

Ю.Н.Антипов¹, Г.Т.Омаров², Б.К.Шаяхметова³

¹Калининградский технический университет, Россия;

²Карагандинский экономический университет Казпотребсоюза;

³Карагандинский государственный университет им. Е.А.Букетова (E-mail: kazahzavod@mail.ru)

Методика преподавания сложных систем и проблемы их разработки

В статье рассмотрена методика разработки программного обеспечения для сложных систем и проблемы её преподавания. Известно, что в обучении общение с преподавателем является центральным местом в учебном процессе. Однако роль преподавателя не сводится к передаче и интерпретации знаний. Его основная функция — управление познавательной деятельностью и активизация творческой активности студента, поэтому в обучении использованы методы, требующие активного участия студента, — активные методы обучения. Именно с этих позиций в работе исследована методика разработки программного обеспечения сложных систем.

Ключевые слова: сложные системы, программное обеспечение, методика разработки, компьютерные технологии, процесс обучения, требования к программным системам, учебный план, модель учебного процесса, мультимедийные средства, объектно-ориентированные концепции.

Исследование компьютерных технологий в процессе обучения позволяет эффективно готовить специалистов, знакомить их с новыми идеями и моделями. Большинство современных программных систем объективно очень сложны. Это сложность обуславливается многими причинами, главная из которых логическая сложность решаемых ими задач. Эту сложность можно существенно сгладить, если при обучении использовать такие компьютерные технологии, которые обеспечивали бы индивидуальность этого процесса, диалоговые формы взаимодействия студентов и систем обучения.

В наше время, когда созданы мощные компьютерные сети, появилась возможность переложить на них решение сложных ресурсоемких задач, о компьютеризации которых раньше никто и не думал. Сейчас в процесс компьютеризации вовлекаются совершенно новые предметные области, а для уже освоенных областей усложняются уже сложившиеся постановки задач.

Для решения задач такого класса необходимо совершенствовать содержание форм и методов обучения информатики на основе развития творческого мышления студентов: исследование психолого-педагогических закономерностей познавательной деятельности студентов и поиски эффективных путей руководства ею и др.

В свете этого тезиса необходимо отметить, что дополнительными факторами, увеличивающими сложность программных систем, являются:

- сложность формального определения требований к программным системам;
- отсутствие удовлетворительных средств описания поведения дискретных систем с большим числом состояний при недетерминированной последовательности входных воздействий;
- коллективная разработка;
- необходимость увеличения степени повторяемости кодов.

В последнее время наблюдается активизация творчества в области педагогики. Большинство педагогических работников, движимых самыми различными мотивами, предлагают обществу свое собственное, авторское видение педагогических проблем и их решение. При этом авторы педагогических новшеств вырабатывают новые понятия, принципы, технологии. Но далеко не всегда в их работах можно найти аргументированное обоснование новизны и полезности предлагаемых нововведений с соответствующим анализом их содержания в сравнении с известными объектами педагогической и практической деятельности [1].

В этой связи и исследуется новый термин — сложность определения требований к программным системам. Сложность определения требований к программным системам обуславливается двумя факторами. Во-первых, при определении требований необходимо учесть большое количество различных факторов. Во-вторых, разработчики программных систем не являются специалистами в автоматизируемых предметных областях, а специалисты в предметной области, как правило, не могут сформулировать проблему в нужном ракурсе.

Известно, что педагогический процесс — способ организации образования, заключающийся в целенаправленном отборе и использовании внешних и внутренних факторов развития участников —

образовательных отношений. Признаками педагогического процесса являются цели, принципы, содержание, методы, средства и формы их взаимосвязи, их количество и объем, их внутренняя структура, последовательность и чередование.

В этом аспекте рассмотрим отсутствие удовлетворительных средств формального описания поведения дискретных систем. В процессе создания программных систем используют языки сравнительно низкого уровня. Это приводит к ранней детализации в процессе создания программного обеспечения и увеличивает объем описаний разрабатываемых продуктов, который, как правило, превышает сотни тысяч операторов языка программирования. Средств же, позволяющих детально описывать поведение сложных дискретных систем на более высоком уровне, чем универсальный язык программирования, не существует.

Известно, что в обучении общение с преподавателем является центральным местом в учебном процессе. Однако роль преподавателя не сводится к передаче и интерпретации знаний. Его основная функция — управление познавательной деятельностью и активизация творческой активности студента. Поэтому в обучении используются методы, требующие активного участия студента, — активные методы обучения. При этом важное значение приобретает организация коллективной исследовательской деятельности, актуализирующей учебно-познавательные и профессиональные мотивы для студентов, развивающей их творческое мышление.

