

решения систем одАНЛДУ [6]. Последняя составляет основу измерительной части программной системы Web-SynDic [7].

Библиографический список

1. Богоявленский Ю. А., Корзун Д. Ж. Общий вид решения системы линейных диофантовых уравнений, ассоциированной с контекстно-свободной грамматикой // Труды Петрозаводского государственного университета. Сер. «Прикладная математика и информатика». Вып. 6. Петрозаводск: Изд-во ПетрГУ, 1997. С. 79–94.
2. Корзун Д. Ж. Об одной взаимосвязи формальных грамматик и систем линейных диофантовых уравнений // Вестник молодых ученых. 2000. № 3. СПб: Изд-во СПбГТУ, 2000. С. 50–56.
3. Корзун Д. Ж. Grammar-Based Algorithms for Solving Certain Classes of Non-negative Linear Diophantine Systems // Труды международного семинара Finnish Data Processing Week at the University of Petrozavodsk (FDPW'2000): Advances in Methods of Modern Information Technology. Vol. 3. Петрозаводск: Изд-во ПетрГУ, 2001. С. 52–67.
4. Slopes solver. <http://www.ncc.up.pt/~apt/dioph/>
5. Кулаков К. А., Корзун Д. Ж. Generating Homogeneous Systems of Equations for Testing and Experimental Analysis of Linear Diophantine Solvers // Труды международного семинара Finnish Data Processing Week at the University of Petrozavodsk (FDPW'2003): Advances in Methods of Modern Information Technology. Vol. 5. Петрозаводск: Изд-во ПетрГУ, 2005. С. 259–278.
6. Кулаков К. А. Генерация систем однородных неотрицательных линейных диофантовых уравнений и ее приложения: Магистерская дис. Петрозаводск, 2005. 82 с.
<http://cs.karelia.ru/~kulakov/research/2005/work.pdf>
7. Богоявленский Ю.А., Корзун Д.Ж., Кулаков К.А., Крышень М.А. Проект Web-SynDic: Разработка программного обеспечения для удаленного решения линейных диофантовых уравнений в неотрицательных целых // Труды данной конференции.
8. Giles F., Pulleyblank W. Total dual integrality and integer polyedra // Linear algebra and its applications. No. 25, 1979. С. 191–196.
9. Схрейвер А. Теория линейного и целочисленного программирования: Пер. с англ. Т. 1. М.: Мир, 1991. 360 с.; Т. 2. М.: Мир, 1991. 342 с.
10. Корзун Д.Ж. Синтаксические алгоритмы решения неотрицательных линейных диофантовых уравнений и их приложение к моделированию структуры нагрузки канала Интернет: Дис. ... канд. физ.-мат. наук. Петрозаводск, 2002. 185 с.
11. Contejean E., Devie H. An Efficient algorithm for solving systems of linear diophantine equations // Information and computation. 1994. Vol. 113. No. 1. С. 143–172.
12. Корзун Д.Ж. О существовании порождающей КС-грамматики для произвольной линейной диофантовой системы // Труды Петрозаводского государственного университета. Сер. «Математика». Вып. 6. Петрозаводск: Изд-во ПетрГУ, 1999. С. 34–40.
13. Prabhaker Mateti, Narsingh Deo: On Algorithms for Enumerating All Circuits of a Graph // SIAM J. Comput. 1976. 5(1). С. 90–99.
14. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. СПб.: БХВ-Петербург, 2003. 1104 с.

ПРИМЕНЕНИЕ ЯЗЫКА ТЕРНАРНОГО ОПИСАНИЯ В ДВУЯЗЫЧНОМ МАШИННОМ ПЕРЕВОДЕ

О.И. Куцос

ОНУ им. Мечникова

Одесса, Украина

olga_kutsos@ukr.net

На современном этапе научного развития все большую актуальность приобретает рассмотрение лингвистических проблем в контексте современных информационных технологий, в частности в сфере межъязыкового компьютерного перевода. Во многом это связано с процессом информатизации общества, потребностью охвата максимального объема информации при минимальных усилиях. Разнообразные компьютерные программы – вспомогательный инструментарий как для профессионального перевода, в дальнейшем корректируемого, так и для бытового, призванного дать приблизительное представление о переводимом материале.

Двужызычный машинный перевод предполагает учет множества семантико-синтаксических особенностей. Рассмотрим, к примеру, соответствия между русским и английским языками. Так, синтаксически английский и русский языки более всего отличаются порядком слов. Если в английском предложении сначала употребляется подлежащее, потом сказуемое, следом за ними дополнение, обстоятельство места и обстоятельство времени, то в русском порядок слов будет инверсным – от второстепенных членов к главным. Кроме того, в каждом языке существуют синтаксические конструкции, характерные только для данного языка (например, английские конструкции с формальными подлежащими *there* и *it*). Во многих случаях английский язык более экономичен в средствах выражения мысли, чем русский.

Каким образом осуществляется машинный перевод? Сначала производится ввод текста и отождествление данной словоформы с определенной лексемой. Затем проверяется наличие идиом, фразеологизмов, омонимов, полисемических слов. После этого производится окончательный грамматический анализ и синтез выходных словоформ.

В результате такого перевода текст нередко оказывается искаженным: нарушен синтаксический порядок слов, смысловые связи, наблюдается грамматическая несогласованность. На наш взгляд, данная проблема решается при помощи включения в качестве вспомогательного механизма машинного перевода логического формализма, известного как язык тернарного описания (ЯТО).

ЯТО основан на тройке фундаментальных логических категорий – вещи, свойства и отношения. Другой, не менее важной тройкой, фигурирующей в ЯТО, являются категории определенного, неопределенного и произвольного. В этом преимущество ЯТО перед другими логическими формализмами, исходящими из более узкой категориальной базы: он более полно и адекватно отображает проявляющиеся в натуральном языке особенности мышления. Любая мысль может быть представлена как некоторое отношение между вещью и свойством, вещью и отношением, одновременно между вещью, свойством и отношением и т.д.

При всем многообразии лингвистических средств упомянутые логические категории присутствуют в любом человеческом языке. Таким образом, с помощью ЯТО может быть осуществлена компьютерная обработка текстов различной сложности, предусматривающая: 1) перевод входного языка в формулы ЯТО; 2) осуществление логического вывода; 3) обратный перевод на выходной язык.

Первые шаги в этом направлении уже сделаны – нами были проведены исследования, касающиеся взаимоотношения и сопоставления формул ЯТО, с одной стороны, и слов, словосочетаний и предложений русского языка – с другой. Таким образом, ЯТО может стать основой машинного перевода как с участием человека, так и (в перспективе) полностью автоматизированного.

ИСТОРИЯ РАЗВИТИЯ ОТЕЧЕСТВЕННОГО ПРОГРАММИРОВАНИЯ ДЛЯ СПЕЦИАЛИЗИРОВАННЫХ ЭВМ

В.В. Липаев

Институт системного программирования РАН

Москва, РФ

tom@ispras.ru

Основные особенности развития в стране специализированных вычислительных машин в 50–80-е годы XX века.

Особенности функциональных задач для специализированных ЭВМ реального времени.

Характеристики специализированных вычислительных средств, применяемых в системах реального времени.

Модели внешней среды на ЭВМ для отладки и испытаний комплексов программ реального времени.

Концепция ПРОМЕТЕЙ-технологии для обеспечения жизненного цикла комплексов программ реального времени:

- методы распределения вычислительных ресурсов и дисциплин диспетчеризации в объектных ЭВМ реального времени;
- методы построения кросс-систем автоматизации программирования для объектных ЭВМ реального времени;
- особенности системы отладки программных компонентов вне реального времени;
- особенности средств для динамической отладки и испытаний программ объектных ЭВМ в реальном времени.

Результаты применения ПРОМЕТЕЙ-технологии для разработки комплексов программ реального времени.

СЕРГЕЙ АЛЕКСЕЕВИЧ ЛЕБЕДЕВ – СОЗДАТЕЛЬ ПЕРВОЙ В КОНТИНЕНТАЛЬНОЙ ЕВРОПЕ И В СОВЕТСКОМ СОЮЗЕ ЦИФРОВОЙ ЭЛЕКТРОННОЙ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ (МЭСМ)

И.М. Лисовский

Институт точной механики и вычислительной техники им. С.А. Лебедева РАН

Москва, РФ

irina_lis@hotmail.com

До создания макета электронной счетной машины (МЭСМ) в г. Киеве осенью 1950 года Сергей Алексеевич ЛЕБЕДЕВ был уже известным ученым, академиком АН УССР, лауреатом Сталинской премии. Его научная деятельность началась в студенческие годы в МВТУ им. Н.Э. Баумана, где читали лекции ученые с мировым именем. В дипломном проекте, выполненном под руководством выдающегося ученого, основателя Всесоюзного электротехнического института им. В.И. Ленина (ВЭИ) К.А. Круга, Сергей Алексеевич разработал новую в то время проблему устойчивости параллельной работы электростанций.

В апреле 1928 года С.А. Лебедев получил диплом инженера-электрика и стал преподавателем МВТУ им. Н.Э. Баумана. В этом же году он был зачислен младшим научным сотрудником ВЭИ, где вскоре возглавил группу, а затем и лабораторию электрических сетей.

В 1931 году завершалось сооружение 30 районных электростанций по плану ГОЭЛРО, которым предусматривалось создание единой энергосистемы европейской части СССР и соединение ее в дальнейшем с энергосистемами Сибири и других районов.

В апреле 1931 года на Всесоюзной конференции по электропередаче больших мощностей на дальние расстояния токами сверхвысоких напряжений в числе основных докладчиков выступил С.А. Лебедев с докладом «Устойчивость параллельной работы крупных электрических станций», который был опубликован в журнале «Электричество». Необходимо отметить, что значение проблемы устойчивости при передаче энергии на большие расстояния очень велико. Можно утверждать, что устойчивость систем является основным фактором, ограничивающим дальность передачи энергии переменным током.

В других публикациях этого периода Сергей Алексеевич излагал теорию статической и динамической устойчивости многомашинных энергосистем при больших возмущениях и переходных процессах, а также методики инженерных расчетов устойчивости и способы ее повышения. Особо следует отметить большую статью, написанную совместно с А.И. Колпаковой, «Кустование электрических станций и создание Единой высоковольтной сети», которая вошла в коллективный многотомный труд «Генеральный план электрификации СССР».

В монографии «Устойчивость параллельной работы электрических систем», написанной в соавторстве с П.С. Ждановым, излагались теория, методы расчета и способы повышения устойчивости энергосистем. В то время не было работы в мировой научной литературе так полно освещавшей проблему устойчивости энергосистем. Во втором издании в монографии уделялось большее внимание методике расчетов динамической устойчивости и были значительно расширены главы книги, посвященные сложной теории переходных процессов в синхронных машинах. На протяжении многих лет эта книга широко использовалась в научно-исследовательских, проектных, производственных организациях, а также в качестве учебного пособия для вузов.

Решением Высшей аттестационной комиссии от 23 октября 1935 г. (протокол № 34 / 100) гражданин ЛЕБЕДЕВ Сергей Алексеевич утвержден в ученом звании ПРОФЕССОРА по кафедре «Электрические станции и сети».

Профессор Лебедев С.А. продолжает преподавать в МВТУ и заниматься исследованиями в ВЭИ, в котором была достаточно мощная производственная база, позволявшая довольно быстро внедрять в практику результаты исследований. Под его руководством и при непосредственном участии в ВЭИ были разработаны, изготовлены и введены в эксплуатацию в тресте «Теплоэлектропроект» (Москва) и энергосистеме «Уралэнерго» (Свердловск) модели оригинальной конструкции и с высокой степенью автоматизации расчетов, которые рассмотрены в статье С.А. Лебедева «Модель сетей переменного тока системы ВЭИ». Это направление моделирования широко развивалось в СССР и за рубежом вплоть до появления цифровых электронных вычислительных машин.

Фундаментальное исследование С.А. Лебедева по обоснованию возможности искусственной устойчивости электропередач и энергосистем на основе новых электронных автоматических регуляторов напряжения представляло особый научный и практический интерес. Эта работа составила содержание его докторской диссертации, которую он защитил в 1939 году.

Работая с моделями электрической системы переменного тока и электронными регуляторами напряжения, Сергей Алексеевич все чаще останавливается на изучении возможностей ламповых схем с двумя устойчивыми состояниями. В 1939 году он предлагает молодому сотруднику лаборатории магнитной дефектоскопии Секции электросвязи АН СССР, перед которой стояла задача обнаружения дефектов в сварных швах железнодорожных рельсов, использовать счетчик импульсов на электронной схеме с двумя устойчивыми состояниями. Этот молодой сотрудник Анатолий Владимирович Нетушил успешно справился с

задачей и защитил кандидатскую диссертацию на тему «Анализ триггерных элементов быстродействующих счетчиков импульсов», оппонентом по которой был С.А. Лебедев.

Возраставший интерес к электронным триггерам и двоичной системе счисления подтверждает воспоминание жены Сергея Алексеевича Алисы Григорьевны Лебедевой о том, как в первые дни войны по вечерам, когда Москва погружалась в темноту, Сергей Алексеевич уходил в ванную комнату и там при свете газовой горелки писал непонятные ей единицы и нолики. Со временем электронные триггеры и двоичная система счисления станут основой вычислительной техники.

Если бы не война, работу над созданием цифровой электронной вычислительной машины с использованием двоичной системы счисления Сергей Алексеевич начал бы раньше, об этом впоследствии он говорил сам. Война изменила направленность научной и исследовательской деятельности Сергея Алексеевича и поставила перед ним в конце 1941 года задачи оборонной тематики.

В октябре 1941 года вместе с ВЭИ Сергей Алексеевич эвакуировался в Свердловск. Здесь он в удивительно короткие сроки разработал быстро принятую на вооружение Красной Армии систему стабилизации танкового орудия, которая позволяла наводить и стрелять без остановки танка, что сделало танк менее уязвимым.

Второй крупной работой в Свердловске было создание летательного аппарата (торпеды) с головкой самонаведения (совместно с доктором техн. наук Д.В. Свечарником). Продувка моделей торпеды производилась в г. Жуковском в 1944 году, когда ВЭИ вернулся в Москву. Натурные испытания торпеды состоялись в 1945–1946 годах на Черном море. В октябре 1946 года в Евпатории торпеда прямым попаданием поразила баржу. Оценки комиссии были самыми высокими. Началось создание сверхточного оружия, которое появилось в США значительно позже.

Для систем стабилизации танковой пушки и автоматического устройства самонаведения на цель авиационной торпеды оборонной промышленностью были разработаны и изготовлены аналоговые вычислительные элементы. На них Сергей Алексеевич в 1945 году создал первую в Советском Союзе электронную аналоговую вычислительную машину для решения систем обыкновенных дифференциальных уравнений, которые часто встречаются в задачах, связанных с энергетикой.

С начала 40-х годов на путях автоматизации умственного труда жизнь поставила особенно сложные задачи в области регулирования и управления процессами. Резко возросли требования оборонной промышленности и в целом народного хозяйства. Увеличился объем и сложность математических и логических задач, которые необходимо было решать в максимально короткое время.

Нужны были новые подходы, новые идеи. Они овладевали С.А. Лебедевым с начала 40-х, но реализовываться стали значительно позже.

В 1945 году президент Академии наук Украины академик А.А. Богомолец предложил С.А. Лебедеву баллотироваться в академики и занять должность директора Института энергетики АН Украины с условием переезда в г. Киев.

Вероятнее всего, кандидатуру Сергея Алексеевича предложил академику А.А. Богомольцу близкий к нему Л.В. Цукерник, который хорошо знал научные труды С.А. Лебедева, работал с ним в ВЭИ и в то время был директором Института энергетики АН УССР. (Позже он дарил С.А. Лебедеву свои публикации в области устойчивости энергосистем, сопровождая их, например, такими надписями: «Моему замечательному учителю и вдохновителю, дорогому Сергею Алексеевичу Лебедеву – на добрую память. Л. Цукерник».)

В сентябре 1945 года Академия наук Украины избрала С.А. Лебедева академиком. В 1946 году семья Лебедевых переехала в г. Киев. Сергей Алексеевич стал директором Института электротехники, вторая половина Института энергетики стала называться Институтом теплоэнергетики. Сергей Алексеевич к су-

шествующим лабораториям электротехнического профиля добавляет лабораторию моделирования и регулирования, продолжая совместно с лабораторией Л.В. Цукерника работы по исследованию в области технических средств стабилизации энергосистем и устройств автоматики.

За разработку и внедрение устройства компаундирования генераторов электростанций для повышения устойчивости энергосистем и улучшения работы электроустановок Лебедеву С.А. и Цукернику Л.В. в 1950 году была присуждена Сталинская премия.

Что стало толчком к началу практических шагов по развертыванию работ в совершенно новом направлении науки и техники? Вероятнее всего, завершенность обдумывания идеи, исключительная эрудированность, предыдущий опыт и научное бесстрашие Сергея Алексеевича.

К концу 1947 года идея облачилась в четкие формы структурной схемы цифровой вычислительной машины, основных электронных схем, выбора элементной базы, временных диаграмм, представления чисел, количества двоичных разрядов, системы команд и состава операций. Сергею Алексеевичу уже было чем делиться с научной общественностью.

В январе – марте 1948 года он собирает на семинаре в Институте электротехники АН УССР видных ученых академиков М.А. Лаврентьева, А.А. Дородницына, Б.В. Гнеденко, А.Ю. Ишлинского, чл.-корр. АН Украины А.А. Харкевича, других математиков и физиков научно-исследовательских институтов, где ставит на обсуждение результаты своей многолетней изыскательской деятельности.

На семинаре обсудили состав и характер задач, структурную схему и ее характеристики, выбрали форму представления числа с фиксированной запятой перед старшим разрядом, что существенно сокращало сроки разработки, приняли решение строить машину на 17 двоичных разрядах, включая разряд для знака числа и необходимость предусмотреть в конструкции возможность добавления нескольких разрядов. Трехадресная система команд была принята почти без обсуждения. Команда, которую нужно выполнить, должна состоять из кода операции: сложить, вычесть, умножить, разделить, сложить и вычесть модули числа, т.е. без учета их знака, сравнить, сдвинуть и т.д., всего 13 различных операций; кода адреса в запоминающем устройстве (ЗУ) первого числа: слагаемого, множителя, делителя... второго числа: второго слагаемого, второго множителя, второго делителя... а также кода адреса ячейки ЗУ, куда необходимо направить результаты вычислений. Электроника должна была обеспечить быстроедействие 50–100 операций в секунду, а программное управление – автоматизацию счета. Причем программу можно было бы усложнять, добиваясь максимального упрощения электронных схем.

Идея начала овладевать массами. История показывает, что новые идеи, которые изменяют мир, выдвигаются одиночками, а затем уже приходят организаторы науки, коллективы, предприятия, армия чиновников и потребителей.

Результаты своих изысканий и обсуждений на семинаре С.А. Лебедев, по рекомендации вице-президента Академии наук Украины М.А. Лаврентьева, доложил Президиуму АН УССР и начальнику отдела ЦК Компартии Украины. Вскоре последовало решение Института электротехники АН УССР в самые короткие сроки разработать и сдать в эксплуатацию электронную цифровую вычислительную машину, для создания которой было выделено полуразрушенное здание в поселке Феофания (в 15 км от г. Киева).

Сергей Алексеевич в середине 1949 года сформировал из сотрудников Института электротехники коллектив для работы в Феофании. Двум «остепененным» товарищам: заместителю заведующего лабораторией Дашевскому Л.Н. и секретарю партийной группы лаборатории Шкабара Е.А. были поручены организационные заботы, обеспечение материалами и комплектующими изделиями, заключение договоров, организация социалистического соревнования и пр.

Молодые специалисты – Абальшникова Л.М., Беляев М.М., Ботвиновская Е.Ф., Гладыш А.Л., Дашевская А.А., Дедешко Е.Е., Заика В.А., Кондалев А.И., Крайницкий В.В., Лисовский И.М., Михайленко Н.А., Мозыра Ю.С., Окулова И.П., Пархоменко И.Т., Пещух Т.И., Пиневиц М.М., Погребинский С.Б., Рабинович З.Л., Рапота З.С., Семеновский А.Г., Фурман Н.И., Черняк Р.Я. – образовали группы по названиям устройств. Арифметическую группу возглавил Пиневиц М.М., центрального управления – Гладыш А.Л., электронного запоминания – Погребинский С.Б. (с марта 1950 – Лисовский И.М.), электриков – Беляев М.М., конструкторскую – Крайницкий В.В., аспирантов – Рабинович З.Л. Группа аспирантов под руководством Сергея Алексеевича разрабатывала систему моделирования автопилотов летательных аппаратов.

Настал день и на прикомандированном автобусе мы едем, большей частью по булыжной дороге, смотреть новое место работы. Слева – лес, справа – поле. Въехали в прекрасный лес, изрытый заросшими травой воронками, оставшимися после войны. Посреди леса – монастырь, к нему ведет липовая аллея, рядом монастырские постройки, сад, озеро. Предложенное нам двухэтажное полуразрушенное здание находилось рядом с монастырем. Начали его восстанавливать, оборудовали помещения, спортивную площадку.

Физическую работу сочетали с умственной – начали исполнительскую работу по реализации идеи, мечты Сергея Алексеевича Лебедева. Имея единственное печатное пособие – Г. Дж. Рейх «Теория и применения электронных приборов» – приступили к исследованиям методов цифровой обработки информации. После выбора доступной тогда элементной базы началось исследование ламповых импульсно-потенциальных элементов и принципиальных схем. Перед тем как зарегистрировать отработанную схему в журнале пригодных для использования, Сергей Алексеевич проверял на стенде каждый вариант. Первые схемы были громоздкими, перенасыщенными комплектующими изделиями. За упрощение схем Сергей Алексеевич ввел специальный коэффициент, который учитывался при начислении премии.

Много времени было потрачено на начальный период проектирования. Хорошо работающие схемы удавались с большим трудом. С возникавшими вопросами все обращались к Сергею Алексеевичу. Он был очень терпелив, внимателен к каждому исследователю, часто садился рядом то с одним, то с другим, брал паяльник, сдвигал на лоб очки и начинал работать над схемой. Помнил хорошо не только основные схемы, но и все их варианты. Если долго не добивался требуемого результата, уходил в лес, на любимый пенек, и там, вычерчивая отдельные элементы схемы, временные диаграммы и производя расчеты часто только на пачках папирос, находил ошибки и радостный возвращался к прерванной работе.

Большая занятость в Феофании не позволяла Сергею Алексеевичу уделять много внимания общеинститутским проблемам. В этом ему оказывали помощь заместитель по науке Л.В. Цукерник и секретарь парторганизации института И.В. Акаловский.

Нельзя переоценить участие академика М.А. Лаврентьева с самых начальных этапов развертывания работ по созданию МЭСМ. Благодаря его усилиям в Феофании было передано здание для нашей лаборатории. Рядом с этим зданием в монастыре располагался отдел Института математики АН УССР, директором института был М.А. Лаврентьев.

Сотрудники отдела на полигоне, недалеко от дома, где жил во время пребывания в Феофании М.А. Лаврентьев, проводили экспериментальные взрывы, а в свободное время играли в волейбол на нашей площадке.

М.А. Лаврентьев – известный ученый-математик, академик, доктор технических и физико-математических наук (имел два диплома), дважды лауреат Сталинской премии. Во время Отечественной войны работал, впервые в мире, теорию коммуляции при взрывах, что позволило создать эффективные противотанковые снаряды и мины. Решил ряд других важных проблем для отечественной артиллерии. Пользовался в своих математических исследованиях моделями на аналоговой аппаратуре и потому, может быть, первым

понял, познакоившись с работами по МЭСМ, какие неограниченные возможности для моделирования появляются на электронной цифровой машине с автоматическим программным управлением. Академик М.В. Келдыш, директор Института прикладной математики АН СССР, даже будучи председателем приемной комиссии на испытаниях МЭСМ, эту находку для математиков недооценивал и был против того, чтобы С.А. Лебедев избирался в академики АН СССР по отделению физико-математических наук.

В Феофании М.А. Лаврентьев часто навещал С.А. Лебедева, подолгу с ним беседовал, интересовался нашими успехами.

Постановлением Президиума АН СССР от 15 марта 1950 года академик М.А. Лаврентьев был назначен директором Института точной механики и вычислительной техники (ИТМ и ВТ) АН СССР. Приказом № 40 от 20 марта 1950 года академик М.А. Лаврентьев назначил заведующим лабораторией № 1 ИТМ и ВТ действительного члена АН УССР, профессора, доктора технических наук С.А. Лебедева. Ряд сотрудников, работавших в Феофании, был зачислен в штат лаборатории № 1 ИТМ и ВТ (приказ № 53 от 1 апреля 1950 года) и откомандирован в Институт электротехники АН УССР (г. Киев, улица Чкалова, дом 556).

Для обеспечения работ, ведущихся группой сотрудников лаборатории № 1 в Феофании, академик М.А. Лаврентьев разрешил передачу из Москвы в Киев необходимых приборов и оборудования, а также материалов для ведения НИР. Приказом по ИТМ и ВТ № 73 от 8 мая 1950 года материально-ответственной была назначена А.Л. Гладыш. Таким образом, в Феофании появились две группы сотрудников – киевская и московская. В московскую группу входили: М.А. Беляев, А.Л. Гладыш, И.М. Лисовский, Ю.С. Мозыра, П.М. Остапчук, Т.И. Пецух, С.В. Пегребинский, З.С. Рапота, А.Г. Семеновский.

Первые же успехи в автономной отладке вызывали бурную радость, взаимные поздравления и веселье, во время которых тратились все сбережения общественной кассы, накопившиеся в виде штрафов и вознаграждений за удачные и неудачные короткие устные рассказы в автобусных поездках Киев – Феофания – Киев. Все путешествие при хорошей погоде длилось около часа, в течение которого автобус сотрясаясь от смеха. Основные остряки не давали пощады никому, но насмешек не допускали, понимали, что тех слов, где есть хоть капля яда, и в шутку говорить не надо. Шутки не повторялись, хороший безобидный смех сближал, возбуждал энергию и бодрость, лица светлели. Однако были и такие, которые относились к юмору и сарказму отрицательно, они же запрещали произносить слова «кибернетика», «триггер», «вольтова дуга» и т.п.

После первых успехов начали появляться гости, в частности будущий главный конструктор ЭВМ «Стрела» Ю.Я. Базилевский со своими сотрудниками. Они особенно внимательно обо всем расспрашивали, все рассматривали, записывали. Сергей Алексеевич подолгу объяснял им работу отдельных узлов, показывал опытные образцы пальчиковых ламп, ртутных колб, оксиферов, на которых мы собирали схемы памяти. Затем началось целое паломничество ученых, инженеров, сотрудников ЦК Компартии и Совета Министров Украины. Заинтересовался новыми возможностями в области вычислительной техники и 1-й секретарь ЦК КП Украины Н.С. Хрущев. В 1950 году Феофанию посетил секретарь ЦК Компартии Украины И.Д. Назаренко. Свое впечатление от увиденного он выразил одним словом – «колдовство». После его отъезда Сергей Алексеевич сказал: «Этот статный красавец, служитель религии, награжден природой величавостью, чтобы скрыть недостатки ума». Вообще к партийным функционерам Сергей Алексеевич относился трепетно. Все понимали, какую большую роль играла Коммунистическая партия в стране и в судьбе каждого человека в отдельности.

С самого начала работ в Феофании Сергей Алексеевич рассматривал МЭСМ как макет, на котором необходимо в самом широком диапазоне провести испытания и результаты учесть в будущей быстродействующей машине. В короткой записке в Совет по координации Академии наук СССР Сергей Алексеевич пи-

сал: «Быстродействующими электронными счетными машинами я начал заниматься в конце 1948 года. В 1948–1949 годах мной были разработаны основные принципы построения подобных машин. Учитывая их исключительное значение для нашего народного хозяйства, а также отсутствие в Союзе какого-либо опыта их постройки и эксплуатации, я принял решение как можно быстрее создать малую электронную счетную машину, на которой можно было бы исследовать основные принципы построения, проверить методику решения отдельных задач и накопить эксплуатационный опыт. В связи с этим было намечено первоначально создать действующий макет машины с последующим его переводом в малую электронную счетную машину. Чтобы не задерживать разработку, запоминающее устройство пришлось выполнить на триггерных ячейках, что ограничило его емкость. Разработка основных элементов была проведена в 1949 году. <...> К концу 1949 года были разработаны общая компоновка машины и принципиальные схемы ее блоков. В первой половине 1950 года изготовлены отдельные блоки и приступили к их отладке во взаимосвязи...»

Отладка блоков проходила при остром недостатке измерительных приборов и трудностях согласования вопросов импульсных связей и взаимодействий. В МЭСМ был заложен синхронный принцип работы. Все ее устройства, каждое из которых выполняло свои самостоятельные операции, должно было работать строго синхронно. Малейшие расхождения во временных циклах не допускались. Обеспечивалась такая работа при помощи синхронизирующих импульсов.

Облегчение наступило только после установки блоков в сварной общий каркас машины, завершения межблочных соединений и ввода пульта управления. На нем была изображена мнемоническая схема, расположены измерительные приборы, переключатели и ключи управления. Для испытаний и наладки машины, кроме автоматического и полуавтоматического режимов, был предусмотрен ручной ввод импульсов сдвига на электронные запоминающие устройства чисел и команд. По сигнальным неоновым лампам на пульте управления можно было проследить работу всех элементов машины.

