

# Триединая задача педагогического эксперимента в области ИТ-образования

Санкт-Петербургский государственный университет  
информационных технологий, механики и оптики  
**А.А. Шальто,**  
докт. техн. наук, профессор



А.А. Шальто

В статье рассматриваются результаты педагогического эксперимента в области ИТ-образования, в рамках которого решаются три задачи: повышается качество обучения; проводится научная работа; совершенствуется технология автоматного программирования.

***Отделить учебное от научного нельзя. Но научное без учебного все-таки светит и греет. А учебное без научного – только блесит.***  
**Н.И. Пирогов**

В 1991 г. в России были разработаны основы технологии автоматного программирования [1], которая была

названа также «Switch-технология» [2] или «Программирование с явным выделением состояний». В настоящее время автоматное программирование рассматривается в работе [3] в качестве одного из стилей программирования.

В соответствии с парадигмой автоматного программирования программы предлагается строить так, как строятся системы управления технологическими процессами, в которых выделяются источники информации, управляющее устройство и объекты управления. При этом в качестве управляющего устройства используется детерминированный конечный автомат. Он реагирует на входные воздействия и формирует выходные

Настоящего специалиста университет может вырастить, лишь если в нем «учебное и научное» неразделимы.

воздействия, «указывающие» объектам управления, что они должны делать. В управляющем устройстве может быть более одного автомата. Эти автоматы могут взаимодействовать между собой, например, за счет вложенности.

Предлагаемая технология была апробирована при создании ПО для судовых систем управления [2]. Эта технология, первоначально предложенная для систем логического управления, требовала своего развития.

Все это происходило в России в середине 90-х годов, когда финансирование научных исследований было фактически свернуто. По этой причине привлечь к научным исследованиям молодежь было практически невозможно, особенно учитывая то, что для проведения указанных исследований требовались программисты, которые не имели и не имеют проблем с трудоустройством как у нас в стране, так и за рубежом.

В отсутствии материальных стимулов для развития технологии приходилось использовать единственный моральный стимул – аспирантуру, которая в очной форме «защищает» от армии. При этом, как бы аспирантура ни называлась (очная или заочная), суть от этого не изменялась, так как на первом месте у аспирантов была работа за деньги, а диссертационные исследования находились в «сухом ос-

татке». В это время я руководил двумя аспирантами, совместно с которыми, в частности, удалось разработать вариант технологии создания ПО для событийных систем [4].

Однако, несмотря на все мои усилия, непрерывно проводить исследования в указанной области не удавалось, так как «ребята хотели не только работать, но и жить», что нормально только в нормальной обстановке, а не в сложившейся в нашей стране в то время. Известно, что «надо иметь мужество, чтобы требовать с людей работу, не имея денег». Мужество у меня было, но этого оказывалось недостаточно, так как единственный источник финансирования (отечественные гранты) не позволял ни одному из нас оставить основную работу.

В общем, мне не удалось «справиться» с двумя аспирантами, но исследования необходимо было продолжать, так как иначе, по крайней мере, для меня жизнь становилась бессмысленной, и надо было в России найти путь для выхода из этой, как многие считали, безнадежной ситуации.

Именно это, но несколько другими словами сказал ученый секретарь Совета при Президенте РФ по науке и высоким технологиям М. Ковальчук: «Большого финансирования науки наша экономика не потянет. Но выход есть. Нужно искать новые формы и

организацию научных исследований и внедрения технологий. Важным является также вопрос об интеграции науки и образования» («Российская газета», 11.02.2004).

С 1998 г. я начал преподавать на кафедре «Компьютерные технологии» Санкт-Петербургского государственного университета информационных технологий, механики и оптики (С-ПбГУ ИТМО), студентами которой являются одаренные в области информатики и программирования дети, отобранные по всей стране. За последние годы несколько команд наших студентов добились выдающихся результатов на студенческом чемпионате мира ACM [5].

В течение 5 лет я в традиционной форме преподавал этим студентам автоматное программирование: читал лекции, принимал экзамены и выполненные ими курсовые работы. Несмотря на все их таланты, мои призывы к ним о помощи в развитии предлагаемой технологии, например, применительно к объектно-ориентированному программированию, ни к чему не приводили: ребята просто отбывали номер – и все.

