаются объем знаний (body of knowledge) выпускников и описание конкретных учебных курсов, посредством которых выпускники овладевают этими знаниями.

В [1] предлагается строить отдельные руководства по разработке учебных планов для подготовки специалистов в каждой из пяти дисциплин. Описание объема знаний по дисциплине «программная инженерия» организовано в виде трехуровневой иерархии. На верхнем уровне иерархии находятся области преподаваемых знаний, представляющие собой отдельные поддисциплины программной инженерии. Каждая сфера состоит из своей группы тематических модулей, которые обозначаются аббревиатурой сферы с добавлением порядкового номера. Каждый модуль состоит из тем. Название «Интерфейсы информационных систем» не является ни названием модуля, ни темы.

В [1] рекомендуется среди прочего в качестве одной из целей обучения формировать у

студентов инженерный склад ума, основанный на математическом подходе к решению проблем. Для этого в преподавании других дисциплин инженерного профиля задачей обучения является формирование целостной системы взаимосвязанных понятий. Несмотря на то, что важность математической компетенции выпускника неоднократно отмечена в [1], там отсутствует в явном виде рекомендация формирования у выпускника именно целостной системы взаимосвязанных понятий, представляющих объем знаний по информационным технологиям, как одна из целей преподавания. Эта система должна охватывать ключевые концепции, принципы и проблемы программирования.

В настоящей работе обосновывается целесообразность преподавания обсуждаемого курса как средства формирования объема знаний о взаимосвязи ключевых концепции и принципов программирования.

I. ОТБОР ПРЕПОДАВАЕМОГО МАТЕРИАЛА УЧЕБНОГО КУРСА

В учебных программах ведущих вузов, готовящих программистов, имеются курсы, целью которых является формирование общего взгляда на объем знаний по информационным технологиям в целом. Часто этот курс называется системным или теоретическим программированием. Например, в МГУ имеется курс «Фундаментальные аспекты системного программирования. Теория и практика разработки системного ПО и компонентов ядра ОС Linux». В этом курсе основой объединения ключевых понятий программирования является задача построения операционной системы (ОС). В БГУ на кафедре технологий программирования читается дисциплина «Системное программирование». Следует отметить, что дисциплина «Системное программирование» не везде используется для формирования общего взгляда, часто это спецкурс для подготовки системных администраторов (сисадмин). Системный администратор — это высший IT менеджер, определяющий как и куда будет двигаться всё IT в компании, а в конечном итоге

на сколько просто и удобно будет пользоваться «благами ИТ» каждому конкретному пользователю. Во многих вузах основой курса «Системное программирование» является задача построения компилятора языка программирования.

В обсуждаемом курсе основой материала для преподавания является не задача, а понятие, именно понятие взаимодействия. Информационное взаимодействие – это воздействие объектов или агентов друг на друга, приводящее к изменению их состояния. Понятие взаимодействия является одним из центральных в кибернетике и информатике. Это понятие связано с понятием интерфейса. Слово интерфейс пришло в профессиональный язык программистов из английского языка, где его определяют как: Interface is a shared boundary between two separate subsystems (сопряжение, линия раздела, перегородка двух подсистем) или как Interface is a point of interaction between subsystems (точка взаимодействия подсистем). Понятие взаимодействия естественно подходит для объединения как ключевых так и самых современных понятий программирования.

Текущее состояние набора понятий, связанных с программированием, фиксируется в глоссариях терминов (например [3]). Глоссарий [3] содержит около 12000 понятий. Очевидно, учебный курс не может использовать такое количество понятий. В преподавании курсов, формирующих математический подход к решению проблем, стремятся обойтись небольшим количеством понятий (обычно порядка 30). Глоссарий терминов, связанных с компьютерами, для пенсионеров [3] содержит 37 понятий. Для профессионального обучения в обсуждаемом курсе выбраны 150 понятий. Довольно, часто отбор терминов в глоссарии осуществляется статистическими методами, путем анализа литературы. В глоссарии обсуждаемого курса отобраны понятия ядра информационных технологий, встречающиеся во всех четырех основных дисциплинах. Порядок обсуждения выбранных понятий формировался так, чтобы создать представление о их внутренней взаимосвязи.

II. МЕСТО КУРСА В УЧЕБНОМ ПЛАНЕ

О понятиях «Кривая обучения» (learning curve) и «Крутая кривая обучения» сейчас часто говорят и пишут в области компьютерных наук. Кривую обучения можно рассматривать как представление зависимости умственных затрат на обучения чему-либо от времени. Умственные затраты могут быть измерены в числе изученных понятий. Именно такой подход используется в рекомендациях [1]. Число понятий, которыми должен овладеть выпускник, должно постоянно возрастать. Понятия ядра информационных технологий, встречающиеся во всех четырех ос-

новных дисциплинах, рекомендуется изучать на этапе первоначального обучения. При таком подходе обучаемым трудно улавливать взаимосвязи понятий, разделенных временем в процессе обучения, и компетентность формируется случайно. Однако, овладение системой понятий ядра программирования уже требуют определенного объема знаний. По этой причине обсуждаемый курс в учебном плане расположен на завершающей стадии обучения.

III. ОЦЕНКА ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

В начале этого курса обучаемые уже знакомы с понятиями, которые будут изучаться. Но этот курс не повторение пройденного. Так как целостность системы знаний в обсуждаемом курсе является важной характеристикой успешности усвоения курса, то нужна специальная методика ее контроля, состоящая в проверке «интеграции знаний». В качестве меры целостности системы понятий используется степень взаимосвязи между его базовыми понятиями, которую демонстрирует студент [5]. В начале курса студенты выполняют тест, в котором задачей является указание взаимосвязей терминов из глоссария курса. Результат теста – это граф, вершинами которого являются понятия глоссария, а ребрами – связи между понятиями. Цель курса состоит в том, чтобы этот граф был односвязным. В конце курса студенты выполняют этот тест повторно. Курс усвоен успешно, если в графе, построенном по результатам повторного теста повышается связность. Этот метод контроля усвояемости курса использовался при чтении курса последние три года. Односвязный граф никогда не возникал в тесте, проводимом в начале курса. В повторном тесте мера связности повышалась у подавляющего большинства студентов.

1. IEEE/AIS/ACM Joint Task Force on Computing Curricula. Computing Curricula 2005. The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering. 2005 (<http://www.computer.org/curriculum> или <http://www.acm.org/education/curricula.html>).
2. Рекомендации по преподаванию информатики в университетах / Ред. В. Л. Павлов, А. А. Терехов. – СПб.: Изд-во СПбГУ, 2002. – 367С.
3. Glossary of Computer Related Terms [Electronic resource] / Hohn M. W. – 2011. – Mode of access: <http://www.math.utah.edu/wisnia/glossary.html> – Date of access: 1.09.2016.
4. Glossary of Computer and Internet Terms for Older Adults [Electronic resource] / National Institute on Aging – 2016. – Mode of access: <http://www.cheyney.edu/information-technology/documents/glossarycomputerterms.pdf> – Date of access: 1.09.2016.
5. Еремин Е.А. Количественная оценка целостности системы базовых понятий как мера усвоения учебного материала / Образовательные технологии, 2014, N 4, с.78-98.