Как только завершилась автономная отладка блоков и устройств МЭСМ, все материалы с ее результатами были уложены в большой кожаный портфель, с которым Сергей Алексеевич уехал на короткий срок в Москву. Следом понеслись телеграммы с просьбой о немедленном возвращении, так как неожиданно остановилась комплексная отладка. Не у кого было спросить совета. Каждый хорошо знал свои блоки и устройства. Только что назначенный начальником МЭСМ Лисовский не сумел без Сергея Алексеевича обеспечить дальнейшее продвижение настройки. Это был самый ответственный и сложный этап работы, на котором также возникало много трудностей. Как и на предыдущих этапах создания МЭСМ, большие затруднения разрешал только Сергей Алексеевич, находя необходимые технические решения.

Для МЭСМ была установлена круглосуточная работа, для обслуживающего персонала – трехсменная, но поскольку его было недостаточно, приходилось работать по две смены. К этому времени в Феофании были организованы питание и ночлег. Сергей Алексеевич ежедневно работал за полночь, часто забывал о еде. Все реже отвлекался на аналоговое моделирование, проводившееся в соседней с машинным залом комнате. Наступило время, когда аппаратура, спроектированная на максимальные возможности радиоламп, и люди работали в самых предельных режимах. Социалистическое соревнование стало бессмысленным.

Много трудностей было преодолено на этапе комплексной отладки МЭСМ до полного подтверждения практикой правильности выбранных схем и их реализации. Впервые особенно обрадовались, наблюдая устойчивую работу универсального арифметического устройства и сумматора с цепочкой сквозных переносов, в котором реализовывались все арифметические и логические операции. Больше всего радовало, конечно, то, что все было новое, оригинальное. Опыта – никакого.

Об ЭВМ ЭНИАК и основных принципах Дж. Неймана по построению ЭЦВМ с хранимой программой узнали только в 50-х годах, когда появились рекламные публикации. Может быть, отсутствие этой информированности положительно отразилось на выбранном пути создания дискретных электронных машин?

Сергей Алексеевич самолично и так глубоко и всесторонне проработал основные принципы, структуру и технические решения, что в дальнейшем не потребовалось вносить сколько-нибудь значительные коррективы и дополнения.

В августе 1950 года А.А. Ляпунов составил первую программу для вычисления факториала числа. МЭСМ безукоризненно ее выполнила. Дальше усложнение программ и задач пошло быстрыми темпами. А.А. Ляпунов говорил, что за три месяца работы на МЭСМ он получил колоссальный опыт программирования, машинных методов реализации алгоритмов и цифрового моделирования.

Сергей Алексеевич второй раз собрал все материалы по теперь уже функционирующей машине в свой большой светло-коричневый портфель и повез их в ИТМ и ВТ АН СССР, где им было развернуто проектирование большой электронной счетной машины (БЭСМ).

6 ноября 1950 года Сергей Алексеевич устроил в Феофании первый показ широкому кругу специалистов в разных областях науки и техники первой отечественной электронной цифровой машины, на которой решались тестовые и простейшие вычислительные задачи.

После увеличения количества блоков запоминания для расширения емкости запоминающего устройства, отработки операций сложения и вычитания, умножения и сравнения МЭСМ была представлена приемной комиссии в составе ученых: Н.Н. Доброхотова, А.Ю. Ишлинского, С.Г. Крейна, С.А. Лебедева, Ф.Д. Овчаренко, И.Т. Швеца. В акте от 5 января 1951 года комиссия зафиксировала окончание в 1950 году разработки, изготовления и наладки МЭСМ и рекомендовала провести в дальнейшем ряд усовершенствований.

До сентября 1951 года реализовывались рекомендации приемной комиссии и проводились ранее запланированные Сергеем Алексеевичем работы по усовершенствованию МЭСМ.

Была разработана и введена в эксплуатацию система постоянных чисел и команд. Введена фотографическая запись результатов. Приспособлена сортировка с перфокартами для ввода исходных данных в машину. Разработана система управления магнитным запоминанием, завершён монтаж и отладка управления. Макет превращается в малую электронную счетную машину, и работа ее демонстрируется правительственной комиссией и комиссии экспертов.

1 июля 1951 года за № 2759-1321 выходит Постановление Правительства, обязывающее ввести в эксплуатацию электронную (малую) вычислительную машину в четвертом квартале 1951 года.

В сентябре 1951 года МЭСМ стала решать более сложные задачи со множеством алгебраических и дифференциальных уравнений в частных производных с сотнями неизвестных.

На ней стало возможным моделирование в самых разных областях науки и техники, что на аналоговых машинах было недостижимо. МЭСМ могла обеспечивать автоматическое управление процессами и реальными объектами.

25 декабря 1951 года МЭСМ была принята комиссией Академии наук СССР (председатель – академик М.В. Келдыш) и передана в эксплуатацию.

Появление МЭСМ послужило мощным толчком к постановке и решению актуальных задач вычислительной математики в области ядерной физики, ракетной баллистики, расчета линий электропередач Кулибышев – Москва и др. Решение вручную этих задач надолго задержало бы развитие важных направлений отечественной науки и техники.

Сергей Алексеевич Лебедев реализовал свою многолетнюю мечту: создал цифровую электронную счетную машину, помогал ему в этом небольшой коллектив сотрудников Института электротехники АН Украины и Института точной механики и вычислительной техники АН СССР.

Безусловно, эта работа по созданию МЭСМ заслужила самых высоких оценок, почетных званий, премий, наград, и она была представлена Академией наук Украины (Постановление Президиума № 12 от 11 января 1952 года) с необходимыми материалами на соискание Сталинской премии. В список по традиции были внесены: ученый, организатор, партийный функционер, но не было рабочего. В Феофании инженеры и техники выполняли не только функциональные обязанности, но и всю физическую работу. По непонятным причинам премия не была присуждена. Может быть, комиссия, рассматривавшая представление, не смогла оценить научного прорыва или побоялась того, что все напоминало о заклеенной «лженауке кибернетике»: реализация логических операций, возможность электронного цифрового моделирования в любых областях науки и техники, управление процессами и объектами. Может быть, по другим причинам, но премий и наград никто не получил. Да и благодарности вынесли очень скупое: нельзя было выделять кого-то, весь коллектив работал одинаково самоотверженно и все (всего-то 22 человека) заслуживали в равной мере ПООЩРЕНИЙ.

Хотя Сергей Алексеевич сразу выбрал двоичную систему счета и ламповые триггеры как элементы памяти для МЭСМ, в Феофании с самого начала были развернуты в комнате запоминающих устройств научно-исследовательские работы (НИР), где велись исследования других систем счета и других схем запоминающих устройств. Инженеры группы ЗУ создали испытательные стенды, а на отдельном столике установили счетчик импульсов, с которого Сергей Алексеевич начинал демонстрацию гостям возможностей цифровых ламповых устройств. Счетчик нас никогда не подводил, в отличие от основной аппаратуры МЭСМ, где иногда случались эффекты.

Все работы группы фиксировались в аппаратном журнале: положительные и отрицательные стороны других систем счета по сравнению с двоичной (особенно 8-ричной и 16-ричной); количество смонтированных ячеек памяти за смену; количество отбракованных ламп, которые подбирались по примерно одинаковым коэффициентам усиления двоянных триодов 6Н8, а затем 6Н15; количество ламп, которые вышли из строя сразу при испытании их в стенде; замечания по параметрам комплектующих изделий и изменения, вносимые в принятые схемы. Журнал контролировал и обрабатывал Сергей Алексеевич.

В ЗУ использовалось примерно 75% всех ламп, содержащихся в МЭСМ. Все ячейки ЗУ чисел и команд были смонтированы и установлены в МЭСМ инженерами группы ЗУ. В схемах триггеров использовались лампы 6Н15, не требовавшие подбора по коэффициентам усиления.

Центральный коммутатор на 6Х6 тоже был смонтирован в группе ЗУ. Самой надежной была в то время 6У6, которая использовалась в катодных повторителях и усилителях. Ее наработка на отказ превышала 500 часов.

На стендах мы учились с максимальной скоростью обнаруживать неисправности. Сергей Алексеевич требовал научиться устранять отказы быстрее, чем они возникали. Мы старались и устанавливали рекорды. Благодаря этой выучке на аппаратуре МЭСМ была обеспечена довольно приемлемая устойчивая зона работоспособности. Самым слабым звеном в триггере были 2-ваттные сопротивления типа ВС в анодной цепи, которые часто горели.

Сергею Алексеевичу не нравилось ламповое ЗУ, и мы искали ему замену на самых неожиданных направлениях: от бумажного, которое усиленно предлагал Л.И. Гутенмахер, до магнитофона и магнитного барабана разработки Института физики АН Украины. Впервые в августе 1949 года мы посетили Институт физики АН УССР и познакомились с нашими сверстниками – младшими научными сотрудниками Власовой

Евгенией, Гофманом Владимиром, Акопьян Гретой и техником «золотые руки» Шулейко Михаилом, которые под руководством заведующего отделом технической физики чл.-корр. Академии наук Украины Александра Александровича Харкевича (впоследствии академика АН СССР) преодолевали трудности магнитной записи на ферромагнитную пленку плохо поддающимися установке магнитными головками по образующей барабана. В кабинете А.А. Харкевича Сергей Алексеевич, завершивший в начале 1949 года монографию «Малая электронная счетная машина АН УССР» (издана в 1952 году), рассказал молодым энтузиастам магнитной записи о цели нашего посещения и о двоичной системе счисления, о которой они не имели ни малейшего представления. Рассказ сопровождался демонстрацией арифметических операций в двоичных кодах по рукописи монографии.

Сразу в группе по умным вопросам, которые свидетельствовали о быстро схваченном существе и понимании требующейся переориентировке, выделился лидер – Власова Е.М. Кроме того, она была еще и очень красивой, что, естественно, вызывало желание чаще посещать Институт физики и более подробно излагать наши требования к запоминающим устройствам на магнитном барабане: постоянству амплитуды считываемых сигналов, их синхронности; точности статической и динамической балансировки; минимальному биению поверхности; однородности, прочности и долговечности магнитного покрытия и т.п. Лишь бы продлить приятное собеседование.

Срок завершения работ по договору о создании ЗУ на магнитном барабане с Институтом физики истек в 1950 году. После испытаний барабана в январе 1951 года актом сдачи-приемки было зафиксировано завершение разработки. В Феофании барабан был установлен в машинном зале МЭСМ и введен в состав машины в августе 1951 года. Его отладкой кроме названных участников разработки вместе с нашими инженерами с конца 1951-года занимался только что закончивший Киевский политехнический институт сотрудник Института физики Р.Г. Офенгенден. ЗУ на магнитном барабане было лучше ЗУ на магнитофоне тем, что обладало большой универсальностью и значительно меньшим временем обращения.

Московский институт автоматики предложил нам испытать ртутные колбы в качестве линий задержки. Тяжелые сооружения в термостатах разместили в полуподвальном помещении, где были сосредоточены агрегаты электропитания. Связи с электронными блоками осуществлялись экранированным кабелем. Кодовые последовательности циркулировали в ультразвуковой ртутной среде. Каждый тракт имел входной и выходной пьезопреобразователь. Среднее время выборки составляло около 300 мкс.

ОЗУ на ртутных колбах имело ряд преимуществ перед ламповым, но громоздкость, высокие требования к условиям окружающей среды, низкая надежность пьезоэлементов не вдохновляли.

Интересными в этих поисках были ЗУ на электронных лучевых трубках, которые нам поставил НИИ-160 из подмосковного Фрязино. Одна трубка для каждого двоичного разряда. Возрастала емкость оперативной памяти, но сигнал считывания иногда зашумлялся. Необходимо было предпринимать специальные меры: регулировать уровень считываемого сигнала. Время выборки составляло около 12 мкс.

Как-то А.А. Харкевич привез нам несколько десятков германиевых диодов, которые мы сразу пустили в дело. У них оказалось малое прямое сопротивление и большое отношение обратного сопротивления к прямому. Это позволило улучшить характеристики электронных схем там, где можно было ламповые диоды заменить полупроводниками.

Не помню, когда мы занялись швейным делом. У нас появились оксиферные сердечники, которые использовали связисты, и ферритовые сердечники из Ленинграда. В срочном порядке всей группой мы стали нанизывать их на шины с порядочной долей скептицизма. Результаты испытаний полученных оксиферных и ферритовых ЗУ не были плохими, но мы недоумевали, почему так шумят и рассыпаются магнитные

кольца. Попросили разрешить наши затруднения самого большого тогда авторитета в Союзе по помехоустойчивости Александра Александровича Харкевича. Он очень внимательно на осциллографе изучал шумовое поле и помехи, создаваемые в ферритовых ЗУ, и предположил, что процесс записи единицы начинается раньше (или выше по графику остаточной индукции), при этом разрушается ноль за счет помех и при его считывании возникающий большой сигнал помех может перегревать магнитные сердечники и разрушать их. С этим можно было бороться. Но Сергея Алексеевича смущала большая трудоемкость создания ЗУ на оксиферовых и ферритовых сердечниках, очень низкая ремонтпригодность, большой разброс величин снимаемых сигналов, сильное шумовое поле и, может быть, наше недоверчивое отношение к этому новшеству. Дальнейшие поиски в этом направлении были остановлены.

Однажды Сергей Алексеевич набросал в нашем журнале электрическую схему на триоде с трансформатором с 6 обмотками, 3 полупроводниковыми диодами, запоминающей емкостью, 2 сопротивлениями и сказал: «Лампу можешь выбрать любую, а лучше разные, трансформатор намотай, диоды, емкость, сопротивления подбери и схему испытай, результаты я посмотрю». Это была самая интересная схема триггера, которую мне пришлось собирать и испытывать. Она имела следующие преимущества перед схемой триггера с потенциальными связями: простота, высокое входное сопротивление, очень низкое выходное, что позволяет легко согласовывать с простыми логическими схемами на диодах; отсутствие жестких требований к разбросу параметров лампы, а также к разбросу величин параметров деталей; большая скважность; малые затраты энергии питания; большая крутизна фронта и спада и, как следствие, большое скородействие схем динамического триггера. Отличная схема, но комплексная отладка МЭСМ не оставляла больше времени на дальнейшие НИР, а также на переделки в собранных и функционирующих схемах.

Сергей Алексеевич решил, что мы выполнили приказ директора ИТМиВТ № 73 от 8 мая 1950 года о ведении НИР в Феофании.

На специальном ужине в феврале 1951 года Сергей Алексеевич отметил завершение разработки магнитного барабана для ЗУ и подписание 5 января приемной комиссией акта об окончании разработки, изготовления и наладки МЭСМ в декабре 1950 года и от имени всего коллектива поблагодарил Михаила Алексеевича Лаврентьева, Александра Александровича Харкевича и его жену Евгению Михайловну за участие и помощь в проведении НИР и создании МЭСМ.

Мне посчастливилось работать с Сергеем Алексеевичем в Киеве и в Феофании с июля 1949 года. Рос быстрее других в коллективе: инженер, инженер-конструктор, начальник группы, начальник МЭСМ, что имело как положительные, так и отрицательные стороны. В ноябре 1951 года был призван в Советскую Армию. До декабря 1951 года участвовал в подготовке МЭСМ к предъявлению на испытания.

Работать с Сергеем Алексеевичем было очень легко. Он никогда не повышал голоса, даже на явно провинившихся. Относился ко всем исключительно ровно и справедливо. Не было у него любимчиков, основных или ближайших, все были равны. Всегда отмечал даже небольшие успехи своих сотрудников. Во время отдыха и туристических походов не чужд был озорным проделкам и шуткам, которые произносил абсолютно серьезно. Привлекал откровенностью, доброжелательностью, увлеченностью, а также умением увлечь других и создать убежденность в каждом равного участия в общем деле. Создавал атмосферу радости труда, которая рождала энергию и желание трудиться. Всегда в научно-практической работе стремился сделать то, чего еще никто не сделал. Под влиянием замечательных качеств Сергея Алексеевича в коллективе воспитывались высокая сознательность, острое чувство ответственности, инициативность, профессионализм. Пример, которым был Сергей Алексеевич для молодого коллектива, – самый действенный учитель того, как надо работать и жить.

Сергей Алексеевич обладал самым дорогим даром природы: сильным умом, преследующим практически цели, веселым, насмешливым и добрым нравом.

Академик А.А. Дородницын как-то в воспоминаниях поставил вопрос: «Знал ли Сергей Алексеевич латинскую пословицу "Age quod agis" – "Делай то, что ты делаешь!"». Сергей Алексеевич говорил нам близкую по содержанию: «Что исполняешь, исполняй хорошо». А из латинских я слышал только одну. Когда Сергей Алексеевич увозил в большом портфеле в Москву первые материалы по МЭСМ, он сказал: «Omnia mea mecum porto». Вообще в Феофании Сергей Алексеевич часто пользовался крылатыми словами, четверостишиями не только из Блока и Гумилева, которых выделял, но и из Пушкина, Шекспира, Кнута Гамсуна, Бунина, Данте, Омара Хайама.

Несмотря на изнурительный режим дня, а иногда и ночи, мы были бодры и веселы, переживали полноту бытия, связанную с самоосуществлением, т.е. были счастливы. Учились работать, овладевали профессией, росли. Дружную работой всех создавалось наше счастье для всех. Самым счастливым был Сергей Алексеевич, потому что он дарил счастье всему коллективу, а каждый в коллективе, конечно, стремился быть счастливым.

Свободное от работы время уделяли спорту. Играли в волейбол, с наступлением зимы переходили на лыжи, которые пользовались меньшим успехом. Но самым яростным увлечением был пинг-понг. В помещении рядом с машинным залом проходили турниры. Первым чемпионом был заместитель заведующего лабораторией Дашевский Л.Н., но скоро он сдал свои позиции, проиграв ракетку. Хорошая была ракетка – с одной стороны пробка для защиты, с другой – мягкая резинка для нападения. Обидный был проигрыш. Впоследствии я успешно играл этой ракеткой на городских и республиканских соревнованиях по настольному теннису.

После завершения этапа в настройке или после сильного ветра, который обесточивал Феофанию, организовывались пешие или автобусные туристические маршруты по живописным местам Киевской области. Сергей Алексеевич был не только талантливейшим ученым, но и необыкновенно обаятельным и галантным мужчиной. Однажды, уже в довольно прохладное время года, когда автобус проезжал мимо озера, Алиса Григорьевна, жена Сергея Алексеевича, воскликнула: «Какие чудесные водяные лилии!» Сергей Алексеевич остановил автобус, молча разделся, влез в холодную воду и поплыл. Через несколько минут большой букет лилий он преподнес Алисе Григорьевне. Мужчины в автобусе были крайне смущены. На какое-то время прекратились обычные на маршрутах поступления в общественную кассу: ни одной остроты, ни штрафов, ни вознаграждений. Наконец Алиса Григорьевна продекламировала:

*За радость легкую дышать и жить
Кого, скажите, мне благодарить?*

Касса опять начинает расти. Самые большие поступления в нее текли тогда, когда с нами в автобусе была Алиса Григорьевна. Она была блистательно умным и необыкновенно находчивым человеком. Все ее очень любили, не меньше, чем любили и уважали Сергея Алексеевича. Алиса Григорьевна была по-настоящему членом нашего коллектива. Во всем помогая Сергею Алексеевичу, она заботилась и о том, чтобы его сотрудники были всесторонне развитыми, гармоничными людьми. Цитировала иногда К.А. Тимирязева: «Надо знать обо всем понемножку, но все о немногом». Устраивала у себя на квартире вечера-встречи с известными деятелями искусства и культуры, на которых бывали Святослав Рихтер, Гмыря, Тимошенко и Березин, многие другие. Проходили вечера исключительно интересно и были очень полезны технической творческой молодежи. Никогда на них не поднимались темы наших профессиональных занятий.

Случалось, в часы вечерних прогулок до ближайшего озера через феофанийский лес, Сергей Алексеевич высказывал вслух мысли о предстоящем развитии вычислительной техники и, особо, программирова-

ния, о непременном их самом широком распространении или, остановившись и глядя спутнику в глаза, как бы шутя, бросал совершенно новые идеи. Однажды он сказал, что универсальная машина хороша только для больших сложных задач вычислительной математики, а для решения какого-нибудь одного относительно узкого класса задач, но достаточно распространенного, целесообразно строить специализированные машины, которые должны быть значительно проще универсальных. Неожданной была мысль о том, что пора уже, по результатам работы МЭСМ, совмещать ввод информации с вычислительным процессом и выполнение операций над матрицами и векторами в вычислителе. Мы еще были так далеки от этого. Ну а мысль о том, что «единственным эффективным способом борьбы с дальними ракетами является посылка встречной ракеты», точку встречи с которой может рассчитать быстродействующая цифровая электронная вычислительная машина и вместе с радиолокационными системами обеспечить ее уничтожение, казалась такой невероятной и неосуществимой: снарядом попасть в летящий снаряд – фантастика, эту идею мы даже не обсуждали. Но спустя одиннадцать лет и она была реализована, как и многие другие новые идеи Сергея Алексеевича Лебедева.

В конце февраля 1961 года я прибыл на Балхашский полигон, будучи офицером Центрального аппарата Министерства обороны. 4 марта 1961 года меня пригласил на командный пункт (КП) системы противоракетной обороны (ПРО) генеральный конструктор Г.В. Кисунько, с которым я был знаком с 1951 года, участвуя в создании системы ПВО для обороны Москвы. На КП я наблюдал, как средства радиолокационной системы обнаружили баллистическую ракету на дальности 1500 км, а ламповые универсальные и специализированные цифровые машины, объединенные в вычислительную сеть, обеспечили:

- построение траектории баллистической ракеты (цели) и непрерывное ее уточнение;
- выдачу указания радиолокаторам точного наведения;
- расчет и выдачу на пусковые установки углов предстартовых разворотов;
- расчет момента пуска противоракеты и выдачу команды на ее пуск с пусковой установки;
- наведение противоракеты на цель в соответствии с боевым алгоритмом;
- и наконец, ПОДРЫВ, на высоте 25 км в результате прямого попадания противоракетой баллистическая головка развалилась на части.

Сбой в центральной вычислительной машине М-40 произошел за 145 секунд до подлета баллистической головки к точке ее перехвата противоракетой. Впоследствии первой системой ПРО было осуществлено еще одиннадцать успешных перехватов баллистических ракет и уничтожение их боеголовок при полном отсутствии сбоев вычислительных средств во время боевых циклов.

Президент Сибирского отделения АН СССР академик М.А. Лаврентьев и президент АН УССР академик Б.Е. Патон неоднократно говорили, что создание Сергеем Алексеевичем Лебедевым в тяжелейшие послевоенные годы первой оригинальной отечественной цифровой электронной вычислительной машины было его научным и трудовым подвигом.

Имя Сергея Алексеевича Лебедева по праву стоит в одном ряду с именами выдающихся ученых мира. На Украине высоко чтут память Сергея Алексеевича и гордятся тем, что первая в континентальной Европе и в Советском Союзе электронная цифровая вычислительная машина была создана на их земле.

В память о Сергее Алексеевиче учреждена премия Академии наук УССР имени С.А. Лебедева; один из проспектов нового района Киева назван именем С.А. Лебедева; у въезда в новое здание Института кибернетики АН УССР воздвигнута стела в память о первой электронной цифровой вычислительной машине в СССР и ее создателе С.А. Лебедеве.

СПЕЦИАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, ИСПОЛЬЗУЕМЫЕ В СИСТЕМАХ НАВЕДЕНИЯ С200 И С300

И.М. Лисовский, И.И. Лисовская

Институт точной механики и вычислительной техники им. С.А. Лебедева РАН

Москва, РФ

irina_lis@botmail.com

В 1968 году С.А. Лебедев после жарких обсуждений в коллективе ИТМиВТ принял предложение разработать специализированный управляющий малогабаритный мобильный высокопроизводительный цифровой вычислительный комплекс (ЦВК) 5Э26 для применения в ракетных системах противовоздушной обороны (ПВО). Генеральный конструктор систем ПВО – Борис Васильевич Бункин.

О реализации такой возможности Сергей Алексеевич мечтал еще в 1950 году в Киеве после создания МЭСМ.

Конечно, эта работа несколько меняла профиль института как разработчика стационарных универсальных ЭВМ и вызывала необходимость отвлечения части сотрудников на ее выполнение. Но нельзя было не учитывать, что коллектив института в ходе создания ЭВМ второго поколения реализовал технические решения важные для спецмашин.

В 5Э26 и 5Э51:

- аппаратный контроль исправности устройств,
- средства для создания многомашинных комплексов с автоматическим переходом на резерв,
- специальный процессор управления внешними устройствами,
- развитая система инженерного обеспечения устойчивого функционирования (в части электропитания и охлаждения).

Изучался опыт ввода и эксплуатации этих ЭВМ на многих объектах систем стратегического назначения и в вычислительных центрах.

В 5Э65 и 5Э67:

- мобильность вычислительных систем для работы в тяжелых условиях эксплуатации.

Трехмашинный вычислительный комплекс 5Э67 с общим полем внешней памяти до настоящего времени эксплуатируется на Крайнем Севере. (Заместитель главного конструктора ВК 5Э67 Игорь Константинович Хайлов за создание 5Э67 удостоен Государственной премии СССР.)

Кроме того, обеспечивалась возможность получения значительных финансовых средств, что позволяло расширять фронт работ по созданию ЭВМ третьего поколения с привлечением производственно-технологических мощностей заводов.

После решения ряда организационных и научно-технических проблем директор института Сергей Алексеевич Лебедев провел крупнейшую за всю историю ИТМиВТ реорганизацию: ресурсы множества разрозненных лабораторий были объединены под руководством шести головных лабораторий (фактически были созданы отделения) – по ЭВМ общего назначения, по ЭВМ специального назначения (включая архитектуру) по электронному конструированию, запоминающим устройствам, САПР и технологии.

ЦВК 5Э26 создавали молодые (в среднем 30-летние), но уже опытные специалисты. Трудились много и напряженно, радуясь важности доверенного им дела и новизне решаемых проблем. Успеху общего дела служили также дружеские и доверительные отношения между специалистами разных профилей. Возглавлял разработчиков главный конструктор ЦВК 5Э26 С.А. Лебедев, заместитель главного конструктора В.С. Бурцев. Основную группу составляли: Кривошеев Е.А., Борисов П.В., Трегубов К.Я., Острцов Ю.Д., Лаут В.Н., Козлов Л.А., Подшивалов Д.Б., Рябов Г.Г., Гушин О.К., Новиков А.А., Рябцев Ю.С., Степанов В.И.

Была определена многопроцессорная архитектура ЦВК 5Э26, обеспечивающая работу до трех модулей центральных процессоров (ЦП) и двух специальных процессоров ввода-вывода информации (ПВВ) с общей памятью, которая состоит из модулей (блоков) командной независимой памяти и модулей оперативной памяти данных. Каждый ЦП имеет память данных типа КЭШ. В ЦВК может поступать большое число внешних прерываний. Модули ЦВК охвачены полным аппаратным контролем исправности. По сигналам аппаратного контроля выполняется автоматическая реконфигурация ЦВК, в результате которой в боевой работе остаются только исправные модули. Все действия по переходу с одной рабочей конфигурации ЦВК на другую занимают не более 17 мс.

Конструктивно ЦВК представляет собой шкаф (высота 1885, ширина 2870, глубина 655 – все размеры в мм; потребляемая мощность 5,5 кВт), который ставится у стенки транспортного средства (фургона или подобного), в шкафу имеются стойки-секции, каждая из которых содержит 7 одноэтажных мест для съемных блоков. Эксплуатация ЦВК осуществляется при подходе к шкафу с одной стороны, ремонт производится заменой блоков. Используются блоки трех размеров по высоте (1-, 2- и 3-этажные). Резервирование в ЦВК осуществляется группами (модули резервирования), имеющими 1–3 блока.

Один этаж блока вмещает до 28 ячеек – типовых элементов замены, имеет каркас с объединительной панелью из двух многослойных печатных плат МПП-2. Ячейка включает шестислойную печатную плату МПП-1 размером 170 x 110 мм, на двух поверхностях которой монтируются корпуса интегральных схем (до 70 шт.), на одной стороне установлен разъемный соединитель с объединительной панелью блока, а на противоположной стороне МПП-1 имеется колодка-ручка с контактами для контрольных точек и сигналов индикации.

В качестве интегральных схем использовались в основном полупроводниковые микросхемы одних из первых отечественных серий – 133 и 130 (ТТЛ-типа).

Важными особенностями МПП ЦВК 5Э26 явились:

- наличие в платах, кроме сквозных, также «глухих» (внутренних) металлизированных отверстий для межслойных переходов;
- наличие специальных слоев «земли» и «питания» с большой площадью проводников шин разводки;
- большая (на конец 60-х годов) площадь МПП (8-слойные МПП-2 имели размеры 280 x 220 мм).

Указанные особенности МПП ЦВК 5Э26, которые в течение многих лет были недоступны для конструкторов других изделий, обеспечили высокое быстродействие и компактность аппаратуры, обусловленные большой плотностью связей при требуемой помехозащищенности, а также высокую технологичность – пригодность к автоматизированному производству, поскольку подавляющее большинство связей в серийных образцах должны были быть печатными.