При традиционном преподавании жизненный цикл курсовых работ был также традиционным: сдача бумажной и электронной версий, хранение в течение некоторого времени в шкафу в надежде что из «этого»

удастся что-либо сделать и, наконец, завершающая фаза – выбрасывание работы на помойку ввиду невозможности доведения ее до «кондиции» кем-либо, кто не является ее автором.

При традиционном преподавании более современной формой является публикация указанных работ в сети Интернет, что и делается отдельными преподавателями, которые, однако, для того чтобы снять с себя ответственность за полученный результат, на сайтах пишут, что «работы как сделаны, так и выложены» (as is) [6].

Следовательно, что-то полезное при традиционном преподавании можно получить от ИТ-студентов, которые практически все работают полную (!) рабочую неделю, только случайно.

Для «серийного» написания достаточно качественных курсовых работ я пошел на педагогический эксперимент, который провожу уже три года.

При этом основное обучение осуществляется не на лекциях, а при личном контакте в ходе выполнения курсовых проектов, при создании каждого из которых **я должен был перестать жалеть свое время**, так как получаемый результат прямо пропорционален затраченным усилиям.

Какого результата можно добиться, если при традиционном

подходе преподаватель дает минутные консультации и за десяток минут принимает работу, сделанную студентом всего лишь за несколько часов?

В результате я стал работать по следующей схеме:

- студенты разбиваются на бригады, не превышающие двух человек, так как при большем их числе кто-то не работает;
- бригада предлагает тему работы, которую она должна выполнять на основе автоматного подхода;
- бригада записывается в очередь для обсуждения хода выполнения работы;
- каждая встреча с бригадой продолжается три с половиной часа, в ходе которой обсуждается не только разрабатываемая программа, но и проектная документация, создаваемая в ходе выполнения работы;
- после завершения встречи бригада вновь записывается в очередь, а на следующий день приходит другая бригада;
- встречи проходят ежедневно, за исключением воскресений и чрезвычайных обстоятельств;
- критерий выполнения проекта прост и нетрадиционен – он должен быть сделан так, чтобы за него не было стыдно;
- сделать работу так, чтобы за нее, по крайней мере, не было

стыдно, обычно меньше чем за три-четыре встречи не удается;

- разработанная программа и проектная документация к ней публикуются на сайте <http://is.ifmo.ru> в разделах «Проекты» и «Визуализаторы».

Таким образом, на каждую работу в среднем я стал тратить 12–15 часов, а студенты – не менее ста часов.

Многие из выполненных работ, ввиду их «товарного вида», опубликованы на дисках, являющихся приложениями к журналу «Мир ПК», тираж которого (около 50000 экземпляров) не типичен для публикации курсовых работ.

Некоторые работы, в которых предлагается то или иное развитие автоматного подхода, небольшими усилиями удается преобразовать в статьи для публикации сначала в журналах или трудах конференций, а затем на сайте <http://is.ifmo.ru> в разделе «Статьи».

Если раньше мне не удавалось «справиться» с двумя «учениками», то при такой организации труда я «справляюсь» с несколькими десятками студентов и аспирантов. Это достигается, во-первых, за счет административного ресурса, во-вторых, из-за нежелания идти в армию после окончания аспирантуры без защиты диссертации, а в-третьих, благодаря

медленно пробуждающемуся интересу к науке.

В результате работы «конвейера» я оказываюсь загружен на «полную катушку», а «ученики могут жить», встречаясь со мной не чаще чем раз в полтора месяца. При этом, правда, стали появляться «ученые», которые хотят встречаться со мной значительно чаще.

Для обобщения указанной деятельности мною была предложена «Новая инициатива в программировании. Движение за открытую проектную документацию» [7], что дополняет такие широко известные в мире направления создания ПО, как *Free Software Foundation* и *Open Source Initiative*. В рамках предлагаемой инициативы на указанном выше сайте опубликовано около 100 студенческих курсовых проектов, на которых отрабатывается технология автоматного программирования.