Наличие современных мультимедийных средств и технологий позволяет создавать качественные программы учебного назначения, дидактическая ценность которых может быть значительно выше обычных лекционных курсов.

В этом плане рассматривается один из аспектов проблемы разработки сложных программных систем — коллективная разработка, так как из-за больших объемов проектов разработка программного обеспечения ведется коллективом специалистов. Работая в коллективе, отдельные специалисты должны взаимодействовать друг с другом, обеспечивая целостность проекта, что при отсутствии удовлетворительных средств описания поведения сложных систем, упоминавшихся выше, достаточно сложно. Причем, чем больше коллектив разработчиков, тем сложнее организовать процесс работы.

Рассматриваемое нами обучение характеризуется гибким характером учебного процесса, который учитывает индивидуальные особенности студента, давая возможность ему самому выбрать удобный для него темп работы. Поэтому важным элементом становится управление учебным процессом. Как правило, это управление осуществляется с помощью специализированного программного обеспечения, предназначенного для доставки и распределения учебного материала, контроля за выполнением учебного плана и контроля качества обучения. Необходимо также в течение всего учебного года осуществлять мониторинг развития личностных и познавательных характеристик студента для корректировки процесса обучения.

Таким образом, новые требования вызывают необходимость построения новой модели учебного процесса, создания качественно новой системы управления образованием, формирования новой модели личности, существующей в условиях открытого информационного общества.

В этой связи следует вспомнить еще об одной проблеме разработки сложных программных систем, о необходимости увеличения степени повторяемости кодов. На сложность разрабатываемого программного продукта влияет и то, что для увеличения производительности труда компании стремятся к созданию библиотек компонентов, которые можно было бы использовать в дальнейших разработках. Однако в этом случае компоненты приходится делать более универсальными, что, в конечном итоге, увеличивает сложность разработки.

Анализ научной литературы по данной тематике показывает, что наилучшие результаты в решении задач оценки качества могут быть достигнуты при использовании методологии системного анализа. Глобальную цель реализации учебного плана, с учетом того, что конечным продуктом данного проекта являются специалисты конкретной специальности, можно сформулировать следующим образом: обеспечение выпуска специалистов с высшим образованием по конкретной специальности, отвечающих требованиям стандартов, востребованных рыночной экономикой, при эффективном использовании материально-технических и кадровых ресурсов университета [2].

Практика показывает, что подавляющее большинство сложных систем как в природе, так и в технике имеет иерархическую внутреннюю структуру. Это связано с тем, что обычно связи элементов сложных систем различны как по типу, так и по силе, что и позволяет рассматривать эти системы как некоторую совокупность взаимозависимых подсистем. Внутренние связи элементов таких подсистем сильнее, чем связи между подсистемами.

В свою очередь, используя то же различие связей, можно каждую подсистему разделить на подсистемы и т.д. до самого нижнего «элементарного» уровня, причем выбор уровня, компоненты которого следует считать элементарными, остается за исследователем. На элементарном уровне система, как правило, состоит из немногих типов подсистем, по-разному скомбинированных и организованных.

Поведение системы в целом обычно оказывается сложнее поведения отдельных частей, причем из-за более сильных внутренних связей особенности системы в основном обусловлены отношениями между ее частями, а не частями как таковыми.

Итак, как бы ни была построена программа по разработке сложных систем, введение в теорию иерархий необходимо, поскольку в природе существует еще один вид иерархии — иерархия «простое — сложное», или иерархия развития (усложнения) систем в процессе эволюции. В этой иерархии любая функционирующая система является результатом развития более простой системы. Именно данный вид иерархии реализуется механизмом наследования объектно-ориентированного программирования.

Несмотря на разнообразие иерархий, изложение информатики как учебного предмета должно выявить ее единство как науки, вооружить студентов общими методами. Вкрапление элементов объектно-ориентированного программирования в изложение информационных дисциплин позволяет укрупнить дидактические единицы на лекциях и практических занятиях, например, под единым углом зрения рассмотреть теорию сложных систем.

Объектно-ориентированные концепции, несомненно, имеют фундаментальное значение не только для самой информатики, но и для прикладных научных дисциплин (физических, технических, биологических и др.). Эти концепции используются не только при построении основ науки, но также в педагогическом процессе. Без объектно-ориентированного подхода часто не удастся осмыслить центральные понятия, например, при проектировании программного обеспечения.

Будучи в значительной степени отражением природных систем, программные системы обычно являются иерархическими, т.е. обладают описанными выше свойствами. На этих свойствах иерархических систем строится блочно-иерархический подход к их исследованию или созданию. Этот подход предполагает сначала создавать части таких же объектов (блоки, модули), а затем собирать из них сам объект.