Для проектирования и автоматизированного изготовления ячеек, панелей и блоков машины 5Э26 (которая была первой ЭВМ третьего поколения в ИТМиВТ и на заводах) потребовалось разработать и освоить в производстве множество новых процессов, материалов, оборудования и оснастки, организовать ряд специальных участков и цехов, переобучить инженеров и техников, технологов и производственников. Впервые были созданы:

- система автоматизированного проектирования на ЭВМ ячеек, блоков и устройств на основе ИС и МПП с выпуском полного комплекта машинной документации, включая управляющие программы для производственно-технологических установок;
- технологическая база, производственные участки автоматизированного изготовления МПП со сквозными и внутренними металлизированными переходами, включая новые материалы (тонкий фольгированный «травящийся» диэлектрик, сухой пленочный фоторезист, специальные фото пленки и многое другое);

- специальное оборудование (координатограф, сверлильные станки, линии химико-гальванических покрытий, прессы и т.п.);
- контрольно-измерительная аппаратура для проверки качества печатного монтажа и металлизации, устойчивости МПП к различного рода воздействиям и др.;
- технологические процессы, инструкции, методики;
- участки со специальным инженерным обеспечением (гермозоны, очистные сооружения);
- методика, оборудование и оснастка для монтажа ячеек и блоков, их доработки и ремонта;
- методика, специальные стенды и оснастка для наладки, контроля и разнообразных испытаний ячеек, блоков и шкафов.

Оперативное запоминающее устройство (ОЗУ) ЦВК 5Э26 было выполнено на новых ферритовых сердечниках, разработанных в ИТМ иВТ, а для обеспечения устойчивости накопителей ОЗУ к механическим и климатическим воздействиям в полевых условиях эксплуатации матрицы из сердечников заливались специальным эластичным компаундом.

Принципиально новым устройством памяти, разработанным специально для ЦВК 5Э26, явилось запоминающее устройство команд (ЗУК), выполненное на ферритовых сердечниках с двумя отверстиями – бикасах. ЗУК удовлетворял очень жестким требованиям к памяти команд ЦВК:

- обеспечивать хранение информации без какого-либо потребления энергии;
- быть устойчивым ко всем видам возможных внешних воздействий – механическим, климатическим, радиационным, электромагнитным;
- иметь достаточно большую емкость;
- обеспечивать возможность простой электрической перезаписи информации для частого внесения изменений в программы.

В начале 70-х годов (да и многие годы потом) вариант ЗУК на сердечниках с двумя отверстиями был лучшим техническим решением, несмотря на большие размеры и массу модуля ЗУК, которые вызвали необходимость сделать специальное устройство для установки модуля в шкаф и извлечения его из шкафа. При разработке и организации производства ЗУК потребовалось решить сложные технические проблемы.

Обеспечению удобной наладки первых образцов, простоты ремонта и обслуживания ЦВК в полевых условиях уделялось много внимания. Для наладки и выявления места неисправности (компонента, связи, контакта и т.д.) были разработаны специальные технологические стенды – производственный на заводе и эксплуатационный в ремонтном органе в войсковых частях. В этих стендах блоки работающего ЦВК размещались не в штатном шкафу, а в специальных разнесенных по площади стойках, позволяющих иметь доступ щупами контрольно-измерительной аппаратуры к элементам на объединительной панели блока и ко всем контактам ячеек. Сигналы с ячеек передавались на технологический пульт для индикации. В ремонтном органе восстановленный блок проходил полный контроль в составе технологического ЦВК, после чего передавался в качестве ЗИП в места эксплуатации ЦВК. Ремонт штатного ЦВК производился оператором заменой блоков (с использованием ЗИП) на основании индицируемых на шкафу сигналов «Авария» по результатам аппаратного контроля и тестирования.

Особое значение придавалось обеспечению надежности выполнения боевой задачи. Кроме структурных методов резервирования с автоматической реконфигурацией при аппаратном контроле исправности модулей для получения требуемых показателей надежности ЦВК осуществлялись самые разнообразные мероприятия по повышению «физической» безотказности всех составляющих ЦВК (компонентов, печатных плат, ячеек, блоков, межблочного монтажа и др.) во всех диапазонах изменения параметров внешних воз-

действий. Схемы ЦВК разрабатывались с использованием руководящих технических материалов, которые были созданы с учетом всех возможных факторов, влияющих на надежность работы схем во всех условиях и режимах работы ЦВК.

Конструкция ЦВК разрабатывалась специалистами Загорского электромеханического завода (ЗЭМЗ), которые имели опыт создания аппаратуры для установки в транспортные средства и работы в полевых условиях.

В работах по ЦВК 5Э26, которые обеспечивали становление отечественной научно-технической и производственной базы высокопроизводительных специальных ЭВМ третьего поколения, участвовало несколько предприятий радиопромышленности СССР: ИТМиВТ – головной разработчик, Загорский электромеханический завод (ЗЭМЗ) – головной производитель и соразработчик, НПО «Алмаз» – головной «комплексник» и организатор работ по технической базе (генеральный конструктор Б.В. Бункин), НПО «Авангард» (по гибридным микросхемам ЗУ), ЦНИТИ (по печатным платам), Львовское ПО им. В.И. Ленина – второй завод-поставщик ЦВК и др.

Необходимые новые материалы, компоненты, технологические процессы и оборудование для изготовления ЦВК 5Э26 создавались на предприятиях электронной, электротехнической, химической, станко-инструментальной промышленности СССР.

При производстве ЦВК применялись известные (в том числе по зарубежным публикациям) методы обеспечения надежности: контроль и испытания материалов и покупных комплектующих изделий, пооперационный контроль в производстве, технологические испытания и тренировки узлов, устройств и комплекса в целом, периодические и отладочные испытания и др.

Кроме общеизвестных методов обеспечения надежности аппаратуры, которые в последующие годы стали традиционными для различных изделий, при создании ЦВК 5Э26 были использованы также особые мероприятия, в том числе испытания партий микросхем на выявление «перемежающихся» отказов и испытания блоков в режиме изменения их температуры. Наличие «перемежающихся» отказов в микросхемах, при которых дефект проявляется некоторое время в виде кратковременных самовосстанавливающихся неисправностей и лишь позже приводит к полному отказу, долгое время не признавалось отечественными поставщиками микросхем. Много времени и труда пришлось затратить главному конструктору 5Э26 С.А. Лебедеву и коллективу института, чтобы добиться поставок качественных микросхем.

При производстве ЦВК 5Э26 были созданы стенды, на которых проводились также испытания комплекса в целом на устойчивость к механическим воздействиям в процессе решения задач.

Принятые меры по повышению надежности ЦВК 5Э26 и накопленные знания имели большое практическое значение, поскольку это был первый отечественный опыт в области надежности специальных ЭВМ третьего поколения.

В настоящее время эксплуатируется более 1000 экземпляров ЦВК 5Э26 и их модификаций.

За создание ЦВК 5Э26 Государственных премий СССР удостоены: Кривошеев Е.А., Острцов Ю.Д., Рябцев Ю.С. Многие из разработчиков получили государственные награды.

Коллектив разработчиков – воспитанников Сергея Алексеевича Лебедева – продолжал разработку вычислительной техники для развития систем ПВО (генеральный конструктор Б.В. Бункин) в лучших традициях школы С.А. Лебедева.

В 1977 году по результатам заводских и государственных испытаний, производства и эксплуатации ЦВК 5Э26 было принято решение разработать новые модели ЦВК 5Э26.

В новых моделях (5Э265 и 5Э266) удалось:

- повысить эффективность аппаратного контроля (АК) как за счет некоторого увеличения аппаратуры АК, так и путем увеличения программного доступа к АК;
- создать такую конструкцию ЦВК, чтобы можно было добраться щупом до большинства контактов в логических блоках во включенном состоянии;
- повысить эффективность проведения реконфигурации с помощью доработки программ, обеспечивающих изменение конфигурации;
- улучшить расположенную на шкафу индикацию состояния ЦВК для обеспечения ремонта и отладки боевых программ.

Архитектура новых моделей осталась такой же, как и у 5Э26, но потребовалось практически заново разработать все ячейки, блоки и шкаф ЦВК.

Ячейки сохранили свои размеры, а конструкция печатных плат была улучшена.

ЦВК выпускались в двух модификациях – малая (5Э262, 5Э266) и большая (5Э261, 5Э265) машины. Отличались они тем, что большая машина имела не одну, а две секции памяти, при этом емкость командной и оперативной памяти удваивалась.

ЦВК 5Э265(6) проявили лучшие характеристики надежности, чем 5Э261(2). В связи с этим были откорректированы требования ТУ на ЦВК в части среднего времени наработки на отказ (увеличено почти в два раза).

Работы по 5Э26 открыли в ИТМиВТ, а также на нескольких заводах новое направление СВТ – семейство управляющих цифровых вычислительных комплексов 5Э26–40У6, которое до сих пор остается важным.

Для развития систем ПВО Генерального конструктора Б.В. Бункина в 1980 году была начата разработка новой ЭВМ 40У6, главный конструктор Е.А. Кривошеев, которая представляет собой программно-совместимый (на уровне подмножества системы команд) с ЦВК 5Э26 вычислительный комплекс с более высокими, чем у 5Э265, характеристиками по производительности, емкости памяти (оперативной и командной), пропускной способности каналов ввода-вывода с широким набором интерфейсов, с улучшенными механизмами прерываний и системы «горячего» резервирования и реконфигурации, с обеспечением ввода данных с клавиатуры и вывода экспресс-информации о ходе боевой работы на печать.

В состав ЦВК 40У6 входит следующая аппаратура:

- 5 центральных процессоров (ЦП) – при резервировании боевая программа может задать требование на любое соотношение рабочих и резервных ЦП, либо работать в режиме деградации;
- 2 процессора ввода-вывода – резервирование 1:1, т.е. одно устройство находится в рабочем режиме и одно резервное;
- 2 блока оперативной памяти – резервирование 1:1, при этом обеспечивается дублирование информации, объем памяти 256 Кбайт;
- 2 блока командной памяти – резервирование 1:1, возможно включение обоих блоков без резерва с удвоением объема памяти, объем памяти 512 Кбайт;
- блок адаптера каналов;
- блок пульта управления – не резервируется и блокируется при боевой работе;
- система питания – позволяет ЦВК работать от сети 50 или 400 Гц;
- внешняя память используется для холодной загрузки памяти команд, ввода информации, смены версий боевых программ;
- телетайп – ручной ввод информации и печать данных.

Каждое логическое устройство размещается в одном блоке, а вся перечисленная аппаратура – в шкафу объемом 2,5 куб. метра.

Требуемая производительность ЦВК 40У6 (2,5 млн. оп./с) достигнута за счет следующих архитектурно-технических решений:

- многопроцессорность;
- аппаратная реализация вычислений элементарных функций;
- быстрая локальная оперативная память в каждом процессоре;
- одновременное обращение процессоров к командной и оперативной памяти, расслоение памяти;
- организация ввода-вывода без обращения к ЦП.

Обмен информацией с системой организуется с помощью специального ПВВ. Такое решение позволяет освободить ЦП от ведения обмена, за исключением пуска каналов при запуске боевых программ. В дальнейшем ПВВ самостоятельно управляет обменом. ПВВ имеет 16 каналов, суммарный темп обмена 2 Мбайта в секунду.

Для обеспечения высоких параметров надежности в ЦВК 40У6 реализована система резервирования, базирующаяся на следующих принципах:

- применена система аппаратного контроля для организации автоматического резервирования и подтверждения достоверности результатов вычислительного процесса;
- посредством практически полного аппаратного контроля (АК) устройств достигается высокая эффективность контроля при выполнении боевых программ в боевом режиме (испытаниями установлено, что АК охватывает 97% аппаратуры 40У6);
- программный контроль в режиме теста проверяет всю программно-доступную аппаратуру, задавая режимы проверки как для цепей, охваченных АК, так и для цепей, доступных только программе;
- каждое функциональное устройство имеет резерв с собственным блоком питания;
- в резервную оперативную память введено дублирование информации;
- штатная конфигурация ЦВК 40У6 для выполнения рабочей программы состоит из одного устройства каждого типа и четырех ЦП;
- реконфигурация во время выполнения боевых программ проводится в течение не более 17 мс;
- начальная конфигурация формируется с учетом результатов прохождения функционального контроля по всему объему аппаратуры;
- в случае недостаточности исправных функциональных блоков для создания штатной рабочей конфигурации или превышения времени реконфигурации формируется сигнал «отказ ЦВК».

Параметры надежности зависят также от организации ремонта. Поэтому в системе аппаратно-программного контроля ЦВК введена индикация на каждой ячейке «авост собственный» и «авост смежный». «Авост собственный» указывает на наличие неисправности в данной ячейке. «Авост смежный» возникает, когда неисправность находится на «ничейной земле», т.е. в аппаратуре, расположенной между последним уровнем контроля в ячейке, откуда приходит некорректная информация, и входным контролем в данной ячейке. Любой «авост» по желанию оператора может блокировать движение информации в ЦВК. Этим достигается наиболее точное указание области аппаратуры, в которой возникла неисправность.

Для ЦВК 40У6 разработана «Методика направленного поиска неисправностей», в которой входной информацией являются сигналы контроля, а выходная информация указывает оператору на необходимое следующее элементарное действие. За несколько шагов неисправность, как правило, локализуется с точностью до нескольких ячеек. Такая организация ремонта не требует персонала высокой квалификации.

В качестве элементной базы для ЦВК 40У6 были выбраны ИС серий 530 и 533, 537 РУ6 (для командной и оперативной памяти), отдельные ИС серий 539 и 169, ИС 537 РУ9А для локальной оперативной памяти. Наличие режима хранения информации в ИС 537 РУ6 при пониженном напряжении дало возможность сохранять командную информацию с выключенным питанием ЦВК. Это обеспечивается введением специального внутреннего источника питания – аккумуляторной батареи. Емкости одной батареи хватает на 3–5 лет, а при соответствующей подзарядке – на весь срок эксплуатации ЦВК.

ЦВК 40У6 разрабатывался в то время, когда предприятия радиопромышленности (НПО «Алмаз», НПО «Авангард» и др.) с участием многих отраслей промышленности СССР в соответствии с постановлением правительства начали решать сложную проблему общесоюзного значения – создание базы автоматизированного серийного изготовления радиоэлектронной аппаратуры (РЭА). При этом предусматривалось разработать унифицированные конструктивы, пригодные для автоматизированного производства узлов и устройств РЭА, создать необходимые материалы, технологические прессы, оборудование и оснастку, типовые участки, цехи и заводы, организовать специализированные производства и их кооперацию. Комплексы 40У6 должны были стать первой крупной продукцией автоматизированного производства, поэтому для них были заданы конструктивы, разрабатывавшиеся в НПО «Алмаз» в качестве унифицированных. Однако в процессе разработки ЦВК 40У6 и организации их производства в ПО «Звезда» потребовалось основательно доработать первоначальный проект металлоконструкций и наполнить его базовыми технологическими решениями по реализации разнообразных связей и электропитания, что было с успехом сделано (по настоянию генерального конструктора системы ПВО Б.В. Бункина) при тесной совместной работе специалистов ИТМиВТ, ПО «Звезда» и НПО «Алмаз».

Программное обеспечение, поставляемое с ЦВК под обозначением КОП-40У6, содержит контрольные и обслуживающие программы, используемые при производстве и эксплуатации.

Контрольные программы включают тесты всей аппаратуры, доступной программному контролю: функциональных блоков, межблочных связей, аппаратуры контроля и управления конфигурацией, внешних связей, периферийных устройств органов управления и индикации.

Обслуживающие программы производят начальную загрузку памяти команд, всю работу с конфигурационным состоянием ЦВК, определяют состояние «ЦВК исправен» и «ЦВК неисправен», обеспечивают оперативный обмен информацией с внешними устройствами и использование дублированной информации в памяти.

Работа обслуживающих программ в части конфигурации ЦВК проводится в условиях жесткого реального времени, обеспечивая во время боевой работы завершение перехода на новую конфигурацию ЦВК и продолжение выполнения боевой программы не более чем через 17 мс после возникновения неисправности.

Встроенная разветвленная система аппаратного контроля пронизывает все схемы ЦВК 5Э26-40У6 и обеспечивает их безотказность при работе в режиме жесткого реального времени с требуемым очень коротким временем реакции на сбои и отказы, а также временем готовности 15 сек.

Все составляющие ЦВК семейства 5Э26-40У6 – архитектура, элементная и конструктивно-технологическая база, программное обеспечение являются полностью отечественными. Реализованные на этой основе комплексы около 1300 экземпляров обеспечили все требования, предъявляемые к вычислительным средствам сложных мобильных систем вооружения. Для этих систем имеется хорошо отработанное программное обеспечение, а также средства его сопровождения, модернизации и развития, имеются ремонтные базы, тренажеры, накопленная методика эксплуатации.

Высокие качества ЦВК 5Э26-40У6 обусловили их долгую жизнь в системах, которые еще много лет будут находиться на вооружении ВС РФ.

Часть систем с ЦВК 5Э26-40У6 была продана зарубежным странам и до сих пор поступает на экспорт.

Задачи поддержания ЦВК семейства 5Э26-40У6 в эксплуатации, их модернизации и развития остаются актуальными и в настоящее время.

За создание ЦВК 40У6 главный конструктор Е.А. Кривошеев удостоен Государственной премии, а ряд разработчиков получили государственные награды.

После известных событий еще остался небольшой костяк коллектива, который способен продолжить развитие семейства 5Э26-40У6. Хочется верить, что скоро опять понадобится создавать в России ЭВМ с отечественной архитектурой и технической базой для систем, которые по своим тактико-техническим показателям не уступают зарубежным, как это было в относительно недавнем прошлом.

После окончания разработки моделей 5Э261, 5Э262, 5Э265, 5Э266 в 1978 году Государственный комитет СССР по делам изобретений и открытий выдал авторское свидетельство № 157346 на изобретение ЦВК 5Э26 авторам: Бурцеву В.С., Кривошееву Е.А., Борисову П.В., Острцову Ю.Д., Степанову В.И., Румянцеву Е.М., Трегубову К.Я., Фадееву М.Ф., Обидину Д.И., Козлову Л.А., Крылову А.С., Рябцеву Ю.С. (заявка № 2271485, приоритет изобретения от 10 января 1980 года).

Госкомитет СМ СССР по делам изобретений и открытий выдал авторское свидетельство № 94472 Кривошееву Е.А., Фадееву М.Ф., Степанову В.И., Обидину Д.И., Вайсеру О.Г. на изобретение процессора обмена для ЦВК 5Э26 в соответствии с описанием и приведенной в нем формулой (заявка № 1584552 с приоритетом от 26 декабря 1974 года).

ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННЫЕ ПРОЦЕССОРЫ ДЛЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Н.А. Лукин

Институт машиноведения УрО РАН
Екатеринбург, РФ
nicklookin@mail.ru

Совершенствование систем реального времени приводит к децентрализации вычислений и, как следствие, к появлению в их составе функционально-ориентированных процессоров (ФОП). Создание рациональных или оптимальных архитектур ФОП возможно лишь на основе совместной оптимизации алгоритмов и архитектур. Особенно это актуально для варианта реализации ФОП как «Системы на кристалле». На примере задачи вычисления математических функций с гарантированной точностью раскрываются все этапы архитектурного синтеза – от выбора численного метода до формирования архитектуры СБИС ФОП.

ВРЕМЕННЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ КОЛЛЕКТИВ (ВНТК) «СТАРТ»

А.Г. Марчук

Институт систем информатики им. А.П. Ершова СО РАН
Новосибирск, РФ
mag@iis.nsk.su

Временный научно-технический коллектив (ВНТК) «СТАРТ» был создан в 1985 году совместным решением Академии наук и ГКНТ на срок 3 года. В Постановлении [1] было записано: «В целях отработки и экспериментальной проверки элементов концепции ЭВМ 5-го поколения...»

Предыстория этого решения в том, что в начале 80-х годов в Японии был запущен национальный проект создания компьютеров пятого поколения. Страна восходящего солнца, добившись существенных позиций на мировом рынке автомобилестроения и бытовой электроники, не желала смириться с почти нулевыми позициями в вычислительной технике. На решение прорывной задачи были брошены лучшие силы и огромные средства.

Государственный комитет по науке и технике СССР захотел выяснить, насколько серьезная угроза заключается в этом японском проекте. Была создана рабочая группа под руководством В. Котова, в то время заместителя директора ВЦ СО АН. В эту группу вошли ведущие специалисты Академии наук по вычислительной технике, программному обеспечению и системам искусственного интеллекта, а также представители промышленности. Перед группой была поставлена задача – не только оценить научную обоснованность и технологическую реализуемость японского проекта, но и предложить концепцию развития компьютеров и программного обеспечения в нашей стране на перспективу.

Это был уникальный шанс. Дело в том, что после создания первых компьютеров, в чем специалисты академических институтов принимали самое непосредственное участие (с творческим пиком в середине 60-х в виде БЭСМ-6), Академия наук была «отлучена» от компьютеростроения, и отдельные коллективы занимались в основном новациями в области программирования. Как итог – страна получила серии ЕС и СМ, несколько идейно не продвигавших, но фиксирующих наше технологическое отставание. А новые разработки типа «Эльбрус» и ПС-2000 выглядели уже недостаточно революционными. Рабочая группа, в которую входили такие признанные специалисты, как Э.Х. Тыугу, Ю.Г. Евтушенко, В.Е. Котов, А.С. Нариньяни, В.М. Брябрин, а также молодые, но уже набравшиеся опыта исследователи и разработчики, собранные из разных городов и научных коллективов СССР, с энтузиазмом принялась за дело. По режимным соображениям работали в почти полной изоляции в течение месяца, результатом деятельности явился развернутый отчет-концепция, получивший, к сожалению, гриф «секретно».

Хорошо помню, что серьезных разногласий, а тем более конфликтов не было. Была творческая атмосфера, научный анализ, уважение к чужому мнению. В отчете совсем мало было написано про японский проект, недостатки которого не хотелось даже критиковать. Будущее показало, что мы были правы и из того японского «чуда» мало что получилось. Кстати, и ведущие западные страны ответили на «японский вызов» программами национального уровня, и результаты американцев и европейцев оказались впоследствии куда более значимыми. Энтузиазм и творческую атмосферу нашей рабочей группы подпитывало ощущение начала большой созидательной работы.

Сформулированные в концепции принципы и контуры технических решений показывали направление (мы сконцентрировались только на одном) технологического развития отрасли, которая в будущем получит название ИТ – информационные технологии. Не вдаваясь в технические подробности, скажем, что была предложена схема построения компьютеров и программного обеспечения, ориентированных на пользователя. На вершине умозрительной конструкции гармонично располагались тогда еще «сверхновые» средства интеллектуализации.

Замечу, что ядром рабочей группы являлись представители Новосибирского вычислительного центра: В.Е. Котов, А.С. Нариньяни, Е.П. Кузнецов, А.Г. Марчук. Что послужило причиной такого доверия? Причиной был проект МАРС (Модульные асинхронные развиваемые системы), его выполнение началось в ВЦ СО АН СССР еще в 1982 году, и за три года уже были получены заметные результаты. Концепция МАРС была признана в научных кругах, полным ходом шло проектирование суперкомпьютера МАРС-М, были установлены деловые связи с отраслевыми организациями в Москве, Северодонецке, Киеве. Подготовленный доклад был представлен, изучен и получил поддержку в ГКНТ, Академии наук и профильных министерствах. Результатом явилось то, что нам было предложено сформировать временный (сроком на три года) коллектив разработчиков для реализации предложенной концепции – ВНТК СТАРТ.

Работа

Основу СТАРТа составили команды, возглавлявшиеся основными участниками рабочей группы. Так получилось не из конъюнктурных соображений, а потому, что за время подготовки концепции было осознано, что наши представления о цели едины, мы хорошо дополняем друг друга и способны делать общее дело. Коллектив был существенно децентрализованным. Самая большая часть ВНТК была в Новосибирске, в нее вошли специалисты из ВЦ СО АН, разработчики и конструкторы из Новосибирского филиала Института точной механики и вычислительной техники (ИТМиВТ), СКБ вычислительной техники (ныне КТИ ВТ), Отдельной прикладной лаборатории Министерства приборостроения. В Москве коллектив состоял из ученых и разработчиков ВЦ АН и возглавлялся Ю.Г. Евтушенко (ныне академик, директор ВЦ РАН) и В.М. Брябриным. Таллинский коллектив объединял специалистов из Института кибернетики АН ЭССР и СКБ. Позже ВНТК «СТАРТ» расширился за счет разработчиков киевского ПО «Микропроцессор». По необходимости привлекались специалисты из других городов и организаций. В разные моменты времени в «СТАРТе» участвовали от 200 до 300 человек.

Все было очень конкретно: надо было разработать и создать набор аппаратных средств и программного обеспечения, определенный техническим заданием. Причем речь шла о пионерских разработках, выполнявшихся на самом современном мировом уровне. Такую задачу и в такие сроки можно было выполнить только молодым коллективом, заряженным энтузиазмом. И то, и другое присутствовало в полном объеме. «Зеленая» молодежь была не только в исполнителях, но и, что важно, среди лидеров команд, и их позиция всегда уважалась и учитывалась. Я, в свои 33, будучи заместителем руководителя ВНТК, уже чувствовал себя умудренным опытом «стариком».

Наиболее полно и комплексно молодежная тема раскрылась в кроносовской команде. Кронос – это название 32-разрядного компьютера, разработанного студентами НГУ, одного из первых в мире в своем классе микроЭВМ. Кронос – это легенда, начавшаяся до СТАРТа, окрепшая и вышедшая на союзную орбиту во время СТАРТа и продолжившаяся после СТАРТа. Кронос и сейчас легенда, теперь уже с налетом ностальгии по молодым годам, для всех тех, кто соприкоснулся с этим неординарным явлением.

Наверное, все началось в КЮТе Академгородка, где Женя Тарасов и Володя Филиппов в разных лабораториях разрабатывали процессор и видеомонитор. А в Казахстане Дима Кузнецов учился программированию по книгам, поскольку компьютеров в их городе не было. А может, история началась в общежитии физфака НГУ, где Женя и Володя уже объединили свои творческие усилия, к ним присоединился Володя Васекин и пришли «настоящие» программисты с матфака Дима (2-й курс) и Алексей Недоря (3-й курс), вместе они организовали клуб Интродер.

Для меня и для рассказываемой истории все началось с того, что в 1983 году упомянутые персонажи, уже студенты 3-го курса НГУ (Алексей – 4-го) пришли ко мне и заявили: «Мы хотим создать 32-разрядный компьютер! И мы знаем, как это сделать». Пришлось поддержать... Больно ребята хорошие. Юношеская смелость в сочетании с полученными в НГУ фундаментальными знаниями творит чудеса, если разглядеть в молодых людях талант, энтузиазм и грамотность, а после этого – доверять и помогать. Чудеса произошли, и в качестве дипломной работы Женя защищал вторую (!) версию процессора Кронос, а Дима – систему кросс-компиляции для этого компьютера. Кстати, Кронос-2 потом был многократно повторен (по оценкам – в 100 экземплярах), поскольку очень многие захотели иметь «настоящий» 32-разрядный персональный компьютер, и это стало так доступно... А паял и налаживал Кроносы Володя Филиппов, он и сейчас любит паять, хотя ныне он заместитель директора ИСИ и у него нет ни такой необходимости, ни тем более времени.

Существенной особенностью проекта Кронос была исключительная, я бы сказал магическая, притягательность его для талантливых студентов. Такого эффективно работающего, стихийно самоорганизованного молодежного коллектива я больше никогда не видел. В выделенных для работы 40 квадратных метрах день и ночь что-то программировалось (программировали обычно ночью), обсуждалось, приходили все новые и новые студенты, им ставили задачи, а уже через небольшое время они чувствовали себя старожилами Кронос-движения. Через этот проект прошло, наверное, человек 30. Сразу скажу, что все они достигли высокой квалификации и сейчас являются лидерами своих нынешних команд, хотя и в разных организациях, городах и странах.

Но вернемся к СТАРТу. Важно было правильно организовать работу и добиться ее эффективности. Нынешним поколениям программистов трудно себе представить, в каких ограниченных условиях разрабатывались программы и системы в 70-х и в начале 80-х. Чтобы существенно улучшить инструментальную поддержку разработок, было принято решение закупить вычислительную технику зарубежного производства. Не все помнят, но тогда существовало строгое американское эмбарго на продажу в СССР современной техники, включая и вычислительную. Поэтому была закуплена персональная техника и рабочие станции производства австралийской фирмы. Несмотря на свои недостатки, австралийская техника внесла огромный вклад в выполнение поставленных задач, поскольку позволила работать, хотя бы в программировании, на достаточно высоком уровне технологической поддержки. Не буду перечислять, чего нам не хватало по сравнению с зарубежными коллегами, все же возможности даже «элитного» снабжения были весьма ограничены, нехватка с избытком восполнялась энтузиазмом и высокой квалификацией разработчиков.

Работа была организована методом «бригадного подряда», т.е. команда брала выполнение конкретного задания под коллективную ответственность. Соответственно, отношения в бригадах были весьма демократичными, а «бригадир» выполнял в большей мере обеспечивающую и внешнюю координирующую роль, нежели роль начальника директивного стиля. Не так давно я с удивлением узнал, что аналогичные принципы проведения разработок используются в фирме Microsoft и некоторых других весьма консервативных компаниях. Соответственно, был совет бригадиров, планерка и многое другое.