Изложенный подход во многом совпадает с системой ИТ-образования, принятой в *Массачусетском технологическом институте*, которую называют «академической базой тренировки морских пехотинцев». В учебном процессе этого института есть лекции, но они служат лишь в качестве пояснения к постановке задачи и концепции возможных ее решений. Для ее решения студенты собираются в неформальные группы, и работа в

них и есть основной элемент обучения. Главным инструментом служат так называемые «библии», которые являются своего рода базами знаний, коллекциями решений аналогичных задач, накопленных за многие годы. Изучая их, студенты одновременно пополняют их своими результатами» [8]. **Принципиальное отличие** этого и предлагаемого нами подхода состоит в доступности проектов в сети Интернет. При этом наш подход позволяет проводить обучение на «проектах» неограниченного круга лиц.

Таким образом, проводимый педагогический эксперимент решает **триединую задачу**: повышается качество обучения, проводится научная работа, совершенствуется технология автоматного программирования.

Изложенное позволило «продержаться» нам до тех пор, пока в компании *eDevelopers Corp.* мои ученики не разработали на языке *Java* инструментальное средство *UniMod* для поддержки автоматного программирования в нотации *UML* в виде плагина к среде разработки *Eclipse* [9], а корпорация *Borland* не открыла совместно с нашим университетом научную лабораторию для исследований в области создания технологий программирования, в том числе и автоматного [10].

Это является возвращением вуза к нормальной жизни, так как проведение научных исследований в нем – это

во многом именно то, что отличает вуз от техникума и курсов повышения квалификации (пусть даже называемых академиями). Центры подготовки специалистов в области высоких технологий должны быть одновременно и исследовательскими центрами [11]. Это соответствует проверенному временем «принципу Гумбольдта», в соответствии с которым настоящего специалиста университет может вырастить, лишь если в нем «учебное и научное» неразделимы [12].

Недостаточные научные исследования могут стать преградой для

развития программирования. Поэтому, например, исследовательское подразделение корпорации *Microsoft* открыло совместно с шестью японскими вузами институт, который будет проводить исследования, направленные, в частности, на применение искусственного интеллекта в программировании [13]. Кроме того, Билл Гейтс озабочен еще и тем, что «число студентов на факультетах, ведущих обучение инженерным специальностям и информационным технологиям, постоянно сокращается», но это уже другая история.

#### ЛИТЕРАТУРА

1. Шалыто А. А. Технология автоматного программирования // Мир ПК. 2003. – № 10. – С. 74–78. [http://is.ifmo.ru/works/tech\\_aut\\_prog](http://is.ifmo.ru/works/tech_aut_prog)
2. Шалыто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998. – 628 с. <http://is.ifmo.ru/books/switch/1>
3. Непейвода Н. Н. Стили и методы программирования. М.: Интернет-университет информационных технологий. 2005. – 316 с. <http://is.ifmo.ru/foundation/moving>
4. Туккель Н. И., Шалыто А. А. SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. 2001. – № 5. – С. 45–62. <http://is.ifmo.ru/works/switch/1/>
5. Богатырев Р. Нас не догонят? Триумф России и провал США. К итогам чемпионата мира 2005 г. по программированию // Мир ПК. 2005. № 5. – С. 60–67. <http://is.ifmo.ru/belletristic/acm2005.pdf>
6. Романовский И. В. Дискретный анализ. С-Пб.: Невский диалект, 2003. – 320 с.
7. Шалыто А. А. Новая инициатива в программировании. Движение за открытую проектную документацию // PC Week/RE. 2003. №40. – С. 38,39,42. [http://is.ifmo.ru/works/open\\_doc/](http://is.ifmo.ru/works/open_doc/)
8. Робертс П. Дом, который построил Гери // Computerworld/Россия. 2004. – № 22. – С. 38, 39.
9. Гуров В. С., Нарвский А. С., Шалыто А. А. Исполняемый UML из России // PC Week/RE. 2005. № 26. – С.18, 19. <http://is.ifmo.ru/works/umlrus.pdf>
10. Шалыто А. А. Корпорация Borland в Санкт-Петербурге: образование, наука, бизнес // IT manager. 2005.– № 3. – С. 88–91. <http://is.ifmo.ru/belletristic/borl/>
11. Велькович С. На вершине пирамиды // Computerworld/Россия. 2005. – № 26. – С. 22, 23.
12. Соснов А. Прав был Гумбольдт // Газета «Поиск». 2005. – № 25. – С. 4.
13. Калландер П. Курс на Восток // Computerworld/Россия. 2005. – № 26 – С. 8.