Объектно-ориентированный подход позволяет глубже понять и проникнуть в сущность, свободно взглянуть на реальные явления в самых разнообразных отраслях знаний, например, программное обеспечение ПЭВМ, алгоритмизация и программирование, базы и банки данных, компьютерные сети и т.д. Объектно-ориентированный подход может быть использован в самом педагогическом процессе как дидактический прием доступного объяснения непонятного. Ведь каждое определение, утверждение можно сформулировать в терминах объектно-ориентированного подхода.

Трудности в изучении информатики иногда связаны с узким взглядом. Воспитание более широкого взгляда на информационные объекты, повышение теоретического, научного уровня делают информатику более доступной в процессе обучения. При этом должное место в преподавании должны занять индукция и правдоподобные рассуждения, аналогия, обобщение и специализация. И с этих позиций рассматривается в педагогике процесс разбиения сложного объекта на сравнительно независимые части, который получил название декомпозиции. При декомпозиции учитывают, что связи между отдельными частями должны быть слабее, чем связи элементов внутри частей. Кроме того, чтобы из полученных частей можно было собрать разрабатываемый объект, и в процессе декомпозиции необходимо определить все виды связей частей между собой.

При создании очень сложных объектов процесс декомпозиции выполняется многократно: каждый блок, в свою очередь, декомпозируют на части, пока не получают блоки, которые сравнительно легко разработать.

Декомпозиционный подход, систематически проводимый в процессе изучения информатики, позволяет представить мыслительную деятельность о процессе обучения какой-либо теме, о виде множества достаточно элементарных мыслительных операций и разработать методику их кооперационного формирования так, как это традиционно делается, например, при изучении аппаратных средств современных компьютеров. Тщательно продуманный декомпозиционный подход поставит методику изучения сложных систем на прочную научную основу. Такой подход позволит разработать достаточно общие регулярные приемы мышления. Существенно и то, что в процессе декомпозиции стараются выделить аналогичные блоки, которые можно было бы разрабатывать на общей основе. Таким

образом, как уже упоминалось выше, обеспечивается увеличение степени повторяемости кодов и, соответственно, снижение стоимости разработки.

Результат декомпозиции обычно представляют в виде схемы иерархии, о которой говорилось выше, на нижнем уровне которой располагают сравнительно простые блоки, а на верхнем — объект, подлежащий разработке. На каждом иерархическом уровне описание блоков выполняют с определенной степенью детализации, абстрагируясь от несущественных деталей. Следовательно, для каждого уровня используют свои формы документации и свои модели, отражающие сущность процессов, выполняемых каждым блоком. Так для объекта в целом, как правило, удается сформулировать лишь самые общие требования.

Список литературы

- 1 *Бадд Тимоти*. Объектно-ориентированное программирование в действии / Пер. с англ. — СПб.: Питер, 1997.
- 2 *Шень А.* Программирование: теоремы и задачи. — М.: МЦНМО, 1995.

Ю.Н.Антипов, Ф.Т.Омаров, Б.Қ.Шаяхметова

Күрделі жүйенің оқыту әдістемесі және оның зерттемесінің мәселелері

Мақалада бағдарламалық камсыздандырудың зерттемесінің әдістемесі күрделі жүйе және оның оқыту мәселесі үшін қарастырылды. Тәлім-тәрбиеде өзара байланыс оқытушымен жанасушылық оқу үдерісте орталық орынға ие. Алайда оқытушының рөлі беріліс және білімді талдап түсіндірулерімен шектелмейді. Оның негізгі қызметі — басқару танымдық қызметпен және студенттің шығармашылық белсенділігін арттыру, сол себептен тәлім-тәрбиеде студенттің белсенді қатысуын қолданатын әдістер — тәлім-тәрбиенің белсенді әдістері пайдаланылады. Осы тұрғыдан күрделі жүйенің бағдарламалық камсыздандыруының зерттемесінің әдістемесі зерттелді.

Yu.N.Antipov, G.T.Omarov, B.K.Shayakhmetova

Teaching methodology of complex systems and problems of development

In this work, the technique of development of the software for difficult systems and a problem of its teaching is considered. It is known, that in training communication with the teacher is the central place in educational process. However, the role of the teacher is not reduced to transfer and interpretation of knowledge. Its main function management of informative activities and activation of creative activity of the student, therefore, the training methods are used, requiring the active participation of students — active learning methods. From these line items in work, the technique of development of the software of difficult systems is researched.

References

- 1 Timothy Budd. *Object-oriented programming in action* / Transl. from English, Sankt-Petersburg: Piter, 1997.
- 2 Shen A. *Programming: theorems and problems*. — Moscow: MCCME, 1995.