Важным элементом деятельности СТАРТа были ежегодные январские сборы в Ивантеевке. Ивантеевка – рабочий поселок под Москвой, где в дачной зоне собиралась наиболее активная часть СТАРТа для самоотчета, планирования и координации действий. Почему в январе? Наверное, потому, что первые сборы были еще до официального начала работы ВНТК, там мы сформировали и уточнили техническое задание, легшее в основу официальных документов, готовившихся к 1 апреля. А потом уже появилась традиция.

Ивантеевка, так мы называли наши сборы, была конференцией ни на что не похожей. Привозилась техника – компьютеры, созданные образцы, проекционный аппарат (тогда это было в большую новинку),

организовывалась работа так, чтобы можно было что-то доложить, что-то обсудить, показать, тут же внести изменения, разработать прототип. Эта была настоящая двухнедельная творческая мастерская, где каждый вносил свой важный вклад в обсуждение и реализацию общих и частных вопросов. На сон практически никогда времени не хватало, благо все – молодые, полные энергии. Большой проект никогда не развивается линейно, всегда есть необходимость вносить существенные коррективы, и они вносились, будучи каждый раз тщательно обсужденными. Например, в основу разрабатываемой параллельной ЭВМ среднего класса сначала предполагалось заложить разработку Таллинского СКБ, выполнявшуюся на стандартных компонентах. После того как совместно была высоко оценена перспектива студенческой разработки Кронос, было принято решение переработать Кронос в соответствии с требованиями проекта и промышленности и использовать его как базу для создаваемых в Таллине рабочих станций и создаваемого в Новосибирске макета параллельной супермини-ЭВМ.

Кроме всего прочего, Ивантеевка способствовала формированию личной дружбы в нашем распределенном коллективе. Эта дружба сохранилась и по сей день, превратившись в своего рода «братство». Теперь, куда бы ты ни поехал, если там есть бывшие стартовцы, тебе будут бесконечно рады, а воспоминаниям при встрече не будет конца.

СТАРТ не был обойден вниманием прессы и телевидения. Правда, внимание весьма своеобразное – увидеть «чудо» и написать о нем как о чуде, а не как о хорошей, правильно организованной, профессиональной работе. Сколько раз мы просили журналистов показать нам свои тексты для хотя бы мало-мальской технической экспертизы – ни разу они не сочли это нужным. Чего только не писали... Официальные инстанции также любили использовать наш материал для решения своих задач, только то, что говорилось с высоких трибун, весьма приблизительно отражало существо дела. Например, М. Горбачев, тогда Генеральный секретарь ЦК КПСС, на очередном съезде ВЛКСМ рассказал, что некий коллектив, возглавляемый молодым доктором Котовым (вряд ли комсомольцы могли посчитать молодым 50-летнего!), создал суперкомпьютер с быстродействием 100 млн. операций в секунду (перед нами таких задач не стояло).

А вот решения неизбежно стоящих перед молодыми людьми социальных задач мы организовать не смогли... Система жилищного обеспечения была уже в таком кризисе, что каждое место в общежитии, каждая квартира «выбивались» с большим боем и в целом мало улучшали ситуацию. Нерешенность бытовых проблем позже, в 90-е годы, послужила одной из существенных причин отъезда за рубеж многих наших молодых талантов.

Результаты

Точно в назначенный срок, перед завершением трех лет, правительственным решением была создана Государственная межведомственная комиссия для приемки работ, выполненных ВНТК «СТАРТ». Нам было что показать.

Вот неполный перечень результатов, предъявленных комиссии: четырехпроцессорный действующий макет многопроцессорного комплекса МАРС-Т, построенный на базе разработанных процессоров Кронос 2.6, снабженный мультипроцессорной мультипрограммной операционной системой нашей разработки, успешно выполнивший предложенные тестовые параллельные программы; рабочая станция ПИРС, оснащенная полным набором системного программного обеспечения, включая две операционные системы Unix и Excelsior (разработка группы Кронос), снабженная достаточным для многих применений набором прикладных программ; язык параллельного программирования БАРС, язык архитектурного проектирования Поляр, язык и система программирования НУТ, дающие новые средства интеллектуализации; набор про-

грамм для персональных ЭВМ и рабочих станций «Спектр»; компиляторы с языков Модула-2, C++, Фортран; система автоматизации проектирования сверхбольших интегральных схем, графические средства, САПР машиностроения. И это не полный список.

Первые два года СТАРТа показали, что разрабатывается нечто совершенно новое, достойное быть тиражируемым и примененным в народном хозяйстве. Поэтому на завершающем этапе были подключены новые группы конструкторов, разработчиков промышленных программ, разработчиков микропроцессоров. Было принято решение работать на перспективу, вовлекая в процесс внедрения разработок отраслевые институты, КБ и заводы. В результате завершение СТАРТа не означало завершения работ, поскольку НИИ уже перешел в ОКР и создание специальных систем. Научные и творческие контакты множились и распространялись по городам и странам, в первую очередь – странам СЭВ. Планировалось, что Кронос будет устанавливаться в качестве бортового компьютера на самолеты и космические аппараты, разработка микропроцессорного набора вышла на завершающую стадию, к производству рабочих станций и параллельных компьютеров проявили конкретный интерес самые различные фирмы, программное обеспечение внедрялось и тиражировалось на самых различных платформах, включая и тогда еще новые ПЭВМ, системы САПР адаптировались к жестким условиям промышленной конструкторской деятельности.

К сожалению, большая часть замыслов не получила развития из-за произошедших в конце 80-х сначала экономического, а потом и политического коллапсов. Тем не менее заметная часть разработок нашла свое место и в новых условиях. Анализируя результаты, я оцениваю главный итог работы не в выпущенной продукции, а в научных результатах и творческом и профессиональном росте тех людей, для которых СТАРТ был не просто работой, а порой неуклонного движения вперед, порой становления как специалиста и лидера. Жизнь «разбросала» стартовцев по самым разным точкам земного шара, но верность профессиональному долгу и призванию сохранили большинство. Мы поддерживаем теплые отношения и сейчас. Наука – дело интернациональное, и те, кому досталось работать за рубежом, и те, кто работает в области информационных технологий в России, делают общее дело, вспоминая молодость как пример настоящей работы.

Мнения участников о СТАРТе

Энн Харалдович Тыгу, Институт кибернетики, Таллин, Эстония

«Среда программирования НУТ <http://cs.ioc.ee/~nut/> была постепенно доведена до состояния готового продукта. Ее применение дало ряд интересных научных результатов, и она до настоящего времени используется исследователями для моделирования сложных динамических процессов» [3].

Виталий Телерман, ныне ведущий разработчик Dassault Systèmes, Франция

«С одной стороны, можно сказать, что результаты, полученные еще в СТАРТе 15 лет тому назад, наконец-то внедрены в производство, пусть и по "французскому" пути. Но, к сожалению, не все еще сделано, необходимость чего-то только-только начинает доходить до понимания разработчиков и очень многое остается пока невостребованным» [2].

Виктор Михайлович Брябрин, Broad Street Software Group, Мериленд, США

«Во времена СТАРТа все это придумывалось и тут же реализовывалось со всей изобретательностью, искусством, изяществом и энтузиазмом благодаря таланту и профессионализму наших сотрудников и аспирантов – участников проекта. Все это, конечно, можно сказать почти обо всех участниках СТАРТа – в Москве, Новосибирске и Таллине» [4].

Вадим Евгеньевич Котов, Университет Карнеги-Меллон, США

«Идеи модульных асинхронных развиваемых систем (МАРС), легшие в основу разработок СТАРТа, я (творчески переработав) продолжил в Лабораториях Хьюлетт-Паккард в своей работе над архитектурой сверхбольших систем (Systems of Systems). Эти работы привели к "сервисной" архитектуре (Service-Oriented Architecture), в которой все уровни сверху донизу представляют собой федерации независимых взаимодействующих компонентов – сервисов, размещенных на сети – федерации независимых серверов. Этот же подход я использую в нынешних проектах по созданию высоконадежных систем для НАСА» [5].

Александр Семенович Нариньяни, РосНИИ искусственного интеллекта, Москва

«Главное приключение было летом, которое мы (Котов, Брябрин, я и другие коллеги) провели в кабинете Гурия Ивановича, разрабатывая концепцию проекта и вызывая массу экспертов и чиновников для обсуждения разных аспектов этого непростого документа» [6].

Андрей Хапугин, ООО «Эксельсиор», Новосибирск

«Чаще всего вспоминается статья, по-моему, в "Комсомолке", о группе Кронос, которая начиналась со слов "Дмитрий Кузнецов в свои 25 лет – суперпрограммист международного класса, в чем и отдает себе полный отчет". Прохода Дмитрию Кузнецову после этого не было не одну неделю..." [7].

Библиографический список

1. Постановление ГКНТ и Президиума Академии наук СССР от 19 марта 1985 г. № 101/48 .
<http://start.iis.nsk.su/archive/eaindex.asp?lang=1&did=23849>
2. Телерман В. Воспоминания о СТАРТе. **<http://start.iis.nsk.su/persons/telerman/index.shtml>**
3. Тыугу Э.Х. СТАРТ в Таллине и что из этого получилось.
<http://start.iis.nsk.su/persons/tyugu/index.shtml>
4. Брябрин В.М. Проект СПЕКТР. **<http://start.iis.nsk.su/persons/bryabrin/index.shtml>**
5. Котов В.Е. Ответы на вопросы анкеты. **<http://start.iis.nsk.su/persons/questionnaire/kotov.shtml>**
6. Нариньяни А.С. Ответы на вопросы анкеты.
<http://start.iis.nsk.su/persons/questionnaire/narinyani.shtml>
7. Хапугин А. Ответы на вопросы анкеты.
<http://start.iis.nsk.su/persons/questionnaire/khapugin.shtml>
8. Марчук А.Г. 20 лет легендарному ВНТК «СТАРТ» // Наука в Сибири. 2005. № 16 (2502).

СИБИРСКАЯ ШКОЛА ИНФОРМАТИКИ АКАДЕМИКА А.П. ЕРШОВА КАК НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ФЕНОМЕН

А.Г. Марчук, И.А. Крайнева, Н.А. Черемных, Н.А. Куперштох

Институт систем информатики им. А.П. Ершова СО РАН

Институт истории СО РАН

Новосибирск, РФ

mag@iis.nsk.su, cora@iis.nsk.su, cher@iis.nsk.su, skn@demakova.net

В докладе рассматривается феномен Сибирской (или Новосибирской) научной школы информатики и программирования, созданной академиком А.П. Ершовым, одним из пионеров теоретического и системного программирования. Используется дескриптивный метод повествования, поскольку авторы ставят своей целью рассказать о своеобразии данной научной школы, а также осветить роль личности лидера в становлении и развитии научной школы, каковым, несомненно, был академик А.П. Ершов.

Среди различных социальных форм организации научных сообществ особое место занимают научные школы. Изучению этого феномена посвящена обширная литература, которая была обобщена Н.А. Куперштох [1]. Ее историографический очерк можно дополнить некоторыми публикациями на данную тему в Интернет [2]. Кроме того, отметим новую публикацию известной исследовательницы социального феномена научной школы Е.З. Мирской [3], в которой дан социологический анализ, описаны признаки научной школы, роль и функции научной школы в развитии науки, рассмотрен ее жизненный цикл.

В исследовании феномена научной школы можно проследить эволюцию от теоретического до практического подхода, который связан с решением государства о финансовой поддержке ведущих научных школ. Понятие «научная школа» не является строго детерминированным. Тем не менее в [4] оно трактуется как относительно небольшой научный коллектив, объединенный не столько организационными рамками, не только конкретной тематикой, но и общей системой взглядов, идей, интересов, традиций, сохраняющейся, передающейся и развивающейся при смене научных поколений. В этом контексте немаловажно получить четкое представление о конкретных научных школах.

Появление научных школ по информатике и программированию относится к 50-м гг. XX века. Первыми их летописцами были сами программисты, А.П. Ершов – один из них. Подготовленные им совместно с М.Р. Шура-Бурой небольшие по объему совместные публикации по истории программирования в СССР [5] не потеряли своей актуальности до сегодняшних дней. В преддверии 75-летия со дня рождения академика А.П. Ершова на сайте Института систем информатики Сибирского отделения РАН (ИСИ СО РАН), который носит его имя, осуществлена электронная публикация этих изданий. В свое время они были изданы небольшим тиражом в 150 экземпляров и сразу же стали библиографической редкостью. Публикация снабжена небольшим комментарием, в котором говорится о неиссякаемом интересе А.П. Ершова к истории становления программирования и неоднократном обращении к этой теме.

В 1990-е гг. появилась серия публикаций по истории кибернетики и информатики в СССР, в которых представлены воспоминания участников событий и архивные материалы [6]. Д.А. Поспелов в структуру информатики включает полтора десятка научных направлений, составляющих в настоящее время основные области ее исследования. Среди научных школ в информатике он выделяет пять наиболее значимых: это школы А.А. Ляпунова, М.А. Гаврилова, О.Б. Лупанова – С.В. Яблонского, В.М. Глушкова и М.Л. Цетлина – М.М. Бонгарда [7]. Представляется, что А.А. Ляпунова скорее можно причислить к создателям нескольких школ-направлений: личность этого ученого была настолько многогранной, а интересы настолько многосторонними, что говорить о нем как о руководителе одной научной школы неправомерно. А.П. Ершов был, пожалуй, самым ярким учеником А.А. Ляпунова, создателем Сибирской школы программирования.

Впервые подробный очерк научных достижений А.П. Ершова и его школы был написан его другом и ближайшим коллегой И.В. Поттосиным [8]. По его инициативе были подготовлены два сборника статей и документов, посвященных истории Новосибирской школы программирования [9]. В сборниках нашли отражение разноплановые материалы: анализ выполненных проектов, история их создания и использования, воспоминания о людях и событиях конца 1950–1980-х гг., когда Андрей Петрович возглавлял отдел программирования сначала в Институте математики, а затем в Вычислительном центре Сибирского отделения.

В 2000 г. при финансовой поддержке Microsoft Research начал осуществляться проект по созданию электронной версии архива академика А.П. Ершова, документы которого хранятся в Мемориальной библиотеке его имени в ИСИ СО РАН в Новосибирске. Архив создавался на протяжении всей жизни ученого и охватывает значительный временной период – с начала 1950-х до конца 1980-х гг. Систематизация документов, массив которых составляет более 550 единиц, проводилась по тематико-хронологическому принципу. В настоящее время этот проект близится к завершению. Исследователи по истории науки получают возможность доступа к уникальному своду источников электронного архива А.П. Ершова, размещенных на сайте: <http://ershov.iis.nsk.su/archive>.

Первые публикации на основе материалов архива были сделаны сотрудниками института [10], некоторые источники вошли в сборники статей и документов по истории кибернетики [6]. В подготовленную к 75-летию со дня рождения А.П. Ершова (2006) книгу «Андрей Петрович Ершов – ученый и человек» включены воспоминания коллег, учеников и друзей (в том числе и из других стран), его собственные яркие статьи, письма и стихи, а также наиболее интересные документы архива. В этих публикациях нашла отражение только часть материалов, и это естественно, поскольку основная работа по изучению архива и введению в научный оборот его документального массива только разворачивается [11].

Школу программирования академика А.П. Ершова можно отнести к исследовательскому типу научных школ. Тем не менее она отличалась рядом особенностей. Традиционным для научной школы считается сочетание зрелых и начинающих исследователей. Однако в период становления отдела А.П. Ершова формировался коллектив сверстников с разницей в возрасте в 2–3 года. Молодость коллектива не была препятствием на пути научного творчества. Первый крупный проект, который отдел программирования начал воплощать еще в ВЦ АН СССР, а продолжил в Институте математики СО АН СССР, – создание Альфа-транслятора – был завершен лишь на год позже двух других аналогичных проектов, разработкой которых руководили более опытные ученые, доктора физико-математических наук М.Р. Шура-Бура (ТА-2) и С.С. Лавров (ТА-1). А.П. Ершов, организаторские способности которого проявились достаточно рано, в силу ряда обстоятельств за короткий период стал идейным и формальным лидером коллектива программистов в Новосибирске.

Становлению А.П. Ершова как руководителя способствовали не только его деловые качества, высокая требовательность к себе, которая давала ему право требовать с других, но и такие черты характера, как доброжелательность, открытость, такт, а также сознательное желание постичь науку управления. Незадолго до отъезда в Новосибирск Андрей Петрович обратился к своему старшему коллеге М.Р. Шура-Буре, возглавлявшему отдел автоматизации программирования в Институте прикладной математики АН СССР (ныне ИПМ РАН им. М.В. Келдыша). Тот преподавал Ершову уроки менеджмента, которые касались кадрового состава, организации рабочего времени, планирования НИР и других составляющих работы коллектива. Кроме того, А.П. Ершов самостоятельно осваивал навыки руководителя научного подразделения, изучая специальную литературу на эту тему в «Вестнике АН СССР».

Другая характерная черта школы А.П. Ершова – это приверженность научным и общечеловеческим ценностям и традициям, их сохранение и приумножение. А.П. Ершов неоднократно подчеркивал роль выдающихся людей в формировании молодого поколения. В своем дневнике, например, он записал о С.Л. Соболеве: «Вот человек, которому хочется следовать во всем без оглядки!» [13]. Выступая на методическом семинаре ВЦ СО АН СССР с докладом об активной жизненной позиции научного сотрудника, А.П. Ершов отметил, что «изучая биографии великих людей, не надо бояться восхищаться людьми. Человек, который способен восхищаться чужим успехом или достижением, освобождается от многих комплексов» [13]. Он неоднократно подчеркивал то научное и нравственное влияние, которое в свое время на него оказали выдающиеся ученые страны М.В. Келдыш, М.А. Лаврентьев, С.А. Лебедев и др. А.А. Ляпунову Андрей Петрович отводил особое место среди своих наставников и восхищался его «абсолютной незаинтересованностью укреплять авторитет общения внешними средствами... в сочетании с не менее абсолютной непреклонностью принципиальных положений» [14].

Одним из основополагающих признаков научной школы является не только существование яркого лидера, но и нескольких поколений – носителей «начального импульса». За двадцатилетний период деятельности А.П. Ершов подготовил 35 кандидатов наук [15]. Как минимум треть этих ученых впоследствии защитили докторские диссертации, продолжили работу в России и за рубежом, обрели своих учеников и последователей. Ядро научной школы информатики А.П. Ершова получило институциональное оформление и развитие. Напомним, что в конце 1950-х гг. отдел программирования первоначально был организован в Институте математики, затем вошел в состав Вычислительного центра Сибирского отделения АН СССР и, наконец, его подразделения послужили основой для создания в 1990 г. самостоятельного института – Института систем информатики СО РАН. Научное наследие А.П. Ершова продолжает оказывать влияние на развитие информатики и программирования как в стенах ИСИ СО РАН, так и за его пределами.

Работы А.П. Ершова оказали стимулирующее влияние на развитие теоретического программирования. В конце 1976 г. ученый пришел к идее смешанных вычислений. Он сформулировал понятие смешанного вычисления как фундаментального принципа системного программирования. Различные аспекты смешанных вычислений – от теоретических и даже философских до сугубо практических – нашли свое развитие в работах его коллег и учеников: М.А. Бульонкова, В.Э. Иткина (Новосибирск), Б.Н. Островского (Барнаул), В.К. Сабельфельда (Россия – Германия), Г.Я. Барздиня (Латвия) и др. В 1997 г. в ИСИ СО РАН появилась лаборатория смешанных вычислений, созданная на основе одноименной научно-исследовательской группы. Руководителем этой лаборатории является ученик А.П. Ершова канд. физ.-мат. наук М.А. Бульонков.

А.П. Ершов – один из основоположников применения графов в программировании. Стали классическими ранние труды ученого, в которых использовались теоретико-графовые методы в решении задач программирования [16]. А.П. Ершов считал, что графы являются основной конструкцией для программиста и обладают огромной, неисчерпаемой изобразительной силой, соизмеримой масштабу задачи программирования. В ИСИ СО РАН под руководством доктора физ.-мат. наук В.Н. Касьянова продолжают работы по применению графов в программировании. В частности, проводятся работы по анализу и систематизации теоретико-графовых алгоритмов и методов в информатике и программировании, завершается создание «энциклопедии» графовых алгоритмов для программистов, готовится ее издание на английском языке.

Академик А.П. Ершов предложил концепцию доказательного программирования, в которой обосновал принципиальную важность развития проблематики верификации программ и систем. В 1988 г. он писал: «Установлена принципиальная возможность придать программированию доказательный характер, снабжать программу сертификатом соответствия объявленным целям ее создания. Верификация программ – это

главный способ нахождения подобного рода сертификатов» [17]. Только через 15 лет известный английский ученый Ч.Э.Р. Хоар, предложив основной метод исследования верификации программ, подтвердил важность проблематики доказательного программирования. В течение последних лет основной проблематикой лаборатории теоретического программирования ИСИ СО РАН (канд. физ.-мат. наук В.А. Непомнящий) является разработка методов и средств верификации программ и систем. Сотрудники этого подразделения вносят существенный вклад в дело подготовки научной смены в институте. Среди 39 кандидатов и 5 докторов наук, подготовленных в ИСИ за 15 лет его существования, треть кандидатов и 1 доктор наук выросли как ученые в стенах лаборатории.

Создание систем программирования и связанные с ними теоретические исследования в значительной мере опираются на все предыдущие работы А.П. Ершова и его отдела в этой области (оптимизирующие трансляторы Альфа для М-220, Альфа-6 для БЭСМ-6 и т.д.). В лаборатории системного программирования ИСИ (руководитель – доктор физ.-мат. наук И.В. Поттосин, затем канд. техн. наук В.И. Шелехов) в 1990-е гг. по заказу Научно-производственного объединения прикладной механики имени академика М.Ф. Решетнева (Красноярск) создана система СОКРАТ для поддержки разработки эффективных и надежных программ для встроенных ЭВМ. Дальнейшее развитие методов анализа программ, в частности потокового анализа и трансформации программ, реализуется в настоящее время в рамках проекта по предикатному программированию. И таких примеров множество.

Еще одной отличительной особенностью Сибирской школы программирования является высокая социальная ответственность, то, что ранее называлось гражданской позицией. Лучше других осознавая, какой серьезный сдвиг произойдет в обществе в связи с применением компьютеров во всех сферах производства, А.П. Ершов стал неутомимым поборником идеи преподавания программирования и информатики в школе. Он одним из первых предложил ввести в школьное обучение курс «Основы информатики и вычислительной техники», составил необходимые учебные планы и написал первый учебник по этому курсу. Андрей Петрович принадлежал к тем ученым, которые формировали не только методику обучения школьному программированию, но и методологию многоступенчатого обучения компьютерной грамотности в стране. Основные направления научно-исследовательских работ в области школьной информатики, сложившиеся в 1980-е гг., актуальны и поныне. К ним принадлежат создание учебной литературы для изучения информатики и программирования; формирование системы ранней профориентации школьников, разработка информационных систем, поддерживающих постановку школьного предмета «информатика»; организационная, общественная и популяризаторская работа по созданию положительного отношения к информатике.

Характерным признаком научной школы А.П. Ершов считал ее включенность в мировое научное сообщество. Эта традиция продолжается в ИСИ СО РАН. В мае 1991 г. состоялась первая конференция «Перспективы систем информатики (PSI)», которая с тех пор организуется в среднем раз в три года, является крупным международным событием и проходит на высоком научном уровне. Конференция носит мемориальный характер и посвящается выдающемуся вкладу А.П. Ершова и его соратников в развитие информатики. В 2006 г. прошла Шестая международная конференция PSI, приуроченная к 75-летию со дня рождения А.П. Ершова. Цель конференции состояла в представлении и глубоком обсуждении передовых достижений в таких областях информатики, как теоретические основы анализа и разработки программ и систем; методология и технология программирования; новые информационные технологии; расширение контактов и обмен мнениями между учеными Востока и Запада. Труды конференций публикуются издательством Springer-Verlag в серии «Lecture Notes in Computer Science». Доклады, представленные в оргкомитет конфе-

ренции, проходят конкурсный отбор. Количество иностранных ученых обычно составляет третью часть от всех участников.

В жизненном цикле научной школы прослеживается прямая зависимость ее существования от способности выполнять новационные и образовательные функции [3]. В этом смысле Сибирская школа программирования находится на той стадии развития, когда производство знания и воспроизводство научных кадров корректируются новыми социальными условиями. В настоящее время сотрудники Института систем информатики готовят кадры на механико-математическом факультете и факультете информационных технологий в Новосибирском государственном университете. Лаборатории института стали базовыми площадками для студентов НГУ: ежегодно свыше 70 студентов и магистрантов защищают дипломы под научным руководством сотрудников института. В ИСИ активно функционирует аспирантура, работает диссертационный совет по специальности «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

Авторы выражают благодарность сотрудникам Института систем информатики имени А.П. Ершова СО РАН к.ф.-м.н. В.А. Непомнящему, д.ф.-м.н. И.Б. Вирбицкайте, д.ф.-м.н. В.Н. Касьянову, к.т.н. В.И. Шелехову, к.ф.-м.н. М.А. Бульонкову, к.ф.-м.н. Ф.А. Мурзину и Н.Ф. Воронко за информацию, предоставленную для данного доклада.

Библиографический список

1. Куперштох Н.А. Научные школы в России и Сибири: проблемы изучения // Философия науки. 2005. № 2 (25). С. 93–106. Статья размещена на сайте Отделения ГПНТБ СО РАН: <http://www.prometeus.nsc.ru/science/schools/docs/scisch.ssi>
2. Анкета научной школы. Академия управления при Президенте Республики Беларусь // <http://www.pacademy.edu.by>; Научные школы Пермского государственного университета; Девяткова Л.И. Академик Рем Викторович Хохлов – ректор Московского университета. Фрагменты книги // <http://www.phys.msu.su>.
3. Мирская Е.З. Научные школы: история, проблемы и перспективы // Науковедение и новые тенденции в развитии российской науки. М. 2005. С. 244–265.
4. Ведущие научные школы России. Вып. 1. М., 1998.
5. Ершов А.П., Шура-Бура М.Р. Становление программирования в СССР: Начальное развитие. Новосибирск, 1976. 49 с.; Ершов А.П., Шура-Бура М.Р. Становление программирования в СССР: Переход ко второму поколению языков и машин. Новосибирск, 1976. 42 с.; Публикации размещены на сайте Института систем информатики имени А.П. Ершова СО РАН (ИСИ СО РАН): <http://www.iis.nsk.su>.
6. Малиновский Б.Н. Очерки по истории компьютерной науки и техники в Украине. Киев, 1998. С. 452; Очерки истории информатики в России / Сост. Д.А. Поспелов, Я.И. Фет. Новосибирск, 1998; Алексеев А.С. М.А. Лаврентьев и информационно-вычислительные технологии // Наука в Сибири. 2000. № 44–45; Алексей Андреевич Ляпунов / Сост. Н.А. Ляпунова, Я.И. Фет. Новосибирск, 2001; Колмогоров и кибернетика / Сост. Д.А. Поспелов, Я.И. Фет. Новосибирск, 2001; Леонид Витальевич Канторович: человек и ученый / Сост. В.Л. Канторович, С.С. Кутателадзе, Я.И. Фет. Новосибирск, 2002; История информатики в России: ученые и их школы / Сост. В.Н. Захаров, Р.И. Подловченко, Я.И. Фет. М., 2003.
7. Поспелов Д.А. Становление информатики в России // Очерки истории информатики в России / Сост. Д.А. Поспелов, Я.И. Фет. Новосибирск, 1998. С. 8–9. С. 35–36.
8. Поттосин И.В. Андрей Петрович Ершов: жизнь и творчество // Ершов А.П. Избранные труды. Новосибирск, 1994. С. 5–29. Публикация размещена на сайте Института систем информатики имени А.П. Ершова СО РАН (ИСИ СО РАН): <http://www.iis.nsk.su>.

9. Становление новосибирской школы программирования. Мозаика воспоминаний. Новосибирск, 2001. 194 с.; Новосибирская школа программирования. Переключка времен. Новосибирск, 2004. 242 с. Публикации размещены на сайте Института систем информатики имени А.П. Ершова СО РАН (ИСИ СО РАН): <http://www.iis.nsk.su>.
10. Черемных Н.А., Курляндчик Г.В. Библиотека и архив академика Ершова // <http://www.iis.nsk.su>; Крайнева И.А., Черемных Н.А. Личный архив академика А. П. Ершова в Интернете // Отечественные архивы. 2001. № 5; и др.
11. Куперштох Н.А. История сибирской школы информатики академика А.П. Ершова // Модернизация российского образования. Новосибирск, 2005. С. 377–383. Бульонкова А.А., Крайнева И.А. Андрей Петрович Ершов: страницы жизни // Вопросы истории информатики. Вып. 5. Новосибирск, 2005. С. 34–44.
12. Ершов А.П. Дневник заведующего отделом // Архив А.П. Ершова. Папка 35. Л. 106.
13. Ершов А.П. Об активной позиции научного сотрудника // Архив А.П. Ершова. Папка 242. Л. 216.
14. Ершов А.П. Учитель // Очерки по истории информатики в России. Новосибирск, 1998. С. 194.
15. Список аспирантов А.П. Ершова, защитивших кандидатские диссертации // Архив А.П. Ершова. Папка 320. Л. 46.
16. О программировании арифметических операторов // Докл. АН СССР. 1958. Т. 118, № 3. С. 427–430. – Англ. пер.: On Programming of Arithmetic Operations // Commun. ACM. 1958. V. 1, № 8. P. 3–6; Сведение задачи распределения памяти при составлении программ к задаче раскраски вершин графов // Докл. АН СССР. 1962. Т. 142, № 4. С. 785–787.
17. Ершов А.П. Предисловие редактора // Непомнящий В.А., Рякин О.М. Прикладные методы верификации программ. Москва, 1988.

ИЗ ИСТОРИИ РАЗВИТИЯ САПР В СССР

Б. Н. Матюхин

Московский государственный университет приборостроения и информатики
Москва, РФ

Работы по САПР в Научно-исследовательском центре ВТ (НИЦЭВТ) со дня его основания в 1968 году возглавил Юрин О.Н.

Над этими проблемами работали тогда во многих НИИ и учебных институтах.

Первые работы были обобщены в книге «Применение ВМ для проектирования цифровых устройств», вышедшей в 1969 году под редакцией Матюхина Н.Я. Эта книга дала мощный толчок развитию САПР в СССР.

Проблемами автоматизации проектирования ЭВМ Юрин начал заниматься, работая еще в Конструкторском бюро промышленной автоматики (КБПА), из которого потом и создали НИЦЭВТ.

В отделе САПР, им возглавляемом, выделились основные направления работ:

- техническое проектирование – Розанов В.А., Малышева И.И.;
- моделирование на уровне микрокоманд и регистров на основе языка «ОСС-2» – Мясин О.Ф., Беляев В.Ф.;
- логическое моделирование на уровне вентилях и транзисторов – Вахнов Р.С.;
- построение тестов проверки – Зубаков Ю.А., Меркулов В.Г.;
- автоматизированный ввод и вывод машинной документации – Наголкин А.Н., Широкий Ю.П.

Эти работы легли в основу создаваемой Единой системы автоматизированного проектирования (ЕСАП).

Работы по созданию общей базы данных возглавили Лазарев А.И. и Наголкин А.Н. Она была названа «Главный информационный массив (ГИМ) ЕСАП».

С помощью ЕСАП была выпущена документация на многие серии ЕС ЭВМ.

В 1982 году группа ведущих специалистов по диагностике ушла из НИЦЭВТ в ЦНИИ «АГАТ».

Там под руководством Меркулова В.Г. и Матюхина Б.Н. была разработана и выпущена тестовая документация проверки на несколько типов специальных ЭВМ, использованная при их производстве.

ОСОБЕННОСТИ ПРОВЕДЕНИЯ РЕИНЖИНИРИНГА БИЗНЕС-ПРОЦЕССОВ В ТЕЛЕКОММУНИКАЦИОННЫХ КОМПАНИЯХ

Г.Г. Мирошников

Астраханский государственный университет

Москва, РФ

miroshnikov@mail.ru

Развитие рынка телекоммуникационных услуг вынуждает операторов связи сокращать время между внедрением технологий и предложением готовой услуги абонентам. Так, на начальном этапе предоставления новых услуг часто алгоритм их организации содержит ряд недоработок, для исправления которых в дальнейшем проводится реинжиниринг соответствующих бизнес-процессов. Основными его фазами являются: организация, документирование, анализ, изменение (инновация) бизнес-процессов, внедрение и управление. На этапе документирования особенно важным считается правильный выбор методологии моделирования бизнес-процессов, которая позволит наиболее полно и подробно описать механизм функционирования существующих бизнес-процессов, провести их анализ и оценку. Фаза изменения (инновации) в ряде случаев включает в себя реализацию автоматизированной системы, предназначенной для более эффективного выполнения бизнес-процессов.

Концептуальная модель подобных систем может быть построена с использованием той же методологии, что была выбрана на этапе документирования. Это позволит оптимально отразить все изменения еще до программной реализации. В предоставлении такой услуги, как широкополосный доступ к сети Интернет по технологиям xDSL, участвует, как правило, несколько отделов или служб телекоммуникационной компании. Наличие системы, объединяющей воедино информацию, необходимую этим структурам для выполнения своих рабочих функций, а также автоматизирующей некоторые операции, позволяет повысить эффективность выполнения соответствующих бизнес-процессов. Анализ алгоритма предоставления этой услуги показывает, что автоматизированная система должна хранить информацию о полном ее цикле: с момента подачи заявки до подключения конечного пользователя. Необходимым является и наличие в системе данных о емкости оборудования на каждом из участков его размещения, статусе каждой его единицы. Это позволяет на основе математических моделей сделать прогноз по задействованию оборудования и принять соответствующее решение об изменении его количества на том или ином участке, а также помогает в проведении необходимых маркетинговых исследований.

MATHEMATICAL MODELLING AND MODERN COMPUTER TECHNOLOGIES IN PLANNING RADIOTHERAPY OF THE CANCER OF THE BLADDER

E.V. Molchanova, L.Ya. Klepper

Petrozavodsk, Russia
molchanova@karelia.ru

Central Economic and Mathematical institute of the Russian Academy of Science (CEMI RAS)
Moscow, Russia

The basic purpose of radiation therapy (RT) consists of a choice of such conditions of irradiation of the center of tumoral disease which will lead to irreversible destruction of tumoral process without serious (irreversible) radiation complications in normal bodies and tissue of an organism. Conditions of the irradiation include a single dose of irradiation, a total dose (TD) and the scheme of realization of the irradiation in time. The determination of optimum conditions of irradiation presents the difficult problem. It is enough to point that to date the choice of a rational tumoral dose, TD, is realized on the base of collective and personal experience of the radiologists.

The purpose of research carried out by us consist:

1. In development mathematical models (MM) for the description of resorbition probability (RP) the center of tumoral disease as functions from type of tumoral cells, volume of a tumoral tissue and its conditions uniform (with a constant single dose) irradiations (a method of calculation of equivalent conditions of irradiation (CECI)).
2. In creation of MM for the description of tissue complication probability (TCP) in normal bodies and tissue as functions from volumes of tissue and conditions of their irradiation.
3. In use of the created MM for determination of an optimum dose of an irradiation of the center of tumoral disease and a box of the tumoral tissue maximizing criterion of successful radiation treatment $F=RP*(1-TCP)$. Here $(1-TCP)$ there is a probability of absence of tissue complication probability in a box tumours.
4. In application of the developed formalism for direct numerical calculation by means of the COMPUTER optimum TD for a cancer of a bladder.

Adjustment of mathematical models was made on the basis of clinical data about radiation treatment of a transitive-cellular cancer of a bladder which have been systematized in the Republican oncological clinic of Petrozavodsk. As a result of the lead researches the new, theoretically proved and practically confirmed data expressing dependence TD from RP and TCP for various volumes of a tumour of a transitive-cellular cancer of a bladder are received. They can be used as the methodical grant at planning RT of a transitive-cellular cancer of a bladder. The problem of determination optimum TD for a transitive-cellular cancer of a bladder is solved. It has made 49,5 Гр at the scheme of an irradiation: 2 times a day on 1,6 Gy for a session five times a week (RP reaches 71 %, TCP (a radiation cystitis) – 36 %, the volume of a tumour made no more than 5 % from volume of a bladder).

СЕМЕЙСТВО МАЛЫХ ЭВМ «НАИРИ»

Г.А. Оганян

Московский государственный институт радиотехники, электроники и автоматики (технический университет)
Москва, РФ

Высокий уровень математической школы, наличие высококвалифицированных конструкторов и технологов, развитая производственная база в Армении в середине 50-х годов прошлого века предопределили решение правительства СССР о создании в Ереване Института математических машин.

Основными задачами института стали научно-исследовательские, проектно-конструкторские разработки в области создания отечественных электронно-вычислительных машин (ЭВМ) и построение на их основе автоматизированных систем управления для использования в проектных, научно-исследовательских и производственных организациях, занимающихся обработкой информации и решением широкого круга проблем управления.

Первым директором института стал один из самых выдающихся математиков Советского Союза академик Сергей Мергелян. В институте были созданы функциональные подразделения по проектированию аппаратных и программных средств, включая их конструкторское и технологическое оснащение. Были сформированы цеха для изготовления образцов устройств и узлов, в том числе источников питания. Созданные цеха в дальнейшем объединились в опытное производство, а позже в Опытный завод ЕрНИИММ.

В начале 60-х годов прошлого столетия пост директора института занял Фаддей Тачатович Саркисян, нынешний президент Национальной академии Республика Армения, по инициативе которого были поставлены и решены крупные научно-технические, производственные и организационные задачи. В институте сформировались направления по созданию малых ЭВМ семейства «Наири», ряда моделей Единой серии универсальных ЭВМ (ЕС ЭВМ), а также научно-тематический центр по проектированию автоматизированных систем управления (АСУ) специального назначения. Был взят курс на крупносерийное производство средств вычислительной техники как на заводах Армении, так и на предприятиях других республик СССР.

Разработка первой модели машин семейства «Наири» была осуществлена в течение 1962–1964 гг. (главный конструктор Г.Е. Овсепян).

Принципиально новые схемотехнические решения, развитое программное обеспечение, ориентированное на решение возникающих в инженерной практике технических задач, позволили, с одной стороны, сформировать базовую архитектуру всего семейства малых ЭВМ «Наири», просуществовавших до начала XX века, а с другой – создать одну из самых распространенных в СССР малых ЭВМ, нашедшей широкое применение в научно-исследовательских институтах промышленности и высших учебных заведениях страны.

Разработанная архитектура ЭВМ «Наири» была запатентована в Англии, Японии, Франции, Италии.

Рассмотрим некоторые из запатентованных решений.

Впервые для класса малых ЭВМ в начале 60-х годов прошлого века была предложена 36-разрядная архитектура процессора. В целях минимизации аппаратных затрат арифметическое устройство состояло только из одного аппаратного 36-разрядного универсального регистра – сумматора. В качестве остальных регистров было предложено использование группы фиксированных ячеек оперативного запоминающего устройства машины. Для каждой фиксированной ячейки были определены микрооперации чтения и запоминания, что обеспечивало независимость от цикла всей оперативной памяти и резкое увеличение быстродействия.

Принципиальной особенностью предлагаемой архитектуры ЭВМ явилось использование постоянного долговременного запоминающего устройства кассетного типа, используемого в двух целях:

- 1) организация памяти микропрограмм;
- 2) память для хранения встроенного прикладного технологического программного обеспечения ЭВМ.

Объем каждой кассеты составлял 2048 36-разрядных слов. Необходимая разрядность микрокоманд (72 разряда) обеспечивалась путем одновременного считывания информации с двух кассет. Остальной объем ДЗУ (14 тысяч 36-разрядных слов) был выделен для хранения компиляторов с языков типа «Ассемблер» и «Бейсик», пакетов программ решения дифференциальных уравнений, всего спектра задач линейной алгебры, программ непосредственного счета разнообразных арифметических выражений в диалоговом режиме, программ управления пишущей машинки и перфоленточного ввода/вывода, построения графиков и диаграмм, а также комплекса технологических программ проверки всех узлов как на этапах производства, так и при эксплуатации ЭВМ.

Временные характеристики ДЗУ (цикл считывания 8 мкс) позволили реализовать весь спектр рассмотренных задач с временными характеристиками как минимум на порядок лучше, чем в зарубежных малых ЭВМ, в которых в качестве накопителей программного обеспечения использовались запоминающие устройства типа «магнитный барабан».

Крупносерийное производство ЭВМ «Наири» было организовано с конца 1964 года на двух заводах в Армении, а также на одном из самых крупных заводов СССР – Казанском заводе ЭВМ.

Наиболее ярко отличительные особенности первой модели семейства «Наири» проявились на юбилейной Лейпцигской ярмарке весной 1965 года.

На ярмарке демонстрировались малые ЭВМ различных фирм и стран (Англии – фирма ICL, Франции – фирма Bull, ФРГ – фирма Zuse и др.). ЭВМ «Наири» была единственной микропрограммной ЭВМ с расширенной разрядной сеткой (36 разрядов), обеспечивающей высокую производительность, повышенную точность вычислений (остальные ЭВМ имели разрядные сетки 8, 16 разрядов).

Быстродействие ЭВМ «Наири» было вне конкуренции, учитывая тот факт, что в ЭВМ «Наири» программное обеспечение было размещено в постоянной памяти с циклом выборки 8 мкс, а в остальных ЭВМ программы хранились во внешних запоминающих устройствах типа «магнитный барабан».

Следующей моделью семейства «Наири» стала «Наири-2», созданная в 1966 году. Основными особенностями этой ЭВМ явились увеличенная емкость оперативной памяти, использование более производительных устройств ввода-вывода и улучшенные конструктивные решения.

Сопоставляя ЭВМ «Наири» и «Наири-2» с нынешним уровнем вычислительной техники, можем отметить, что указанные ЭВМ в принципе сопоставимы по своим характеристикам (за исключением объемных характеристик), включая так называемый «дружественный интерфейс пользователя» с персональными ЭВМ архитектуры IBM PC, построенных на микропроцессорах Intel серии 486, серийно выпущенных 20 лет спустя.

Разработчиками ЭВМ «Наири-2» было получено дополнительно пять авторских свидетельств, включая свидетельство на изобретение, позволяющее проводить выполнение логических операций «и», «или» непосредственно в оперативной памяти, ее фиксированных ячейках, без какого-либо обращения к арифметическому устройству.

Создание в начале 70-х годов прошлого века серии семейства ЭВМ «Наири-3» («Наири 3-1», «Наири 3-2», «Наири 3-3») было продиктовано настоящей необходимостью системного использования ЭВМ в различных автоматизированных системах управления (многотерминальных системах коллективного пользования, системах автоматизированного управления производством радиоэлектронной аппаратуры и др.).

Значительное расширение области применения серии «Наири-3» потребовало пересмотра определенных компонентов архитектуры ЭВМ. Переход от области решения инженерных задач в область систем управления потребовало значительного расширения архитектуры ЭВМ, возможностей машинного языка, пересмотра системы ввода-вывода информации, перехода к многомашинным структурам, повышение достоверности обрабатываемой информации, реализации многотерминального доступа к источникам информации в режиме, близком к реальному времени.

При этом была поставлена задача максимального использования существующего в тот период времени программного обеспечения, включая пакеты прикладных программ, разработанного для других ЭВМ и широко используемого в находящихся в эксплуатации автоматизированных системах управления производством.

Успешная реализация поставленных задач была осуществлена благодаря использованию целого комплекса структурных и схемотехнических решений, апробированных в различных моделях серии «Наири-3» и защищенных многими авторскими свидетельствами.

Принципиально новым явилась организация микропрограммного устройства управления, имеющего двухуровневую структуру.

Первый – нижний – уровень представлял собой память минимальной емкости, в которой хранился набор всех необходимых микрокоманд, задающих все множество микроопераций, управляющих аппаратным контуром ЭВМ.

Второй уровень задавал последовательность адресов микрокоманд, реализующих микропрограммы выполняемых в ЭВМ машинных команд или некоторых процедур. В качестве памяти адресов микрокоманд при этом могли быть использованы любые типы запоминающих устройств ЭВМ.

Предложенная двухуровневая структура микропрограммного устройства управления позволила формировать в моделях серии «Наири-3» хранилище практически неограниченного набора микропрограмм. Как показали исследования, в предлагаемой структуре микропрограммного устройства управления была достигнута предельная плотность упаковки микропрограмм, приближающаяся к теоретическому пределу минимального кодирования информации.

Реализация в ЭВМ серии «Наири-3» практически неограниченного объема микропрограмм позволила впервые в СССР и одним из первых в мире практически реализовать методы полной микропрограммной эмуляции не одного, а одновременно нескольких машинных языков, функционирующих одновременно в ЭВМ серии «Наири-3». Выбор ЭВМ, подлежащих микропрограммной эмуляции, определялся объемом прикладных программ, разработанных в эмулируемой ЭВМ.

Так, в моделях серии «Наири-3» были реализованы машинные языки ЭВМ «Наири-1», «Наири-2», «Минск-22». При этом эмуляции подверглись не только отдельные команды, но и идиомы, представляющие некоторые последовательности машинных команд, имеющих определенное назначение. Например, команды управления внешней памятью (магнитные ленты). Микропрограммирование идиомы позволило в некоторых случаях повысить производительность эмулятора на один, два порядка.

Введение аппаратных средств многотерминального доступа к источникам информации (ЭВМ «Наири-3-2») позволило создать аппаратно-программную платформу автоматизированных систем управления производством.

Серия ЭВМ «Наири-3» нашла широкое применение в разных отраслях народного хозяйства страны.

Коллектив разработчиков указанной серии был удостоен Государственной премии СССР (Г.Е. Овсепян – главный конструктор, Ф.Т. Саркисян, Г.С. Сагоян, М.А. Хачатрян, М.Р. Буниatian, Х.К. Эйлезян, В.Г. Ишин, С.А. Туманян, А.В. Закиров), Премии Ленинского комсомола СССР (Г.А. Оганян, А.Г. Геолецян, Э.Л. Джанджулян, И.М.Ермаков, В.Г. Гончоян, Л.А.Карапетян, В.Г. Азатян, Г.К. Асланян).

По результатам разработки трех серий семейства «Наири» было защищено 11 кандидатских диссертаций.

Дальнейшее развитие семейства малых ЭВМ «Наири» воплотилось в создании серии ЭВМ «Наири-4», предметно-ориентированной, в основном, на специальные применения. Выбор архитектуры серии ЭВМ «Наири-4» определился требованиями значительного повышения производительности вычислений, подключения широкого спектра периферийных устройств (дисплеи, принтеры, аналого-цифровое оборудование), построения отказоустойчивых комплексов с встроенной системой аппаратного контроля и диагностики с широким использованием конструкторско-технологической базы, ориентированной на работу в жестких климатических условиях и механических воздействий.

При сохранении принципов двухуровневой организации микропрограммного устройства управления был сделан упор на резкое увеличение быстродействия первой ступени устройств-накопителей микрокоманд, задающей тактовую частоту ЭВМ. Учитывая малый объем этого накопителя, была предложена его организация на мультиплексорах, а в дальнейшем на ПЛИС.

Примером предметно-ориентированного подхода к построению систем явилось создание комплекса средств автоматизации труда конструкторов, технологов широкого профиля – ЭВМ «Наири-4/АРМ».

Наивысшая производительность в моделях серии «Наири-4» была достигнута в 1977 году в ЭВМ «Наири-41» (2 млн. операций в секунду).

При создании большой территориально-распределенной информационно-управляющей системы специального назначения возникла необходимость создания отказоустойчивого вычислительного комплекса, ориентированного на использование как для объектов системы верхнего звена управления систем, так и для массовых низовых объектов системы, являющихся источниками информации системы и образующих контур реального времени. При этом необходимо было реализовать:

- функции высоконадежного доступа к большому числу каналов связи, поступающих от разных направлений системы обмена данными;
- выполнение всех без исключения информационно-логических задач, низовых объектов системы.

При создании вычислительного комплекса на базе ЭВМ были предложены следующие системно-технические решения, отличающиеся патентной новизной:

- сквозной аппаратный контроль процессора и универсальной шины ввода-вывода;
- введение режима «горячего отключения», при котором допускается отключение сопрягаемых устройств ввода-вывода без остановки работы ЭВМ;
- введение в структуру комплекса блоков переключателей и связи каналов универсальной шины ЭВМ для построения многомашинных структур;
- микропрограммная реализация всех операций комплексирования ЭВМ и универсальной шины;
- введение в структуру ЭВМ и ВК специализированного диагностического процессора, на который возлагались функции диагностирования отказавших типовых элементов замены.

Универсальность архитектуры разработанного вычислительного комплекса, совместимого с одной из наиболее массовой в стране ЭВМ СМ-4, с использованием режима микропрограммной эмуляции, позволила реализовать большой класс телекоммуникационных и информационно-логических задач, возлагаемых на объекты системы.

Введение в структуру ВК диагностического процессора позволило резко уменьшить время восстановления ВК, повысив тем самым коэффициент готовности системы. Учитывая массовость низовых объектов, специфику их использования, связанную с территориальной удаленностью, предлагаемые системотехнические решения в совокупности со специальным программным обеспечением позволили создать систему, которая функционировала более 15 лет.

При создании серии ЭВМ «Наири-4» разработчиками было предложено 15 авторских свидетельств, защищено 9 кандидатских и 3 докторских диссертации.

Семейство ЭВМ «Наири», функционирующее почти сорок лет, по своим архитектурным, структурным, техническим и программным характеристикам явилось существенным вкладом в развитие вычислительной техники СССР и представляло оригинальную научную школу в области построения малых ЭВМ.

ЭВОЛЮЦИЯ ПРЕПОДАВАНИЯ КОМПЬЮТЕРНЫХ ДИСЦИПЛИН В МОСКОВСКИХ ВУЗАХ ЗА 40 ЛЕТ

О.В. Парахина

Московский государственный технологический университет «Станкин»

Москва, РФ

parakhina@yandex.ru

В докладе анализируется опыт преподавания компьютерных дисциплин в ряде московских вузов (МГУ, МГТУ «Станкин», МГАПИ (ВЗМИ), МГИУ (ВТУЗ-ЗИЛ) и др.). За 40 лет авторам довелось использовать в учебном процессе (в качестве студентов и преподавателей) различную компьютерную технику: «Урал-1», «Раздан», «Минск», «Днепр», «Наири», Siemens-4004, ЕС ЭВМ, «Электроника ДЗ-28» (15 ВСМ-5), ДВК, IBM-подобные персональные компьютеры и т.д. Огромные изменения произошли за это время и в содержании учебных курсов, и в принципах построения лабораторного практикума, и в техническом оснащении занятий. Мел и тряпка уступили место фломастерам, а затем ноутбукам и проекторам. Сейчас лекции сопровождаются мультимедийными демонстрациями, а студенты с 1-го курса владеют навыками профессионального поиска сетевой информации и использования специализированных программных пакетов. В 1995–2001 годах около 30 студентов вечернего отделения работали под руководством одного из авторов по тематике его лаборатории в академическом институте.

Доклад иллюстрирован фотоматериалами; для некоторых моделей компьютеров приводятся (справочно) их краткие характеристики и образцы программ.

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА, ПРЕДНАЗНАЧЕННОГО ДЛЯ РЕАЛИЗАЦИИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ БИЗНЕС-СИСТЕМ, ИСПОЛЬЗУЮЩИХ РЕЛЯЦИОННУЮ БАЗУ ДАННЫХ

С.Е. Парышев

ООО «Аплер Софтвер»

Москва, РФ

shyberg@mail.ru

В данной статье рассматриваются вопросы реализации объектно-ориентированных бизнес-систем [1]. Выделим основные признаки, характеризующие такие системы.

Бизнес-системы подразумевают необходимость долговременного (иногда несколько десятилетий) хранения данных. Таких данных обычно бывает много, и задача манипулирования этими данными является

одной из основных. Отсюда вытекает необходимость наличия СУБД, как правило, реляционной. Разумеется, это не единственный возможный вариант, но в настоящий момент именно он является одним из наиболее надежных, эффективных и проверенных временем.

Бизнес-системы, как правило, содержат множество различных вариантов «окон экранного интерфейса» [2], т.к. данные должны допускать возможность представления в самых разных формах.

Бизнес-системы редко существуют в изоляции. Обычно они требуют интеграции с другими системами.

Одной из основных задач бизнес-систем является описание и исполнение бизнес-процессов. Эти самые бизнес-процессы (бизнес-логика) порой представляются «как случайные наборы странных условий, которые сочетаются между собой самым непредсказуемым образом» [2] да еще и изменяются во времени.

В качестве примеров бизнес-систем можно привести такие системы, как интернет-сайты, сайты в интранетах, системы управления материально-техническим снабжением, системы управления информацией, системы выплаты заработной платы [2].

Виды объектов и способы их использования

В процессе практической разработки систем автоматизации различных видов бизнес-систем были выявлены определенные виды объектов, характерные для большинства из них.

- *Объекты, имеющие жизненный цикл.* Это основной вид объектов, любой бизнес-системы, т.к. именно объекты данного вида описывают основные бизнес-процессы предприятия, для которого стоит задача разработки системы автоматизации. Сложные объекты с жизненным циклом характеризуются двойственной природой. С одной стороны, они описывают свойства некоторых реальных объектов или процессов, а с другой – они описывают их состояние. При этом состояние во многих случаях управляет правилами изменения свойств. В качестве примера можно привести объект типа «заказ на изготовление полиграфической продукции» (все примеры здесь и далее взяты из системы автоматизации оперативной полиграфии). С одной стороны, заказ характеризуется такими свойствами, как дата создания, дата открытия, клиент, описание продукции, и многими другими. С другой стороны, заказ характеризуется набором состояний, например: «создан», «открыт», «отправлен в препресс», «аннулирован» и т.д. Требования к заполнению свойств во многом определяются текущим состоянием, например дата открытия имеет смысл только для заказа в состоянии «открыт». Связь правил заполнения свойств таких объектов с текущим состоянием объекта обеспечивается обычно путем использования контроллеров (в терминах шаблона проектирования Model-View-Controller).
- *Объект «Состояние».* Иначе – статус. В общем случае состояние сложного объекта с жизненным циклом представляется набором независимых (ортогональных) элементарных состояний. Например, объект типа «Заказ» имеет следующие независимые состояния: активности («Создан», «Открыт», «Завершен»), обеспеченности материалами («Не обеспечен», «Заказан», «В наличии»), оплаты («Не оплачено», «Оплачено частично», «Оплачено полностью») и некоторые другие. Элементарные статусы изменяются независимо друг от друга, либо через пользовательский интерфейс, либо автоматически в соответствии с заданными правилами. Данный вид объектов, как правило, является частью более сложного объекта и отражает один из разрезов его состояний. Таким образом, для сложного объекта с жизненным циклом можно определить произвольное количество независимых элементарных статусных объектов, которые являются неотъемлемой частью в данном случае объекта «Заказ». Это значит, что между данными объектами установлено отношение агрегации, т.е. данные объекты на протяжении всего жизненного цикла заказа существуют одновременно и одновременно исчезают в случае исчезновения заказа.

- *Объекты «Действия»*. Объекты данного вида предназначены для создания других объектов, изменения их состояния либо значения их атрибутов. Действия могут запускаться либо через интерфейс, либо автоматически в соответствии с заданными правилами. Использование объектов данного типа позволяет организовать управление потоком задач (workflow).
- *Справочные объекты*. Объекты данного вида могут быть как простыми, так и составными. В бизнес-системах справочные объекты носят вспомогательный характер. В качестве примера можно привести такой тип, как «клиент», «вид операции», «тип доставки». С помощью объектов данного вида можно значительно упростить и ускорить ввод информации в бизнес-систему.
- *Атрибуты*. Еще более простой случай справочных объектов, используются в качестве перечислимых типов для полей справочных объектов и других видов объектов.
- *Объект «Фильтр»*. Данный тип объектов позволяет формировать фильтры для различных списков. Например, список заказов можно фильтровать по дате создания, по статусу, по владельцу и т.д. В общем случае фильтр имеет визуальное представление в виде формы.
- *Объект «Отчет»*. Данный объект формирует определенный отчет по содержащимся в системе данным. Этот объект в большей степени предназначен для анализа накопленных данных.

Можно выделить два основных вида интерфейсов, используемых при реализации бизнес-систем, – списковые формы и формы редактирования.

Таким образом, для реализации бизнес-систем необходимо решить следующие основные задачи:

- отображение объектов предметной области, требующих долгосрочного хранения на реляционную модель;
- разрешение конфликтов одновременно с доступом к объекту;
- организация работы со списками объектов;
- управление состоянием объектов, имеющих жизненный цикл;
- интерфейсная реализация списков, форм правки, фильтрации, действий.

Уровень данных

Метаданные

В рамках данного программного комплекса разработана логическая структура метаданных. Она описывает структуру видов объектов разрабатываемой системы, связи между видами объектов, отображение объектной структуры на реляционную СУБД и обратно. Для того чтобы с помощью данной логической структуры стало возможно решать наибольшее количество задач, разработаны специальные шаблоны проектирования системы. Реализация в рамках некоторых из них автоматизирована с помощью автоматической генерации кода.

Конечно, реляционная и объектная модель имеют много отличий. Но существуют шаблоны для преобразования между ними. Данные шаблоны обобщены автором и адаптированы для описываемой методологии. Выведены дополнительные правила преобразования, которые охватывают некоторые аспекты бизнес-логики. Например, механизм пересчета объектов типа «Состояние» также вынесен на уровень метаданных. Более подробно об этом будет рассказано далее.

С помощью механизма автоматической генерации кода на основании метаданных мы получаем код классов, объекты которых могут прозрачно отображаться на базу данных с помощью специального

преобразователя данных. Объекты автоматически отслеживают свое состояние по отношению к базе данных. Мы всегда можем узнать, был ли объект изменен после загрузки, был ли он удален или он вовсе не был сохранен в БД.

Базовый класс реализован таким образом, что доступ к полям, описанным на уровне метаданных, осуществляется с помощью методов доступа по их строковому имени. Таким образом, каждый атрибут класса на самом деле представляет собой объект определенного класса. Эти объекты собраны в список. При таком подходе сложно отслеживать синтаксические ошибки, связанные с некорректным формированием имени атрибута.

Для решения вышеописанной проблемы используется автогенерация кода базовых классов для каждого объекта предметной области. Данные классы превращают свойства, реализованные методом отражения [2], в обычные свойства объектов. Таким образом, синтаксические ошибки именования атрибутов теперь будут отлавливаться еще на уровне компиляции.

Кроме того, данный подход делает возможным оснастить каждый такой класс своими собственными методами. То есть фактически мы получаем на выходе полноценный класс, объекты которого обладают набором очень важных дополнительных возможностей (сохранение в БД и т.д.).

Возможность обращения к полям по их строковому имени остается. Она необходима для некоторых аспектов разработки. В частности, для гибкого механизма взаимного пересчета полей, для более гибкого построения отчетов, возможности настройки таких отчетов без перекомпиляции кода, для построения критериальных запросов, для возможности автоматизации контроллеров (MVC [4]), возможно даже для полной автоматизации построения форм редактирования объектов и списковых форм в дальнейшем.

Преобразователи данных

Но этого недостаточно для разработки бизнес-приложений, т.к. мы получили отображение только простых объектов. А в реальной жизни часто встречаются сложные агрегированные структуры объектов, между объектами существуют связи и зависимости.

В данной версии программной инфраструктуры связи описываются с помощью специальных преобразователей данных. Такие преобразователи создаются для каждого графа объектов. Данные преобразователи реализуют определенный интерфейс и имеют базовую реализацию. Они реализуют отображение сложных объектных структур на базу данных и наоборот.

Особо следует отметить, что наличие метаданных открывает такие необходимые возможности, как преобразование объектов предметной области в XML, что необходимо для построения распределенных систем. Для этого в рамках программного комплекса разработаны специальные конверторы. Также существует возможность сохранения сложных объектов в виде XML как в файловой системе, так и в СУБД.

Так как программный комплекс в первую очередь предназначен для разработки корпоративных систем, или, как их еще называют, «учетно-отчетных» систем, то наличие метаданных помогает и для формирования отчетов. Благодаря наличию метаданных стало возможным отделить механизм формирования отчетов от основного приложения. То есть стало возможным формирование отчетов с помощью простых скриптовых языков, что не требует перекомпиляции основного приложения.

Блокировки

Системы управления базами данных, как правило, решают довольно большую часть проблем, связанных с совместным обращением к данным, но в большинстве случаев предметная структура настолько сложна, что данного механизма недостаточно. Для решения этой проблемы программный комплекс осна-

щен механизмом блокировок. Грубо говоря, если один пользователь открыл некий объект для редактирования, другие пользователи смогут получить доступ к данному объекту только на чтение.

Утилита

Для повышения производительности и уменьшения количества рутинного кодирования разработана визуальная среда редактирования метаданных и автоматической генерации стартового кода по метаданным (порядка 10–30% от общего кода). В данной среде описываются объекты системы, связи между ними, взаимное влияние объектов друг на друга (имеется в виду механизм автоматического пересчета состояний объектов). В настоящий момент в процессе реализации программных систем с применением данного программного комплекса выявлено еще некоторое количество аспектов, которые совершенно очевидно также можно вынести на уровень метаданных и автоматического формирования кода. В данном случае речь идет о контроллерах [4], о некоторых аспектах преобразования данных между моделями, о некоторых сложных типах (например, тип «Деньги»). Таким образом, можно еще около 15% кода вынести на уровень автогенерации.

Работа со списками

Для работы со списками объектов используются так называемые репозитории. Такие репозитории реализуются для каждого вида объектов предметной области. Объекты хранятся в репозитории в виде хеш-коллекции, ключом в которой, как правило, является GUID объекта. Вообще следует заметить, что в предлагаемой методологии для уникальной идентификации объектов используются GUIDы. Это позволяет однозначно идентифицировать объект как минимум в пределах базы данных. Также использование GUID позволяет хранить в одном репозитории объекты разных классов. Это бывает необходимо для решения некоторых задач. В частности, такой смешанный репозиторий удобно использовать для реализации визуального отображения иерархической структуры в пользовательском интерфейсе.

Репозиторий в некотором роде является отображением таблицы на модель предметной области. С помощью репозитория мы можем получить коллекцию объектов из конкретной, связанной с данным репозиторием таблицы. Это реализовано с помощью механизма обновления репозитория по заданному критерию. Механизм формирования критериев позволяет полностью избежать SQL запросов в коде программного приложения. Он позволяет осуществлять фильтрацию по базе и по уже загруженной коллекции одним и тем же образом.

В данной методологии существует как минимум две разновидности репозитория – локальные и глобальные. Локальные репозитории, как правило, используются для работы со списками объектов, постоянного доступа к которым из любого места программы не требуется. Глобальные репозитории, как правило, служат для формирования коллекций объектов, обращение к которым производится достаточно часто и при этом из разных модулей программы. Иначе можно назвать данные репозитории справочными.

Для глобальных репозитория также реализован механизм интеллектуального обновления. Иначе говоря, метод обновления необходимо вставлять в коде везде, где возможно обращение к данному репозиторию, но обновляться он будет только в том случае, если данные, содержащиеся в нем, потеряли свою актуальность. Это позволяет повысить производительность приложения и уменьшить трафик внутри сети.

Управление состоянием объектов

В бизнес-системах одной из основных задач является управление состоянием объектов. В данном программном комплексе эта возможность реализована в двух вариантах. В первом варианте на состояние объ-

екта, имеющего жизненный цикл, можно повлиять вручную, с помощью действий. Во втором случае состояние объекта может пересчитываться автоматически. Эта возможность реализована на уровне слоя хранения и метаданных. Для каждого типа объекта указывается список объектов, состояние которых зависит от состояния данного объекта. В дальнейшем при попытке сохранить или удалить объект, имеющий такой список, система автоматически пересчитает статусы зависимых объектов.

В качестве примера можно вернуться к типу объекта «Заказ». Ранее было описан объект «активность заказа». Значение этого статуса меняется по первому алгоритму, т.е. ручным способом. Также заказ имеет статус «состояние препресса». Данный статус зависит от состояний препресса по каждому изделию. И, таким образом, его можно реализовать вторым способом, указав у статусов изделий по препрессу в качестве объекта, на значение которого он влияет, статус «Состояние препресса».

Интерфейс

На основании выделенных типов объектов были разработаны шаблоны для основных видов форм, используемых в бизнес-системах. К таким шаблонам можно отнести:

- Шаблон списковых форм. Предназначен для реализации всевозможных списков объектов. Например, список заказов.
- Шаблоны правки объектов:
 - Шаблон правки агрегированного объекта. Имеется в виду объект, являющийся частью более сложного объекта. Его сохранение не приводит к сохранению в БД. Например, объект «Компонент», являющийся частью объекта «Заказ».
 - Шаблон правки основного узла объекта. Например, объект «Заказ». Его сохранение приводит к сохранению всей его структуры в БД.
- Шаблон фильтров. Формы данного типа предназначены для формирования условий выборки на списковых формах, для формирования условий, по которым строится отчет.

Управление *доступностью* и/или *необходимостью* заполнения тех или иных свойств объекта через форму правки реализуется за счет написания контроллеров форм. Контроллер – это промежуточное звено, обеспечивающее связь интерфейса и содержимого объекта. Одна форма редактирования объекта может иметь несколько различных контроллеров, используемых в разных состояниях объекта. Пример: доступ к объекту «Только на чтение» или доступ с ограниченным набором прав на изменение свойств.

Общая картина разработки в рамках данного ПК

Теперь подведем итог и обрисуем картину разработки в рамках данного ПК в целом.

Разработка объектно-ориентированных бизнес-систем с помощью данного программного комплекса ведется в рамках определенного процесса. Также как и в классическом варианте MDA, первым шагом является исследование предметной области и формирование словаря используемых понятий и условий функционирования системы, которые и составляют вычислительно-независимую модель (Computation Independent Model, далее CIM). Данная модель должна быть максимально общей и не должна включать в себя каких-либо деталей реализации. Для построения этой модели удобно использовать диаграмму классов UML. Можно провести аналогию между CIM и так называемой метафорой разработки в терминах экстремального программирования (К. Бек [3])

После того как СІМ сформирована, можно приступать к разработке платформенно-независимой модели (Platform Independent Model, далее PIM). В предложенном автором подходе это один из наиболее важных этапов. На данном этапе задача разработчика – выделить из предметной области те сущности (классы), которые требуют долгосрочного хранения в реляционной базе данных. На основе полученной модели СІМ необходимо построить модель базы данных. Модель базы данных является эталоном для создания классов предметной области. Тот момент, что за основу берется модель базы данных – одно из отличий описываемого подхода, но это не означает, что необходимо строить процесс именно таким образом. Разумеется, сначала можно построить модель метаданных (PIM) и уже исходя из нее сформировать структуру СУБД. Но дело в том, что в подходе, выбранным автором, гораздо проще учесть то, что мы имеем дело с реляционной моделью, а это накладывает определенные ограничения на структуру данных.

Таким образом, получается, что мы сначала строим на основании модели предметной области по определенным правилам модель базы данных, а затем на основании полученной модели строим метамодель нашей предметной области. Для этого разработана специальная утилита. Она позволяет построить мета-объекты по структуре базы данных. Далее, основываясь на метамодели предметной области, создаются метаобъекты для будущих классов предметной области. Затем с помощью той же утилиты генерируется исходный код для описанных классов.

Библиографический список

1. Макконнелл С. Совершенный код. Мастер-класс: пер. с англ. М.: Изд-во «Торговый дом "Русская редакция"»; СПб.: Питер, 2005.
2. Фаулер М. Архитектура корпоративных программных приложений: пер. с англ. М.: Издат. дом «Вильямс», 2004.
3. Бек К. Экстремальное программирование. СПб.: Питер, 2002.
4. Гамма Э., Хелм Р., Джонсон Р., Влисседес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001.

ПЕРСПЕКТИВЫ СТАНДАРТИЗАЦИИ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ В ОБРАЗОВАНИИ, НАУКЕ И ГОССЕКТОРЕ

Б.М. Позднеев

Московский государственный технологический университет «Станкин»

Москва, РФ

bmp@stankin.ru

В докладе проанализированы состояние и перспективы стандартизации информационно-коммуникационных технологий (ИКТ) в сфере образования, науки и государственного управления. Обоснована необходимость системной разработки национальных стандартов по ИКТ и их гармонизации с основополагающими международными стандартами. Представлены результаты взаимодействия национального технического комитета 461 «ИКТ в образовании» с 36-м Подкомитетом 1-го Объединенного Комитета ISO/IEC «Информационные технологии в образовании, обучении и тренинге».

ПРОБЛЕМЫ ФОРМИРОВАНИЯ ВИРТУАЛЬНЫХ КОМПЬЮТЕРНЫХ МУЗЕЕВ

Ю.Е. Поляк

Центральный экономико-математический институт РАН

Москва, РФ

vera.petrova@ecdl.ru

В последние годы в Интернете появилось огромное количество виртуальных музеев. Так, поиск по запросу Virtual Museum в поисковике Google дает свыше 31 млн. результатов, а Virtual Computer Museum – 6300000. В Рунете Яндекс находит более 150 тыс. документов со словами «виртуальный музей». Среди этих музеев есть как вполне солидные и серьезные (Виртуальный музей авиации – avia.russian.ee, музей деревянного зодчества – karely.narod.ru, музей Николая Рериха – www.roerich.ru), так и довольно экзотичные – музей печали – sorrow@hotmail.ru, «Белорусская соломка» – straw.iatp.by, музей сувенирных спичек – matches.yaroslavl.ru и т.п. Порой словом «музей» именуется подборка из нескольких фотоснимков.

В докладе делается попытка разграничить понятия «виртуальный музей», «сайт реального музея», «электронный альбом», «электронная публикация» и с этих позиций проанализировать организацию и информационное наполнение сайтов Intel Museum – www.intel.com/museum/, The Virtual Museum of Computing – <http://vmoc.museophile.com/>, Computer History Museum – www.computerhistory.org, Европейский виртуальный компьютерный музей – www.icfst.kiev.ua/museum/ или museum.e-ukraine.org/museum-map_r.html, Компьютерный музей Э. Пройдакова – www.computer-museum.ru и ряда других.

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ ОБУЧЕНИЯ В ТЕХНИЧЕСКИХ ВУЗАХ С 70-Х ГОДОВ ПРОШЛОГО ВЕКА ДО НАШИХ ДНЕЙ

И.В. Ретинская, В.С. Ретинский, И.М. Губкина

Американский благотворительный фонд поддержки информатизации образования и науки

Москва, Россия

kto@migmail.ru

Появление в начале 70-х годов прошлого века в Московском институте нефтехимической и газовой промышленности им. И.М. Губкина (ныне РГУ нефти и газа) первых советских компьютеров Наири-К, позволявших организовать с помощью примитивной консоли диалог с пользователем, сразу родило идею использовать их для простейшего программированного обучения и контроля знаний [1]. И хотя оперативная память этого компьютера составляла всего лишь 16 килобайт, с соответствующим быстродействием, оригинальная программа упаковки текста позволила осуществить эту идею и внедрить несколько контролирующих программ в учебный процесс. Этому способствовало и повсеместное увлечение идеями программированного обучения в вузах.

Следующий этап развития вычислительной техники в вузе был связан с внедрением в учебный процесс компьютеров серии ЕС с операционными системами ДОС и ОС, гораздо более мощных, но лишенных возможностей диалога (ввод данных – только с перфокарт, вывод информации – на алфавитно-цифровое печатающее устройство – АЦПУ). На этом этапе вычислительная техника использовалась студентами только для выполнения расчетных заданий.

Появление в начале 80-х годов надстроек над операционными системами – Фокус (разработка НИИВШ) и Primus (разработка МИФИ), позволившими поддерживать многотерминальную систему, обеспечивающую диалог с пользователем в режиме разделения времени, послужили катализатором для нового мощного развития идей компьютерного обучения. Началась разработка компьютерных обучающих и контролирующих программ как методом прямого программирования на различных языках (Фортран, Кобол, PL/I), так и с использованием первых появившихся оболочек или так называемых авторских систем (АОСВУЗ) [2]. Использование ЯОК (Языка Обучающих Курсов), разработки НИИВШ, позволило написать и внедрить в МИНХ и ГП более сорока обучающих и контролирующих программ по физике, математике, русскому и английскому языкам, химии, сопромату, основам программирования, языкам программирования, системному программированию, истории КПСС, деталям машин, технологии переработки нефти и газа и др. Язык обучающих курсов (аналог американской системы СПОК) [3], обладая некоторой процедурностью, позволял непрофессиональному программисту быстро создавать обучающие и контролирующие программы. На этом этапе был накоплен опыт обучения преподавателей, организации учебного процесса, развиты некоторые дидактические моменты, связанные с компьютерным обучением.

В 1988 году в вузе появились первые персональные компьютеры ДВК-3М. Были предприняты попытки конвертации компьютерных обучающих программ, разработанных для ЕС-ЭВМ на ДВК-3М. Следующие несколько лет – конец 80-х и начало 90-х – характеризуются вереницей сменяющегося оборудования и попытками переноса обучающих программ на новые платформы. Электроника, ЕС-1840, Mazovia, IBM XT, AT – вот не полный ряд платформ, пытавшихся завоевать этот рынок. В этот момент начинают развиваться отечественные оболочки для создания компьютерных учебных программ на персональных компьютерах, среди которых наиболее мощные и профессиональные – это Адонис (МИЭТ), АОСМИКРО (МИФИ), Урок и Сценарий [4]. Эти системы обладали такой большой функциональностью, которая позволяла создавать программы нового дидактического уровня: моделирующие, адаптивные, с широким набором возможностей анализа ответа обучаемого. Тем не менее на них практически невозможно было реализовать дидактические идеи нового уровня, позволяющие строить интеллектуальные обучающие системы с базами знаний, логическим анализом действий обучаемого. Хотя попытки создания таких оболочек предпринимались, например система ИСАП, написанная на Прологе (автор Токарева В.С., НИИВШ), но системы такого рода не были доведены до «коробочного» варианта, что затрудняло их широкое распространение. Тем не менее в Государственной академии нефти и газа им. И.М. Губкина была разработана совместно с НИИВШ и внедрена программа, обучающая языку Паскаль, снабженная элементами искусственного интеллекта.

В начале 90-х годов свежая струя в компьютерном обучении появилась благодаря распространению компьютеров с графическим интерфейсом и мультимедийными возможностями. Лидерами в такого рода технологиях были компьютеры Макинтош, появившиеся во многих вузах благодаря деятельности компании DPI. Компьютерные учебные программы, осуществляющие идеи обычного программированного обучения, но с мультимедийными возможностями, выполнялись в основном с помощью оболочки HYPERCARD, имевшей довольно ограниченные функциональные возможности.

Позднее они были вытеснены IBM-совместимыми компьютерами с теми же мультимедийными возможностями, но значительно более дешевыми. Некоторое время споры о преимуществах той или иной платформы продолжались [5], однако в середине 90-х годов они утихли.

В это время в вузах начала появляться возможность выхода в Интернет и вместе с ней появились качественно новые идеи компьютерного дистанционного обучения и новые программные средства, воплощающие эту идею.

Оболочки для дистанционного обучения разработаны практически для одной технической платформы и одного ряда операционных систем, которые стали в силу ряда причин стандартом де-факто в вузах России.

Компьютерные обучающие программы совершенно различного уровня технической и дидактической реализации заполнили информационное пространство Рунета, стали широко доступны всем желающим, так что потребовались дополнительные усилия для их анализа и систематизации [6].

Но это уже новейшая история.

Библиографический список

1. Ретинский В.С., Ретинская И.В. Применение малой ЭВМ «Наири К» для программированного обучения и контроля знаний // Программирование. 1975. № 2. С. 90–93.
2. Ретинский В.С. Методические указания по использованию автоматизированной обучающей системы АОС ВУЗ / МИНХ и ГП им. И.М. Губкина. М., 1984. 30 с.
3. Михалева Г.В., Ретинская И.В. Инструкция по составлению контролирующих и тренировочно-контролирующих курсов в АОСВУЗ / ГАНГ им. И.М. Губкина. М., 1986. 30 с.
4. Ретинская И.В., Шугрина М.В. Отечественные системы для создания компьютерных учебных курсов // Мир ПК. 1993. № 7. С. 55–60
5. Ретинская И.В., Шугрина М.В. IBM и Macintosh в сфере образования // Мир ПК. 1994. С. 90–98.
6. Образовательные Интернет-ресурсы / Под ред. А.Д. Иванникова, А.Н. Тихонова, В.Г. Домрачева, И.В. Ретинской. М., 2004, 286 с.

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ УПРАВЛЕНИИ ИССЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТЬЮ ШКОЛЬНИКОВ

А.А. Рогов

Петрозаводский государственный университет
Петрозаводск, РФ
rogov@psu.karelia.ru

В последние годы все большее число школьников имеют возможность заниматься исследовательской деятельностью, представлять свои работы на выставках, конференциях и конкурсах различного уровня. В целях повышения эффективности процесса организации мероприятий программы «Шаг в будущее» создана система информационной поддержки (СИП) научно-исследовательской деятельности учащихся Карелии. Она представляет собой единую электронную систему обеспечения процесса организации традиционных мероприятий, которая включает базу данных, программу по управлению базой данных и интерактивный сайт (<http://web.petrso.ru/~stf>).

Указанная система решает следующие задачи:

- регистрация участников научных соревнований школьников в интерактивном режиме;
- получение в автоматическом режиме всех информационных материалов, необходимых для обеспечения мероприятия (конференции или выставки) на основе информации заявки участника мероприятия;
- оповещение о результатах отборочных туров, об итогах выставок и конференций с помощью сайта;

- формирование некоторых глав ежегодного сборника информационных материалов «Научно-исследовательская работа школьников Карелии»;
- создание информационной базы для поддержки педагогических задач, связанных с исследовательской деятельностью учащихся, а также для связи с учащимися, общественностью и т.д.

При создании системы информационной поддержки научно-исследовательской деятельности учащихся на этапе разработки в первую очередь было уделено внимание структуре хранимых данных, которая сделала ее расширяемой без необходимости пересмотра имеющейся структуры данных и прикладных модулей, а также была предусмотрена интеграция с внешними источниками информации. При выборе средств реализации учитывалось, что система должна работать по принципу «клиент – сервер», что сможет обеспечить ее наибольшую гибкость, переносимость и масштабируемость.

Разработка предназначена для эксплуатации на персональных ЭВМ типа IBM PC и совместимых с ними в среде Windows, а также может применяться в локальной сети (например, корпоративной сети учебного заведения). Кроме того, реализован WEB-интерфейс доступа к единой БД научно-исследовательской деятельности учащихся Карелии по адресу в сети Интернет: <http://web.petrso.ru/~stf> (программы-браузеры должны быть типа IE 5.0. и выше, а также аналоги).

РАЗРАБОТКА ЭВМ БЭСМ-6

В.И. Смирнов

Институт точной механики и вычислительной техники РАН
Москва, РФ
svi@ipmce.ru

ЭВМ БЭСМ-6 была разработана в 1966 году коллективом Института точной механики и вычислительной техники Академии наук СССР под руководством академика Сергея Алексеевича Лебедева. В архитектуре машины нашли отражение как отечественный опыт вычислительных технологий, так и тенденции развития мировых технологий, что предопределило достижение производительности машины в миллион одноадресных операций в секунду и исключительно длительный срок ее эксплуатации.

При проектировании машины было применено моделирование ее детальной конвейерной структуры.

Система команд (без прямой адресации регистров) включала режимы прямой и косвенной адресации и адресации по стеку.

Были реализованы страничная организация и защита памяти, развитая система прерываний программ, режим защиты операционной системы и другие решения.

Удалось создать гибкую конвейерную структуру центрального процессора с большой пропускной способностью и глубоким просмотром команд вперед (до 6 и более команд) и высокопроизводительное арифметическое устройство с оригинальными алгоритмами выполнения арифметических и логических операций.

Основу системы элементов составляла ячейка с токовым переключателем, выполнявшим функцию запоминающего элемента с парафазным выходом и обширной пирамидой логических элементов на полупроводниковых диодах.

Парафазная синхронизация на основе двух синусоидальных сигналов частотой 10 МГц обеспечивала такт конвейера в 100 нс, что позволяло быстро загружать конвейер после ветвлений при выполнении программ.

Физическая конструкция ячеек и шкафов в виде развернутых панелей и система индикации обеспечивали полный визуальный обзор состояния каждого триггера и доступ к каждому логическому элементу.

Требование большой пропускной способности к системе оперативной памяти со стороны параллельно работающих арифметического устройства, устройства управления и каналов обмена данными с внешней памятью удовлетворялось за счет совмещенной работы восьми блоков оперативной памяти на магнитных сердечниках (с циклом 2 мкс), а затем на интегральных схемах и сверхбыстродействующей памяти ассоциативного типа на регистрах (с циклом 0,3 мкс). Ассоциативная память состояла из ассоциативного буфера команд, выбираемых из оперативной памяти, и ассоциативного буфера данных (результатов операций арифметического устройства), записываемых в оперативную память.

Внешняя память на магнитных барабанах и магнитных лентах, а затем и на магнитных дисках, подключалась через шесть параллельно работающих быстрых каналов.

Система тестирования и диагностики, позволившая добиться большого времени безсбойной работы машины и упрощения эксплуатации, создание для БЭСМ-6 самого обширного отечественного программного обеспечения обеспечили длительный срок ее работы в вычислительных центрах страны.

ОПЫТ ОБУЧЕНИЯ ИНФОРМАТИКЕ СТУДЕНТОВ МАТЕМАТИЧЕСКОГО ФАКУЛЬТЕТА ПЕТРОЗАВОДСКОГО УНИВЕРСИТЕТА

А.В. Соколов

Карельский научный центр РАН

Петрозаводск, РФ

avs@krc.karelia.ru

В докладе проводится сравнительный анализ преподавания таких языков программирования, как Алгол 60, Фортран, ПЛ/1, Паскаль, С++, Java, а также спецкурсов «Структуры данных», «Машинная графика», «Введение в параллельные вычисления».

Обсуждаются некоторые математические модели и алгоритмы оптимального управления такими базовыми структурами данных, как стеки, очереди, список свободных блоков разных длин.

НОВОСИБИРСКАЯ ШКОЛА ЮНЫХ ПРОГРАММИСТОВ

А.Г. Марчук, Т.И. Тихонова, Л.В. Городняя

Институт систем информатики им. А.П. Ершова СО РАН

Новосибирск, РФ

tanja@iis.nsk.su

Становление

Со времен академика М.А. Лаврентьева и чл.-корр. А.А. Ляпунова уделялось большое внимание школьникам Сибири. Их рассматривали в качестве главного потенциала для создания кадров сибирской науки. Создание в Сибири физико-математической школы, подготовка и проведение Всесибирской математической олимпиады, разработка и ведение оригинальных курсов для общеобразовательных школ различных

направлений способствовали решению задачи поиска талантливой молодежи. Видный советский математик и педагог А.А. Ляпунов проводил многообразную педагогическую работу, участвовал в составлении школьных курсов программирования и кибернетики.

По мере объединения усилий педагогов, математиков, программистов в области создания школьного курса, в котором рассматривались средства программирования и происходило знакомство с электронно-вычислительной машиной, началась деятельность двух инициативных групп. Одной из них была группа по применению вычислительной техники научного совета по проблемам образования при Президиуме Сибирского отделения АН СССР. Второй была группа школьной информатики Вычислительного центра Сибирского отделения АН СССР. В составе отдела экспериментальной информатики СО АН СССР было создано первое научно-исследовательское подразделение для формирования концепций и разработки программного обеспечения школьной информатики – группа школьной информатики. Научно-методический семинар «ЭВМ и учебный процесс» начал в стенах ВЦ СО АН СССР свою работу по объединению потенциала науки и образования.

Школьная информатика – начало

Когда заходит речь о школьной информатике, всегда вспоминается имя Андрея Петровича Ершова, личной энергией и обаянием направившего первые шаги движения за компьютерную грамотность.

В дневниках А.П. Ершова [1] можно найти подтверждение его неподдельного интереса к школьникам, относящееся к ноябрю 1961 года. «Сразу после праздников подошла ко мне пятерка симпатичных школьников. Дал им программу получения плоских графов и раскраски их путем выделения тетраэдров». К январю 1964 года относятся следующие записи: «Вчера приехал опять учитель Дорфман из Инской школы. Хотят с понедельника 27-го вести программирование в 10 классе их школы. Вместе с ним был зав. производственным обучением Сергей Генрихович Кубат...»

Первые занятия факультативного курса программирования с практическими работами школьников на ЭВМ относятся к началу 60-х годов уже прошлого столетия. Их проводили сотрудники Института математики, которые впоследствии стали сотрудниками Вычислительного центра СО АН СССР. Проводились эти занятия на базе школы № 10 г. Новосибирска.

В школе № 130 Новосибирского Академгородка было введено программирование как профориентационная специализация в 9–10 классах. Надо отметить, что эта форма обучения значительное время, вплоть до 90-х годов, продержалась в школах и получила достаточно широкое распространение по стране. В рамках этих занятий прошли апробацию несколько учебных курсов. В курсе программирования в школе № 130 прошли проверку курсы, базирующиеся на различных языках программирования. Этими языками стали Алгол, Бейсик, Фортран. В школе было установлено пять терминалов, которые обеспечивали связь в диалоговом режиме с машинами ВЦ СО АН СССР, кроме того, было устройство подготовки перфокарт для программ пакетного режима. В этом терминальном классе проводилась исследовательская и экспериментальная работа по применению вычислительной техники в школьном образовании. Это было не только обучение программированию на языке, но и, в частности, выполнение контрольных работ по английскому языку с использованием автоматической системы распознавания текстовых ответов (то, что сейчас относится к межпредметным связям).

Школы юных программистов

А.П. Ершов решительно поддержал тех, кто стоял у истоков школ юных программистов в Новосибирске. Невозможно рассматривать Школу юных программистов без первых единомышленников Андрея Пет-

ровича. Нина Александровна Садовская, аспирантка ВЦ, и Самуил Исакович Литерат, завуч 130-й школы, организовали и провели первую Летнюю школу юных программистов. К ним энергично присоединился Юрий Абрамович Первин.

Примерно через год (1977) Андрей Петрович пригласил из Харькова в Новосибирск Геннадия Анатольевича Звенигородского, который занялся организацией школы юных программистов (ШЮП). Он создал первую учебную среду программирования – «Школьница», выбрал ключевые идеи языков Робик и Рапира, прошагавшие благодаря широкому охвату Новосибирской школой бывшего Советского Союза по всем городам и весям. А.П. Ершов считал районную ШЮП [2] главной формой работы с ребятами из Академгородка. После трехлетней работы к 1981 году в ней насчитывалось около 200 школьников. Набор новичков производился с сентября. Это были школьники со второго по седьмой класс.

Очень серьезно рассматривался план работы, ориентированный на младших участников. В нем учитывали не только требования к языкам программирования, но также разрабатывали специализированное математическое обеспечение. В учебном плане стояли два теоретических занятия в неделю. Кроме этого в течение четверти школьники благодаря возможности, которую предоставлял Вычислительный центр СО АН СССР в утренние часы суббот и воскресений, выполняли две-три практические работы. В районной ШЮП были привлечены для преподавательской деятельности квалифицированные программисты из числа аспирантов ВЦ и НГУ, инженеры ВЦ, студенты-старшекурсники университета, выбравшие школьную информатику в качестве специализации.

Надо отметить, что на второй год обучения многие юные программисты переходили к решению задач в рамках производственных заказов институтов и предприятий. Например, была создана система анализа первичных структур белковых соединений, информационная система по книгообмену, программа подготовки перфолент для программно-управляемых вышивальных аппаратов. Помимо решения прикладных задач, школьники участвовали в реализации сложного математического обеспечения учебного процесса. Заметными работами было создание модулей транслятора Робика и Рапиры, системы машинной графики «Шпага» и др.

Вскоре (1978) в Академгородок переехала Нина Ароновна Юерман (Гейн), ныне она живет в Екатеринбурге и заведует кафедрой информатики в СУНЦе Екатеринбургского университета. Ею в дополнение к очной ШЮП с сентября 1979 года была развернута Заочная школа на страницах журнала «Квант», лучшие ученики которой приглашались летом в Новосибирск.

Система школ юных программистов, «Школьница», первый школьный учебник информатики, первый в нашей стране школьный кабинет, составленный из отечественных «Агатов» в школе № 166 Академгородка, – все это звенья одной цепи: зарождавшейся и бурно развивающейся школьной информатики. В этой цепи важным звеном была проводящаяся каждое лето в Новосибирском Академгородке начиная с 1976 года Летняя школа юных программистов.

Летняя школа юных программистов

Новосибирской Летней школе (ЛШ) есть чем гордиться. Раннее обучение информатике, ярко проявившееся в интенсивный период проведения ЛШ, отражает многочисленные достоинства этого подхода [3]. Существенную роль в отборе школьников для ЛШ оказал журнал «Квант». Благодаря олимпиаде, проводимой по окончании заочной ШЮП, выявлялись талантливые школьники, которые приглашались на две недели в круг единомышленников.

В течение пяти первых лет устоялась система занятий, которые проводились в ЛШ. Учащиеся распределялись по трем категориям: новички; школьники, знакомые с основами программирования; имеющие

опыт работы юные программисты. На деле оказалось, что двухнедельный срок проведения ЛШ достаточен для того, чтобы усвоить начинающим заниматься программированием школьникам основные понятия программирования и приобрести навыки работы на компьютере.

В рамках ЛШ проводились две конференции. Первая из них проводилась в начале школы и доклады на ней были в основном о работах, которые выполнялись школьниками дома. Вторая (заключительная) конференция предназначалась для рассказа о программах, написанных за две недели пребывания в ЛШ. Некоторые работы по решению председателей секций рекомендовались для использования в работе или публикации. Помимо этого, выполнялась не менее важная задача – увлечь ребят идеями информатики.

Среди любимых детьми преподавателей в ЛШ и одновременно известных в мире величин можно назвать А.Н. Терехова, Н.Н. Бровина, Л.Е. Штернберга, О.Ф. Титова, Ю.И. Брук и многих других, с энтузиазмом разрабатывавших первые уроки программирования для советских школьников. Нельзя не сказать об умении естественно и ненавязчиво применить на практике человеческий фактор работы в команде, взаимодействующей с внешним миром (позже – мастерские), акцентирование внимания на работу в команде по внешнему заказу.

Возможность выхода в большой мир, благодаря многочисленным участникам, прибывшим на ЛШ из городов всей страны и зарубежья (ЛШ с международным участием: ЧССР, Болгария, Польша, Германия, Голландия, Франция, Венгрия). На занятиях в ЛШ бывали люди, об общении с которыми можно было лишь мечтать (приезд Мак-Карти).

Летняя школа вырастила и дала путевку в жизнь большому числу специалистов-программистов, которые успешно работают в ведущих научных институтах СО РАН, в компьютерных фирмах: Microsoft, Intel, Exselsior...

Летняя школа юных программистов им. А.П. Ершова – день сегодняшний

С 2001 года ЛШ проводится во второй половине июля в течение двух недель. Это время удобно для привлечения в качестве преподавательского состава студентов и преподавателей НГУ, научных сотрудников институтов СО РАН. ЛШ является выездным мероприятием. Как правило, это близлежащие к Академгородку туристические и детские оздоровительные центры. Примерно раз в два-три года ЛШ выезжает на Алтай [4].

Основными задачами ЛШ является отбор талантливых старшеклассников, заинтересованных в овладении профессиональным программированием, обучение учеников среднего звена навыкам коллективной работы с применением современных информационных технологий и содействие развитию способностей к практическому программированию учащихся младших классов, а также поддержка педагогов, успешно преподающих информатику и программирование в общеобразовательной системе.

Часть детей приглашается по результатам работы в прошедших Летних школах. Остальные проходят через отбор на командной олимпиаде, мероприятиях программы «Молодые информатики Сибири», Новосибирской областной олимпиады школьников, областной научно-практической конференции школьников в секциях «Информатика» и «Программирование» и других мероприятиях. Каждый из участников проходит предварительное собеседование и заполняет специальную анкету, разработанную для участников Летней школы в ИСИ СО РАН завучем.

Мероприятия программы «Молодые информатики Сибири» ежегодно проводится в период с октября по март, олимпиада школьников – с ноября по январь в три этапа (районная, городская, областная). Областная научно-практическая конференция школьников проводится в апреле после отбора работ на районных научно-практических конференциях. При отборе участников ВКИ и СУНЦ НГУ учитываются результаты их учебы на специальных факультативных занятиях по программированию, проводимых членами оргкомитета. Отбор областных участников проводится по результатам участия школьников в олимпиадах, в конфе-

рениях, а также по результатам работы в Зимней школе по предметам, которые организует УО Администрации Новосибирской области. Младшие школьники отбираются на командной олимпиаде, проводимой на языке ЛОГО. В основном это ребята, которые учатся в школах, традиционно преподающих информатику с начальных классов (гимназии № 1 и 3, ЛИТ и некоторые другие).

Информация о вышеупомянутых мероприятиях распространяется через систему повышения квалификации школьных педагогов и семинары по проведению олимпиад по информатике для школьников совместно с районными методистами, размещается в газетах «Навигатор» и «Эпиграф», развешиваются информационные листки и плакаты в школах, раздаются буклеты на мероприятиях с участием иногородних преподавателей и школьников. Постоянно в течение года информация выкладывается на сайте Летней школы им. А.П. Ершова (ИСИ СО РАН).

Иногородние участники приезжают из Абакана, Ангарска (Иркутская область), Бердска, Искитима, Пашино, Кемерово, Новокузнецка, Омска, Павлодара (Казахстан), Горного Алтая, Алтайского края и других городов Сибирского региона. Стали принимать участие и школьники из Санкт-Петербурга и даже из-за рубежа (Норвегия).

В распоряжении ЛШ бывает порядка 25–30 компьютеров, предоставленных ИСИ СО РАН и участниками. К сожалению, компьютеры являются «разнообразными», их приходится собирать и готовить к ЛШ с большими трудозатратами.

Для обеспечения освещения работы Летней школы используются цифровые фотокамеры и видеокамеры, принадлежащие участникам.

Наиболее существенным прорывом в техническом обеспечении ЛШ стала надежная телефонная (мобильная и GPRS) связь. При подготовке следующих ЛШ надо учесть положительный опыт использования спутниковой телефонной связи на все время проведения занятий. Кроме связи с родственниками участников, будет перспектива развития школы в направлении дистанционного взаимодействия через Интернет со специалистами, не имеющими возможности выехать на базу ЛШ.

Мастерские как основа учебного процесса

На протяжении многих лет (с 1989 года) Новосибирские ЛШЮП проводятся как школы второй ступени с углубленным изучением отдельных предметов по выбору: в отличие от ряда летних школ в других городах, они имеют целью не начальное обучение основам компьютерной грамотности или программирования, а развитие профессиональной ориентации школьников, преимущественно старшего возраста. Спецификой организаторы ЛШ считают отбор участников Летней школы – преимущественно учащихся среднего звена.

Ребята знакомятся с программированием как с производственной деятельностью, с его проблематикой, методологией, творческими и технологическими аспектами. Новыми понятиями и объектами для изучения становятся программный продукт, технологический процесс разработки, грамотная постановка задачи и ее формализация, рациональное распределение и планирование работ, отладка, оформление, документирование, отчет. Для отработки этих понятий учебный процесс в Летней школе рассредоточивается по нескольким (10–15) учебно-производственным мастерским различных профилей – локальным носителям технологических циклов разработки, в которых школьники получают знания и навыки в процессе коллективной работы над единым проектом.

Таким образом, основной формой работы в ЛШ является выполнение поставленной задачи в рамках мастерской [5], где выполняется учебно-производственный процесс, дополненный общеобразовательным циклом. В общеобразовательный цикл входят лекции и спецкурсы по языкам и системам программирования, обзорные лекции по перспективам и проблемам программирования, истории информатики и дисциплин

линам, которые позволяют расширить кругозор учащихся во многих областях науки, а также ежедневная «Задача дня» – конкурс по решению алгоритмических задач. Бывает четырехчасовая командная олимпиада по программированию (по желанию). Ценным украшением учебы является цикл научно-познавательных лекций, читаемых видными академиками.

Учебное время экономится за счет совмещения по времени занятий по языкам программирования, спецкурсов и учебной работы в мастерских. Бывает также несколько традиционных «ликбезных» курсов по особенностям работы с компьютерами. Спектр мастерских формируется предельно разнообразным, на любой вкус и начальные знания. Совокупность тематики мастерских ЛШ подбирается так, чтобы лучше обеспечивать многопрофильность и разноуровневость учебного процесса с целью более адекватной его настройки на индивидуальные наклонности, интересы и способности учащихся.

Главной целью мастерской ставится полное прохождение всего технологического цикла в рамках поставленной задачи, с обязательным отчетом о проделанной работе в конце. Необходимая для этого интенсивность работ заставляет уделять большее внимание стадиям проектирования, как со стороны постановщика задачи, так и со стороны руководителя проекта и организаторов школы. Для многих мастеров, привлекавшихся к работе в ЛШ, оказывалась привлекательна именно возможность апробирования новых методик организации работ и обучения в условиях присущего школам дефицита времени и техники.

Целями вырабатываемой профессиональной ориентации учащихся ЛШ являются:

- расширение знаний учащихся о сферах и способах применения компьютерных технологий, типовых задачах и методах их решения;
- определение и уточнение учащимися области приложения своих способностей;
- приобретение специальных знаний и навыков, проба сил в коллективном проекте.

Жизнь мастерских

Предварительное распределение по мастерским организовано на основе анкетных данных и проводимых собеседований с учащимися. Учитывается желание школьников работать по той или иной тематической линии, о которых можно узнать на сайте Летней школы и по электронной почте.

В небольших группах под руководством опытных программистов-практиков дети работают над оригинальными проектами, параллельно осваивая новые компьютерные инструменты, технологии и приобретая бесценный опыт работы в команде. Задача мастера – не только научить, но и создать обстановку, чтобы каждый участник проекта развивался согласно своим интересам, возможностям и стартовому уровню. Этот уровень может быть различен, но неизменным требованием для участников ЛШЮП являются знание языков программирования и наличие навыков программирования.

Координацию учебной работы и мастерских ведет завуч с помощью главного мастера. В целом работа мастерских, по отдельности и в совокупности, расценивается как вполне успешная.

Для оценки работы мастерских и подведения итогов Летней школы работает жюри. Жюри в первую очередь оценивает рост уровня знаний и их качество, первоочередное значение имеет формулирование цели эксперимента и обоснованность полученного результата. Кроме того, оцениваются потребительские качества программного продукта: удобство пользовательского интерфейса, отладка, предоставление документации, презентабельность. Другими параметрами оценки бывают качество отчетов, понимание учащимися задач и состояния дел в своей мастерской, своей роли в ней.

Члены жюри смотрят результат работы мастерских на рабочих местах, чтобы реально представлять ход работы, беседуют с учащимися и мастерами о ходе работ. Промежуточные результаты работы в мас-

терских, проверяемые завучем и жюри до итоговой конференции, позволяют примерно оценить ожидаемые результаты, дать предварительные оценки работы всем учащимся, выявить слабые места в работе мастеров и вовремя их подкорректировать. Обычно все мастерские успевают подготовить демонстрационные версии, некоторые – довольно качественную документацию.

Итоговая конференция Летней школы проходит в лучших научных традициях. Докладчики со знанием дела представляют выполненные в мастерских проекты, участники конференции задают вопросы, содержание которых говорит о неподдельном интересе к представленным разработкам и о квалификации слушателей.

Общеизвестно, что любителей выступать среди программистов мало, большинство предпочитают безмолвно демонстрировать на компьютере результаты, качество которых зрителям трудно оценить. Но успех в любой профессии существенно зависит от искусства представления результатов. Поэтому участие в итоговой конференции обязательно для всех мастерских. На конференции рассматриваются полученные результаты и выбранные технические решения. Школьники обмениваются рекомендациями по улучшению их разработок. Важную роль играет личность председателя конференции, задающего уважительный стиль общения и обсуждения.

Эффективность Летней школы юных программистов

Для СО РАН проведение ЛШ является важным механизмом привлечения талантливой молодежи в сферу влияния науки и развития отечественной информационной индустрии.

Для родителей ЛШ удачно сочетает летний отдых школьников с получением интересных знаний и востребованных навыков.

Иногородние участники имеют возможность общения по интересам и повышения квалификации.

Для НГУ и в частности для кафедр «Программирование» и «Вычислительные системы» НГУ, а также ФИТ, ВКИ и СУНЦ НГУ проведение ЛШ дало следующие результаты:

- обкатка методик раннего обучения современной информатике;
- привлечение в НГУ абитуриентов, интересующихся программированием, способных в будущем участвовать в конкурсах и проектах НГУ;
- рост профессионального уровня студентов НГУ;
- привлечение к преподавательской деятельности студентов, приобретение студентами навыков работы в качестве руководителей проектов и постановщиков задач (бизнес-информатика).

Для Администрации Новосибирской области представляет интерес, что разрабатывается механизм выездной работы со школьниками, изучающими информатику. Этот механизм может быть распространен на сельские районы, способствует повышению уровня подготовки в вузы сельских участников.

Для ИСИ СО РАН существенно, что сотрудники, аспиранты и студенты ИСИ принимают участие в работе Летней школы, развивается эксперимент по обучению в форме мастерских, идея которых сформулирована и внедрена сотрудниками ИСИ. Выявлена заинтересованность молодежи в новых формах экспериментальной работы в области систем информатики, ведется эксперимент по организации Школы программирования (воскресной и вечерней, дистанционной) для наиболее подготовленных мастерских в течение учебного года, продолжает развиваться плановая тема «Исследование основ информатики и методов преподавания информатики и программирования», привлечено внимание к методическим наработкам.

Для информационной индустрии Новосибирска механизм ЛШ дает полигон для ранней профориентации школьников, а также для сочетания смены деятельности специалистов с вольным экспериментированием и поиском будущих помощников.

Педагогическая идея Летней школы доказала свою состоятельность почти тридцатилетней историей. ИСИ СО РАН бережно хранит традиции школы А.П. Ершова и развивает опыт успешного предпрофессионального обучения молодых талантов в области программирования.

Библиографический список

1. Дневник заведующего отделом // Архив А.П. Ершова. П. 35. Л. 124.
2. Ершов А.П., Звенигородский Г.А., Литерат С.И., Первин Ю.А. Работа со школьниками в области информатики: Опыт Сибирского отделения АН СССР // Математика в школе. 1981. № 1. С. 47–50.
3. Отчет о работе Летней школы юных программистов, проведенной с 14 по 28 августа 1987 года на базе туристского центра «Сибиряк»: <http://ershov.iis.nsk.su/archive/eaindex.asp?did=3936>
4. Отчет о работе Летней школы юных программистов 2004 г.: <http://school.iis.nsk.su/>
5. Марчук А.Г., Тихонова Т.И. Мастерская как форма обучения программированию // Сборник материалов XV Междунар. конф. «Информационные технологии в образовании». Москва, 2005. С. 48–49.

ОПЕРАЦИОННАЯ СИСТЕМА МНОГОМАШИННОГО ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА АС-6

И.Б. Бурдонов, В.П. Иванников, А.С. Косачев, С.Д. Кузнецов, А.Н. Томили

Институт системного программирования РАН

Москва, РФ

tom11@bk.ru

В 70-х годах прошлого века в Институте точной механики и вычислительной техники имени С.А. Лебедева была создана распределенная операционная система многомашинного информационно-вычислительного комплекса АС-6 (ОС АС-6).

ОС АС-6 состояла из взаимодействующих друг с другом через единый интерфейс равноправных операционных систем ЭВМ, входящих в комплекс; обеспечивала сетевое взаимодействие вычислительных процессов в ЭВМ комплекса, а также с процессами в глобальных сетях ЭВМ; обеспечивала для процессов, выполнявшихся в любых ЭВМ комплекса, использование внешних устройств всех ЭВМ комплекса, адресацию и использование внешних устройств в дислокациях; обеспечивала конвейерную работу входящих в комплекс машин («конвейер ЭВМ») для обработки в режиме реального времени больших потоков информации о полетах космических аппаратов.

В операционных системах ЭВМ комплекса имелись средства организации параллельных процессов, в том числе средства соподчинения задач и средства организации в задачах ветвей обработки информации.

Сетевые программные средства ОС АС-6 разделялись на транспортные и функциональные. Транспортные средства были предназначены для передачи данных в комплексе как между операционными системами, так и между задачами пользователей. Функциональные средства, базируясь на транспортных, выполняли функции запроса ресурсов, обмена с устройствами ввода-вывода, вызова программ-посредников и т.п.

В соответствии с принципом равноправия операционных систем, транспортные средства (в отличие от функциональных) были симметричны, т.е. каждая ОС обладала идентичными транспортными средствами.

Взаимодействие с транспортными средствами ОС какой-либо ЭВМ прекращалось в случае ее выхода из строя и автоматически возобновлялось после перезапуска ее ОС.

Транспортные средства, как правило, служили для связи объектов операционных систем разных ЭВМ. В ОС ЭВМ, специально предназначенной для управления внешними устройствами («Периферийная машина»), транспортные средства обеспечивали взаимодействие также и между внутрисистемными объектами.

Передача данных между вычислительными процессами могла происходить либо в режиме «почты», либо в режиме «телефона». В режиме «почты» ассоциация между процессами-абонентами возникала только в момент передачи «письма». В режиме «телефона» такая ассоциация коммутировалась и сохранялась в течение всего сеанса связи.

Распределением устройств между задачами, выполнявшимися в разных ЭВМ комплекса, занимались ОС-владельцы устройств. В них существовали специальные модули – администраторы ресурсов. Администраторы ресурсов взаимодействовали между собой, обмениваясь сообщениями через транспортные средства.

«ОГАС» В.М. ГЛУШКОВА: СОЦИАЛЬНЫЕ И КУЛЬТУРНЫЕ ПРЕДЕЛЫ ТЕХНИЧЕСКОГО ВООБРАЖЕНИЯ

А.Д. Трахтенберг

Институт философии и права УрО РАН
Екатеринбург, РФ
annat@sky.ru

В своей концепции Общегосударственной автоматизированной системы сбора и обработки информации для нужд учета, планирования и управления (ОГАС) В.М. Глушков во многом предвосхитил появление Интернета. Однако при этом он не смог выйти за пределы советских управленческих стереотипов и российского культурного пространства, что и сделало его проект утопическим.

О РАСКРАСКЕ ГРАФОВ

Турсунбай кызы Ырысгуль

Институт систем информатики им. А.П. Ершова СО РАН
Новосибирск, РФ
ryskul@gorodok.net

Задача раскраски вершин графа является одной из самых основных и наиболее изученных задач в теории графов. Изучение задачи о 4 красках для плоских графов породило интерес еще более 150 лет тому назад. Характерной особенностью этих задач является существование объектов, которые по каким-либо причинам не могут быть объединены в одну группу. К задачам раскраски сводятся многие вопросы, возникающие при планировании производства, составлении расписаний, хранении и перевозке товаров, автоматизации проектирования и динамическом распределении памяти ЭВМ и т.д. и имеющие широкое теоретическое и прикладное значение. Поскольку задача определения хроматического числа принадлежит классу полиномиально полных задач, исследования в этой области ведутся в разных направлениях.

В данной работе исследованы теоретические вопросы, возникающие в указанном направлении изучения задачи раскраски. Рассматриваются также задачи, связанные с некоторыми известными обобщениями задачи раскраски.

Раскраской графа G называется такое приписывание цветов его вершинам $V(G)$, что никакие две смежные вершины не получают одинакового цвета. Хроматическое число $\chi(G)$ графа G определяется как наименьшее количество цветов, необходимых для раскраски G , а раскраска G с $\chi(G)$ цветами называется оптимальной раскраской графа G . Задача раскраски графа заключается в нахождении оптимальной раскраски.

Результаты обзора литературы, посвященной алгоритмам раскраски графов, могут быть резюмированы следующим образом:

1. Все известные точные алгоритмы раскраски характеризуются чрезвычайно быстрым ростом времени реализации и объема запоминаемой информации при увеличении размерности решаемых задач, что позволяет использовать эти алгоритмы лишь для раскраски графов с числом вершин в пределах сотни.
2. Ни один из имеющихся приближенных алгоритмов раскраски не гарантирует близости найденного решения к оптимальному. Это объясняется принципиальными причинами: даже задача раскраски вершин графа в число красок, гарантированно меньшее удвоенного хроматического числа, является NP – трудной.

Принципиальные трудности, которые возникают при раскраске графа и нахождении его хроматического числа, вынуждают, во-первых, найти и исследовать практически интересные классы графов, для которых задача раскраски полиномиально разрешима, а во-вторых, вычислить или оценить хроматическое число графа с помощью других, более легко вычисляемых характеристик графа.

Наиболее интересным и практически важным классом, для которого задача раскраски полиномиально разрешима, является класс совершенных графов. С другой стороны, этот класс содержит множество практически интересных подклассов, для которых найдены эффективные алгоритмы раскраски.

КРОНОС: СЕМЕЙСТВО ПРОЦЕССОРОВ ДЛЯ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ

М.Я. Филиппова, Д.Н. Кузнецов, А.Е. Недоря, Е.В. Тарасов, В.Э. Филиппов

Институт систем информатики им. А.П. Ершова СО РАН

Новосибирск, РФ

flm@tech.ru

Историческая справка

Кронос – общее название семейства 32-разрядных процессоров, которые были предназначены для создания микро- и мини-ЭВМ. Архитектура процессоров Кронос ориентирована на поддержку языков программирования высокого уровня (Си, Модула-2, Паскаль, Оккам и т.п.), что позволяло реализовать новейшие концепции в области создания программного обеспечения и использования ЭВМ.

Семейство процессоров Кронос было разработано в новосибирском Вычислительном центре СО АН СССР в рамках проекта MAPC (Модульные асинхронные развиваемые системы) во второй половине 1980-х годов исследовательской группой KRONOS (Kronos research group) временного научно-технического коллектива (ВНТК) «СТАРТ». Руководитель д.ф.-м.н. Вадим Евгеньевич Котов. Ведущие разработчики Дмитрий Николаевич Кузнецов, Алексей Евгеньевич Недоря, Евгений Викторович Тарасов, Владимир Эдуардович

Филиппов. Впоследствии, в 1990 году, на базе ВНТК «СТАРТ» был образован Институт систем информатики имени А.П. Ершова СО РАН (<http://www.iis.nsk.su>).

Процессоры Кронос выпускались опытными партиями в основном для целей построения инструментальных машин для создания и отладки ПО. Основное применение нашли процессоры Кронос 2.6 при мелкосерийном производстве рабочей станции Кронос-2.6WS. Первый образец этой рабочей станции был продемонстрирован на выставке «Наука-88» в Москве в 1988 году. Рабочие станции Кронос-2.6WS нашли применение как инструментальные ЭВМ на ряде предприятий оборонного комплекса СССР, в частности использовались для создания бортового программного обеспечения для спутников в Научно-производственном объединении прикладной механики им. академика М.Ф. Решетнева (НПО ПМ, Красноярск-26).

В настоящее время образцы рабочей станции Кронос-2.6WS сохранились в Музее науки в Лондоне (Великобритания), в Политехническом музее в Москве, в музее Сибирского отделения Российской академии наук (Новосибирск), в Новосибирском госуниверситете (НГУ) и в ИСИ СО РАН в Новосибирске.

Проект КРОНОС

Цель разработки процессоров Кронос – создание универсального 32-разрядного процессора с поддержкой языков высокого уровня для создания ЭВМ открытой архитектуры: от встроенных микроЭВМ и однопроцессорных рабочих станций до многопроцессорных ЭВМ класса супермини.

В качестве базового языка разработки был выбран язык Модула-2, созданный Н. Виртом в 1975 году на базе языков Паскаль и Модула [1]. Унаследовав лучшие черты языка Паскаль, Модула-2 обладала рядом отличительных особенностей, которые делали ее наиболее пригодной для разработки и реализации программного обеспечения, отвечавшего требованиям того периода. К этим особенностям относились:

- модульность (введено понятие модуля и возможность его разбиения на определяющую и реализующую части);
- развитые структуры данных и управления;
- статическая типизация;
- наличие процедурных типов, что позволяет динамически параметризовать процедуры внешними действиями;
- наличие средств программирования низкого уровня, позволяющих ослабить жесткий контроль типов и отображать структурные данные на память.

Авторами был использован опыт разработки персональной ЭВМ Lilith, выполненной под руководством Н. Вирта в Высшей технической школе в Цюрихе, но многие решения были приняты достаточно независимо. Так, при почти неизменном наборе команд доступа к значениям в оперативной памяти, арифметико-логических операций и структур управления была существенно переработана архитектура взаимодействия процессов, прерываний, адресации и работы с внешними устройствами. Много упрощений было достигнуто за счет 32-разрядности процессора.

Наличие 32-разрядного машинного слова позволяло использовать процессоры семейства для решения вычислительных задач. Широкое адресное пространство (до 4 миллиардов слов) давало возможность создания виртуальной памяти для объектно-ориентированных моделей вычислений и тем самым обеспечивало разработку систем искусственного интеллекта. Аппаратная поддержка прерываний по событиям, по синхронизации процессов, а также компактность кода программ обеспечивали успешное применение процессоров семейства Кронос в системах реального времени.

Архитектура процессоров Кронос

Архитектура процессоров Кронос отличалась от традиционных. К основным отличительным чертам можно отнести следующие:

- 1) вычисление выражений производилось на быстром аппаратном стеке небольшой фиксированной глубины с сохранением содержимого стека при вызове процедур-функций и переключении процессов;
- 2) код и область данных любого процесса были разделены, следствием чего являлась повторновходимость всех программ и даже их отдельных частей (модулей);
- 3) отказ от абсолютной адресации даже в сегменте кода. Наличие таблицы смещения начала процедур упрощало их вызовы;
- 4) развитые виды адресации отражали понятия современных языков программирования. Имелась адресация локальных, глобальных, внешних объектов и объектов статически вложенных процедур;
- 5) метод межмодульной адресации позволял организовать динамическую загрузку – связывание – исполнение программ;
- 6) имелся развитый аппарат работы с мультисзначениями.

Для описания особенностей архитектуры процессоров Кронос потребуется ввести и определить некоторые понятия.

Программу, исполняемую в текущий момент процессором, будем называть процессом. Процесс состоит из следующих компонент:

- 1) таблицы загруженных в данный момент сегментов кода отдельно скомпилированных модулей [DFT];
- 2) глобальной области данных [G];
- 3) сегмента кода [F];
- 4) области строковых констант [STRINGS];
- 5) процедурного стека (П-стек) [P].

В скобках [] указаны базовые регистры, относительно которых происходит адресация внутри компонент.

Таблица DFT (data frame table) поддерживает динамическое связывание отдельно скомпилированных модулей.

П-стек, кроме того, размечен следующими указателями:

L – указатель на начало локальной области данных исполняемой процедуры;

S – начало свободной части П-стека;

H – ограничитель П-стека (предел увеличения S).

Переполнение П-стека (перекрытие S и H) отслеживается аппаратно с возбуждением соответствующего прерывания.

В Кронос-архитектуре поддерживались следующие способы адресации: локальный, глобальный, внешний, промежуточный и косвенный. Промежуточный способ адресации представлял собой обращение к нелокальным переменным внутри процедуры и осуществлял проход по статической цепочке процедур. Значения всех структурных переменных (массивов и записей) были представлены косвенно с помощью ссылки на начало выделенного им участка памяти. Структурные компоненты таких значений располагались внутри них непосредственно. Таким образом, удалось ограничить максимальный размер статической процедуры до 256 слов, т.е. процедура в терминах Модуль-2 не может содержать более 256 локальных переменных. Это казалось нам вполне разумным ограничением.

Выражения вычислялись на арифметическом стеке – А-стеке. Компилятор статически отслеживал переполнение А-стека и в случае необходимости сохранял его содержимое в памяти и восстанавливал его.

М-код представлял собой поток байтов, причем по коду операции всегда было известно, сколько байтов непосредственных операндов следовало за ним. Большинство команд однобайтовые. Извлечение команд и операндов из потока производилось с одновременным продвижением вперед счетчика команд РС. М-код имел четыре модификации длины непосредственного операнда: полубайтовый операнд, когда последние 4 бита команды интерпретируются как операнд; байтовый, двухбайтовый и четырехбайтовый операнды, представляемые следующими за кодом команды байтом, двумя или четырьмя байтами соответственно.

Система команд была построена так, что наиболее часто производимые операции имели более короткий код, что увеличивало скорость и снижало размер кода. Так, загрузку на стек выражений чисел 0, 1, ..., 15 выполняли команды LI0, LI1, ..., LI15 – Load Immediate, занимавшие вместе с операндом один байт кода, в то время как для загрузки на стек числа 153 использовалась уже команда LIB – Load Immediate Byte, занимавшая вместе с операндом два байта кода.

Процедуры представлялись своим номером из диапазона 0...255. Кодовый сегмент содержал таблицу смещений процедур относительно его начала. Поэтому команды вызова процедур также имели длину один (для процедур с номерами 0...15) или два байта. Механизм передачи параметров процедурам мог быть различным. Если у процедуры менее 7 параметров, их передача осуществлялась через арифметический стек, что сокращало расходы на обработку параметров в точке вызова. Все арифметико-логические операции работали над одним или двумя верхними элементами стека и помещали результат на место аргументов.

В начале П-стека всегда хранился так называемый дескриптор процесса, представленный копией всех регистров в момент последнего переключения на процесс. Имелась команда переключения процессов, которая сохраняла текущее состояние регистров в дескрипторе задерживаемого процесса, извлекала состояние регистров возобновляемого процесса и продолжала работу в возобновленном процессе. Этот аналог переключения сопрограмм служил мощным средством для организации мультипрограммной работы и синхронизации.

Все прерывания обрабатывались как переключения процессов, указатели которых находились в соответствующем номеру прерывания элементе массива векторов прерываний.

Семейство процессоров Кронос 2.X

Семейство процессоров КРОНОС 2.X включало три разработки: 2.2, 2.5, 2.6. У всех процессоров была одна система команд; они различались лишь по внутреннему функциональному устройству, быстродействию и конструктивному исполнению.

Логику функционирования всех блоков процессора реализовал блок микропрограммного управления. Две шины данных объединяли арифметико-логическое устройство, блок регистров, быстрый аппаратный стек на 7 слов, устройство выборки команд и буфер связи с шиной ввода/вывода. Двухшинная внутренняя структура процессора позволяла выполнять бинарные операции на стеке (сложение, вычитание, логические И, ИЛИ и т.д.) за один такт. Это существенно повышало производительность процессора на основном наборе команд. Микропрограммное управление упрощало устройство процессоров и давало возможность реализовать сложные команды.

Все узлы процессоров были выполнены на отечественных ТТЛ и ТТЛШ микросхемах средней степени интеграции широкого применения серий 155, 555, 531, 1802, 589, 556.

Кронос 2.2 являлся первой реализацией изложенных выше архитектурных концепций. Процессор был выполнен в виде платы в конструктиве ЭВМ «Электроника-60». Кронос 2.2 был полностью совместим со всеми устройствами, поддерживающими протокол обмена по шине Q-bus 22.

Процессор оперировал 32-разрядными словами. Поскольку шина 16-разрядная, доступ к слову данных в памяти происходил за два обращения по шине Q-bus 22. Арифметико-логическое устройство (АЛУ) Кронос 2.2 – 20-разрядное. Адресная арифметика обрабатывалась за один такт, а данные – за два такта. Такая внутренняя организация позволяла разместить весь процессор на одной плате при существовавшей тогда элементной базе. Объем прямо адресуемой оперативной памяти процессора составлял до 4 Мбайт, тактовая частота – 4 МГц, производительность – 600 тысяч операций в секунду над стеком.

В отличие от 2.2, Кронос 2.5 являлся полностью 32-разрядным. Он был выполнен в виде двух плат в конструктиве Intel. Интерфейс с внешними устройствами был реализован через шину Multibus-1. Еще одним существенным отличием являлось наличие локальной памяти объемом 0,5–2 Мбайт – в зависимости от применяемых микросхем. Тактовая частота процессора – 3 МГц, производительность – 1 млн простых операций в секунду над стеком.

Кронос 2.6 – дальнейшее развитие процессора 2.5. Он отличался конструктивным исполнением и гибкостью реконфигурации для конкретного применения. Процессор мог быть встроен как в отдельную мини-ЭВМ, так и в мультипроцессорный комплекс.

Кронос 2.6 был выполнен в конструктиве Евромеханика. Размер плат 233,3 x 220 мм (E2). В минимальный комплект входила плата обрабатывающего тракта (АЛУ, стек, регистры), плата микропрограммного управления, плата локальной памяти (0,5–2 Мбайт) и плата адаптера шины ввода/вывода. Все устройства были объединены локальной синхронной 32-разрядной шиной собственной разработки. Сам процессор не зависел от конкретной шины ввода/вывода. Настройка на конкретную шину производилась с помощью соответствующего адаптера. К локальной шине могли быть добавлены платы памяти, адаптера межпроцессорной связи, контроллера локальной сети, накопителя на магнитных дисках, плата памяти кода (при разделении плат кода и данных), bitmap-дисплея, арифметического вычислителя и других устройств, расширяющих возможности процессора. Тактовая частота процессора – 3 МГц, производительность – 1,5 млн/сек операций над стеком.

Операционная система Excelsior

ЭВМ с процессорами Кронос 2.2, Кронос 2.5 и Кронос 2.6 работали под управлением операционной системы Excelsior (ОС Excelsior), которая предназначалась для решения широкого круга задач. При создании ОС Excelsior разработчики руководствовались следующими принципами:

- открытость системы;
- модульность;
- интегрируемость результатов;
- удобство интерфейса.

Открытость системы

Необходимость обеспечить комфортабельную работу программиста выдвигала требование открытости системы, а именно:

- для работы любого ПО необходимы были только те программные и аппаратные средства, которые задействованы в нем непосредственно. Поясним на примере. Если программист создавал программу для встроенного применения, которое не предполагало работу с файлами и дисковой памятью, и не нуждался в имеющейся файловой системе, ему гарантировалась возможность получить способный функционировать как на рабочем месте, так и в целевых условиях программный продукт, в который не входила файловая система и все к ней прилагающееся;

- любое созданное программистом ПО автоматически наращивало программные средства и входило в их состав на равных правах с базовым ПО.

Модульность

Модульность программного обеспечения достигалась прежде всего свойствами языка Модула-2. Модульность ПО поддерживала, в свою очередь, динамическую загрузку задач. Динамическая загрузка позволяла исключить из цикла «редактирование – компиляция – запуск» этап сборки задачи и тем самым значительно ускоряла этот цикл. Кроме того, динамическая загрузка избавляла от необходимости хранить многочисленные копии кода часто используемых модулей, например библиотек. А отсутствие многочисленных копий кода, в свою очередь, давала ощутимую экономию внешней памяти (по сравнению с «линкованием» со стандартными библиотеками традиционных систем).

Интегрируемость

Примером внутренней интегрируемости являлись все программные средства ОС Excelsior, построенной как модульная расслоенная иерархическая система. Опыт эксплуатации машин КРОНОС как автоматизированных рабочих мест программиста в различных коллективах показал, что большая часть создаваемого программистами ПО выглядела не только как самостоятельный программный продукт, но и как набор библиотек, реализующих абстрактные типы данных. Эти библиотеки можно было использовать в другом программном обеспечении, они постоянно пополняли набор «строительных материалов», из которых программисты Кроноса складывали свои программы.

Удобство интерфейсов

Удобство интерфейсов – главнейший из принципов, положенный разработчиками во все разработанные системы. Поскольку в условиях оснащения оборудованием различных типов не удастся жестко ориентировать ПО на тот или иной его тип, нужны были самонастраивающиеся средства, максимально полно использующие возможности каждого конкретного вида оборудования, обслуживающего рабочее место программиста. Причем недопустимо было просто минимизировать возможности терминалов, клавиатур и других устройств, поскольку это отрицательно сказалось бы на удобстве работы программиста.

В ОС Excelsior эти проблемы преодолевались с помощью хорошо зарекомендовавшей себя техники абстрактных типов данных. Вся работа с конкретными устройствами была инкапсулирована в различные реализации одного и того же определяющего модуля, поддерживавшего интерфейс всех программных средств с устройствами такого класса. Поэтому как сама система, так и прикладное ПО оперировали с устройствами только в терминах такого определения. Этим дополнительно достигалось полное единообразие работы пользователя с различными (даже еще не знакомыми ему) программными средствами. Этому способствовала также возможность в любой момент работы единообразным для всех программных средств способом получить толковую и понятную подсказку на родном для программиста языке.

В заключение остается заметить, что все программные средства ОС Excelsior были ориентированы на мультипроцессное, многозадачное и многопользовательское применение. Благодаря модульной архитектуре достигался широкий спектр применения процессоров КРОНОС – от встроенных систем реального времени до супермини-ЭВМ. Самое замечательное свойство – легкость программирования (programmability) – делало Кронос незаменимым в приложениях, требовавших постоянного развития и изменения программного обеспечения.

Послесловие

В конце сентября – начале октября 2005 года Никлаус Вирт совершил большое турне по России. 21 сентября в Москве в Политехническом музее состоялась встреча Н. Вирта с участниками исследовательской группы KRONOS Владимиром и Мариной Филипповыми, Алексеем Недорей. От имени Института систем информатики СО РАН В. Филиппов передал в дар Политехническому музею действующую рабочую станцию Кронос-2.6WS. Этот экземпляр был предоставлен НПО ПМ им. М.Ф. Решетнева (г. Красноярск-26 или, в настоящее время, Железногорск), где использовался в 90-х годах в качестве инструментальной машины для создания бортового программного обеспечения для спутников.

Kronos processors family for high level languages

The KRONOS processors family was developed in the SB AS USSR Computing Center within the framework of the MARS project (Modulnye Asinkhronnye Razvivayemye Sistemy – Modular Asynchronous Developable Systems) in the period of 1983–1990.

The goal was to create a multi-purpose processor with hardware support of high-level languages for constructing open-architecture computers: from embedded microcomputers and single-processor workstations to multiprocessor workstations of the super-mini class. Presented below is the overview of the processors' architecture and the operating system for them.

KRONOS is the generic name for the family of 32-bit processors for creating micro- and mini-computers. The KRONOS processor architecture was oriented towards the support of high-level programming languages (C, Modula-2, Pascal, Occam, etc.) which allowed to implement the newest ideas in computer usage. Modula-2 was chosen to be used as the basic development language.

The project authors used creation experience of Lilith computer carried out under the guidance of N. Wirth in the ETH, Zurich; however, many of the decisions were made quite independently. For instance, while the sets of RAM value access commands, arithmetical and logical operations and control structures remained all but unaltered, there were substantial changes in the architectures of process interaction, interruptions, addressing and working with peripheral devices. Many simplifications were achieved thanks to the 32-bit processor.

32-bit machine word made it possible to use processors of the family for solving computational problems. The wide address space (up to 4 billion words) gave the possibility to create virtual memory for object-oriented computation models, thus enabling the development of AI systems.

The hardware support of event interruption and process synchronization together with the compactness of program code guaranteed the successful application of KRONOS processors in real-time systems.

The KRONOS processor family included three products: 2.2, 2.5, and 2.6. Computers with Kronos 2.2, Kronos 2.5, and Kronos 2.6 processors worked under the Excelsior operating system designed for solving a wide range of tasks.

The Kronos 2.6 WS workstations were used by a number of military industry companies. An example of notably fruitful and long-term cooperation was the joint project with M.F. Reshetnev NPO PM (Nauchno-Proizvodstvennoye Obyedineniye Prikladnoi Mekhaniki, Applied Mechanics Research and Production Group) in Krasnoyarsk-26. In October, 2005 an operational Kronos 2.6 worksta.

Библиографический список

1. Wirth N. Programming in Modula-2. Third, Corrected Edition. Springer-Verlag. Berlin; Heidelberg; New-York, 1985.
2. Wirth N. A Fast and Compact Compiler for Modula-2. ETH. Zurich, 1985.

3. Кузнецов Д.Н., Недоря А.Е., Осипов А.Е., Тарасов Е.В. Процессор Кронос в мультипроцессорной системе // Вычислительные системы и программное обеспечение / ВЦ СОАН СССР. Новосибирск, 1986. С. 13–19.
4. Кузнецов Д.Н., Недоря А.Е. Проектирование таблиц символов для языков со сложными правилами видимости // Методы трансляции и конструирования программ / ВЦ СОАН СССР. Новосибирск, 1984. С. 153–158.
5. Васекин В.А., Кузнецов Д.Н., Недоря А.Е., Тарасов Е.В. Кронос: аппаратная и программная поддержка языков высокого уровня // Тезисы докладов Всероссийского методического совещания-семинара по обмену опытом использования ЭВМ в учебной и научно-исследовательской работе студентов. Новосибирск, 1986. С. 92–94.
6. Кузнецов Д.Н., Недоря А.Е. Симфайлы как интерфейс операционной системы // Информатика. Технологические аспекты / ВЦ СОАН СССР. Новосибирск, 1987. С. 68–75.
7. Кузнецов Д.Н., Недоря А.Е., Тарасов Е.В., Филиппов В.Э. Кронос – автоматизированное рабочее место профессионального программиста // Автоматизированное рабочее место программиста. Новосибирск, 1988.
8. Кузнецов Д.Н., Недоря А.Е., Тарасов Е.В., Филиппов В.Э. КРОНОС: семейство процессоров для языков высокого уровня // Микропроцессорные средства и системы. Новосибирск, 1989.

НЕКОТОРЫЕ ВОПРОСЫ ФИЛОСОФИИ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ

Я.А. Хетагуров

МИФИ

Москва, РФ

chief@pcweek.ru

Более полстолетия в мире развиваются вычислительные средства, которые представляются двуединством вычислительных устройств и программирования. За этот период достигнуты фантастические успехи в создании вычислительных устройств (машин) и развитии программирования. Вычислительные средства получили широчайшее использование в различных областях нашей жизни. Опыт развития вычислительных средств дает возможность анализировать принятые и принимаемые решения и рассматривать используемые оценки их рациональности. Представляется, что оценка рациональности решения во многом определяется условиями развития и целями, преследуемыми применением вычислительных средств.

Рассмотрим связь цели применения и условий развития ВС. Если на начальном этапе развития области применения вычислительных средств ограничивались научными и, главным образом, военными целями и имели относительно простые оценки рациональности путей их решения – учитывались два фактора: время получения решения и затраты, то внедрение вычислительных средств в различные области современного общества существенно расширили, усложнили и увеличили требования к ним и к оценкам рациональности их применения.

История развития вычислительных средств показала, что задача увеличения объемов производства вычислительной техники оказалась существенно более простой для реализации, чем обеспечение областей программами решения задач. Это обстоятельство оказало серьезное влияние на развитие вычислительных средств. Они стали совершенствоваться в направлении повышения основных характеристик вычислительных средств – вычислительных устройств (быстродействие и объемы памяти) и производительности программирования (языки высокого уровня).

Для вычислительных устройств повышение быстродействия и объемов памяти главным образом обеспечивалось технологическими средствами путем сокращения размеров элементов, что увеличивало частоту их работы и уменьшало размеры устройства, а также посредством повышения числа параллельно работающих вычислительных устройств. Повышение производительности программирования обеспечивалось широким использованием языков высокого уровня, учитывающих особенности решения задач в различных областях применения вычислительных средств. Использование языков высокого уровня существенно повышало производительность программирования задач, но преобразование программы на языке высокого уровня в программу работы вычислительного устройства (в программу на машинном языке МЯ)) приводило к значительному увеличению ее объема. Суммируя приведенные данные, можно в широком смысле оценивать развитие вычислительных средств как экстенсивное и соответственно определять его роль в экономическом развитии общества.

Экстенсивное развитие вычислительных средств ярко выявило положительную обратную связь двух частей вычислительных средств – вычислительных устройств и программирования. Так, увеличение объема программ ставит задачи по повышению производительности вычислительных устройств, а их совершенствование приводит к возможности увеличения программного обеспечения. Таким образом, эти две части создают условия друг другу для развития, что способствует сохранению существующего паритета в распределении и развитии работ и четкой экономической заинтересованности сторон. В результате экстенсивное развитие вычислительных средств отрицательно влияет на генерацию и внедрение новых идей в построении вычислительных средств.

Важной проблемой развития вычислительных средств является оценка соотношения видов и числа операций у ЯВУ и МЯ и их выбор для построения вычислительных средств. По-видимому, целесообразно провести анализ движения от МЯ к ЯВУ, предполагая аппаратную реализацию каждого шага этого движения. Учитывая признанные преимущества программирования на ЯВУ, с одной стороны, и достижения в технологии минимизации размеров элементов на кристалле, с другой стороны, а также развитие САПРа, практически созданы условия для построения сложных схем операции ЯВУ. Однако, по публикациям, это направление ВС не получило развития, по-видимому, из-за корпоративных соображений, так как резко ограничивался рынок программных продуктов на МЯ, с одной стороны, и существенно изменялись требования к быстродействию вычислительных устройств, с другой стороны (сокращение количества команд на МЯ более чем на порядок, практическое исключение ОСов).

Необходимо отметить, что применение ВС с аппаратной реализацией ЯВУ существенно сокращает затраты фирм, создающих системы с вычислительными устройствами как на программирование и отладку программ и всей системы, так и на эксплуатацию, но это проблемы другой группы фирм.

Следующей важной проблемой развития ВС является оценка универсальности создаваемых вычислительных устройств – имеется в виду возможность эффективно проводить вычисления для широкого круга задач. Для подавляющего числа пользователей ВС (более 80%) и особенно для ВС, работающих в различных областях, главным является минимизация затрат на эксплуатацию. Это достигается двумя путями:

- использованием универсальных вычислительных устройств и широким применением разработанных типовых программ;
- применением специализированных вычислительных устройств, эффективно решающих определенные группы задач и использующих наиболее удобные ЯВУ.

Современный уровень развития технологии изготовления вычислительных устройств, использование САПР, создание базовых технических решений (базовые кристаллы для микропроцессоров,

контроллеров) изменяют оценки целесообразности применения в сторону специализированных решений.

Существующее распределение промышленных и экономических ресурсов ведущих производителей ВС больше заинтересовано в универсальных ВС, так как они обеспечивают экстенсивное развитие и определенный уровень прибыли и паритет между частями ВС. Хотя для потребителей, эксплуатирующих ВС, это приводит к увеличению затрат. Но так организовано взаимодействие в производстве ВС. Это один из важных факторов, сдерживающих реализацию новых идей.

Проблемой развития ВС, несмотря на определенные достижения, является обеспечение достоверности выдаваемой информации. Как известно, достоверность информации (отсутствие ошибок в выдаваемых данных) определяется качеством используемых методов контроля правильности работы вычислительных устройств и выполняемой программы. Необходимо отметить, что важность требований к достоверности информации определяется ограниченным числом потребителей ВС (15–20%), однако ущерб от ошибочной информации может достигать значительных величин. К ошибочной информации, с точки зрения потребителя, отнесем и воздействие вирусов и различных закладок, которые наносят миллиардные ущербы.

Рассмотрим некоторые особенности появления вирусов, которые нарушают порядок выполнения программ решения задач. Необходимо отметить, что появление вирусов в программе не связано с нарушением работы аппаратуры вычислительного устройства, принимаемые меры по устранению влияния вируса осуществляются после его обнаружения и действуют только на этот вирус. Предсказать появление вируса пока, по опубликованным данным, не удавалось. Анализируя программы вирусов, можно отметить, что в них используются команды, связанные с преобразованием адресов и изменениями данных. Возможное число сочетаний подобных команд в программах практически неограниченно. Для исключения влияния вирусов необходимы изменения организации работы вычислительного устройства и условия формирования программ. Поэтому существующий порядок работы ВС периодически будет нарушаться появлением очередного вируса, воздействие которого устраняется созданием очередной антивирусной программы, что связано с дополнительными затратами пользователя и увеличением объема программы и повышением требований к быстродействию.

Существующий экстенсивный порядок развития ВС, определенный главным образом экономическими отношениями между ведущими компаниями мира, может измениться только в результате появления новых условий и потребностей общества.

PICLAB – THE LABORATORY OF NONLINEAR IMAGE PROCESSING ALGORITHMS

V.V. Hryachev

Yaroslavl State University

Yaroslavl, Russia

dcslab@uniyar.ac.ru

In view of the expansion of digital image processing applications (telecommunications, digital video, radio-astronomy, medicine images, etc.), the actual problem is to design software packages for solving scientific and practical tasks and also for training specialists in this sphere. At the same time growing complexity of the modern nonlinear algorithms requires new, more illustrative methods for their visualizations.

To perform scientific and practical tasks and also for educational purposes the digital image processing software package PICLAB («PICTure LABoratory») was designed. The important feature of the PICLAB is the possibility to analysis image from different points of view, both from objective estimates (like signal/noise ratio, mean square error etc.), and also from visual criteria. The paper will give the overview of PICLAB and it's scientific and educational applications.

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА В ОБЛАСТИ ИСТОРИИ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В ПОЛИТЕХНИЧЕСКОМ МУЗЕЕ

Н.В. Чечель

Политехнический музей
Москва, РФ
chachel@polymus.ru

Политехнический музей – научно-методический центр Российской Федерации (научно-методическая база Федерального агентства по культуре и кинематографии) в области истории науки и техники для научно-технических музеев независимо от их ведомственной принадлежности, а также отделов исторических и краеведческих музеев Российской Федерации, имеющих в своем составе коллекции или предметы науки и техники. Музей по своей природе входит в систему научно-исследовательских учреждений. К научно-исследовательской деятельности науковедение относит «особый вид человеческого труда, представляющий совокупность процессов добывания, распространения и применения таких знаний, которыми общество ранее не располагало», т.е. любая исследовательская деятельность, результатом которой является новая информация, полученная при изучении того или иного объекта. Выявляя и собирая предметы музейного значения, храня музейные предметы, создавая экспозиции и проводя просветительскую работу, музеи не могут ограничиваться результатами изысканий, проводимых другими организациями. Им необходимо вести собственные научные исследования, на которые в конечном итоге опирается вся деятельность музея – научно-фондовая, экспозиционная, просветительская и образовательная.

Научно-исследовательская работа Политехнического музея определяется его статусом научно-исследовательского учреждения, имеющего лицензию на проведение этой деятельности. Заключается она в получении новых знаний на основе изучения музейного собрания, включает исследования по профильным дисциплинам – истории науки и техники и музееведческие исследования. Проводимые исследования в области истории науки и техники опираются на фонды, носят источниковедческий характер и направлены на выявление документирующих свойств музейных предметов и их значение в развитии науки и техники. Музейные предметы в этих исследованиях изучаются с точки зрения того нового знания, которое они могут дать истории техники. Музей обобщает новые данные и вводит их в систему сложившихся научных представлений. Музеефицирование вычислительной техники входит в число приоритетных задач Политехнического музея. Коллекции счетных приборов и вычислительной техники собираются музеем более 130 лет, за это время ПМ стал обладателем лучшим и самым полным собранием музейных предметов по истории вычислительной техники.

ИСТОРИЯ И ПЕРСПЕКТИВЫ СЕТИ ПЕРЕДАЧИ ДАННЫХ СО РАН

Л.Б. Чубаров, В.С. Никульцев, С.Д. Белов, Е.В. Никульцева

Институт вычислительных технологий СО РАН

Новосибирск, РФ

chubarov@ict.nsc.ru

Организации удаленного доступа к вычислительным ресурсам и распределенной обработке данных уделялось значительное внимание на протяжении всей истории развития вычислительных технологий в СО АН СССР, а затем в СО РАН. На самых ранних этапах рассматриваемого периода, в отсутствие общепринятых решений для построения информационно-вычислительных сетей общего пользования в современном смысле, основные усилия были направлены на построение специализированных проблемно-ориентированных распределенных вычислительных структур – систем автоматизации эксперимента, систем управления, систем терминального доступа к централизованным вычислительным ресурсам. При их построении использовались разработанные на местах системы передачи данных, собственные протоколы различных уровней, приводящие к уникальным, малоприспособленным для тиражирования и для интеграции решениям.

Тем не менее общеакадемические проекты второй половины 70-х и начала 80-х годов – «Академсеть», ВЦ КП – заложили основы сетевой инфраструктуры СО РАН и ННЦ. К сожалению, работы по упомянутым проектам не были достаточно скоординированы. Ориентация проектов на терминальный доступ с использованием выносных терминалов, использование на транспортном уровне технологий X.25 и непродуманность перспектив дальнейшего развития привели практически к остановке работ по этим проектам и к тому, что значительная часть инфраструктуры была просто-напросто заброшена. Появление в 80-х годах значительного количества мини- и микроЭВМ, совместимых с семейством PDP-11 компании DEC, до некоторой степени стимулировало интерес к сетевой проблематике, но прорабатывавшиеся решения были, как правило, ограничены построением сети отдельной организации или даже отдельных подразделений и не могли рассматриваться в качестве универсального подхода к построению распределенной гетерогенной структуры.

Работы по созданию прототипа того, что стало впоследствии сетью Академгородка, Новосибирского научного центра и всего Сибирского отделения РАН начались в первой половине 90-х годов, когда группа специалистов нескольких крупных организаций Академгородка столкнулась с потребностью в оперативных службах обмена сообщениями и файлами, в удаленном доступе к вычислительным ресурсам – всем тем, что составляет основу современных информационно-вычислительных сетей (ИВС).

В процессе изучения доступных решений было обнаружено, что существует достаточно серьезная инфраструктура, базирующаяся на кабельных линиях двух типов – традиционные телефонные линии связи (ТФОП) и широкополосные кабели с малым затуханием (ЗКП), построена разветвленная кабельная канализация. Эта инфраструктура имела радиальную топологию, обеспечивая подключение большинства институтов Академгородка к центру, расположенному в здании ВЦ по адресу: Лаврентьева, 6, по линиям связи обоих типов.

Это создавало идеальные условия для построения прототипа ИВС, объединяющей ИЯФ, ИАиЭ, ВЦ, ИСИ и НГУ. В качестве оконечных узлов были применены компьютеры класса AT-286, работающие под управлением MS-DOS, сетевой операционной системы KA9Q, сетевые серверы класса AT-386 под управлением ОС «Interactive UNIX». Узлы институтов были, как правило, объединены в локальную сеть на основе технологии Ethernet, хотя в ряде организаций встречались и более экзотические технологии (например, ARCNet в НГУ). В качестве сетевого коммутирующего оборудования использовались специализированные КАМАК-контроллеры на базе разработанной в ИЯФ микроЭВМ «Одренок» и интерфейсных КАМАК-модулей, обеспечивающих подключение к стандартному последовательному порту персонального компью-

тера на скоростях до 38 Кбит/сек и выше, в зависимости от расстояния. Уже в начале 1993 года работал прототип ИВС, объединявший локальные сети ИЯФ, ИСИ, ИАиЭ и одиночные компьютеры ВЦ и НГУ. Принятая схема, в которой объединялись полноценные локальные сети организаций, легко могла быть модернизирована путем замены маршрутизирующих узлов на более современные и использующие другие линейные технологии.

Уже на этих ранних этапах в качестве перспективного направления развития системы предусматривалась архитектура, не ограниченная ведомственными рамками (в составе команды активно работали представители НГУ), ориентированная в первую очередь не на применение активно разрабатываемых в тот период практически во всех организациях собственных технических, программных и архитектурных решений, а на повсеместное использование архитектуры TCP/IP, которая была относительно мало известна в России в это время. Открытость системы, базирование ее на существующей кабельной инфраструктуре ВЦКП делали относительно несложным расширение системы и включение в ее состав как дополнительных вычислительных ресурсов локальных сетей уже подключенных организаций, так и подключение новых организаций, имеющих ранее подключения к ВЦКП.

Таким образом, в качестве важнейших предпосылок успеха этого «самодеятельного» проекта можно назвать следующие обстоятельства:

- наличие группы энтузиастов сетевых технологий, хорошо знакомых с современными сетевыми архитектурами, в первую очередь с архитектурой TCP/IP;
- наличие практически готовой медной инфраструктуры;
- открытая заинтересованность ряда институтов и организаций, готовых к сотрудничеству и к выделению определенных ресурсов.

В начале 1993 года реализуемый проект привлек дополнительное внимание со стороны активистов Международного научного фонда (С. Мушер), которыми было предложено попытаться заручиться поддержкой МНФ. Предварительная версия проекта была составлена к концу 1993 года и позже представлена в экспертный совет МНФ, где и была одобрена в начале 1994 года.

Реальное финансирование проекта со стороны МНФ началось позже. В рамках проекта было предусмотрено оснащение сети современным телекоммуникационным оборудованием (маршрутизаторы Cisco Systems, модемы RAD Communications, рабочие станции Sun Microsystems и др.). Это позволило подключить к сети «Интернет Академгородка» большинство окрестных институтов на новом технологическом уровне – на скоростях 2 Мбит/сек на широкополосных кабелях и на меньших скоростях в случае ТФОП (ЦСБС, школа № 130, ЦКБ). Успех начальных этапов реализации проекта «Интернет Академгородка» обеспечил дальнейшую поддержку со стороны МНФ.

Приблизительно год спустя, в конце 1995 года, удалось построить и запустить в эксплуатацию один из важнейших телекоммуникационных ресурсов сети – спутниковый канал связи Новосибирск – Гамбург. Кооперативный характер всего проекта, неоднократно подчеркивавшийся во многих его документах, дополнительно проявился в реализованной схеме управления и использования этого канала: оборудование было предоставлено в рамках проекта INTAS, эксплуатация оборудования спутникового терминала обеспечивалась подразделениями ИЯФ, оплата спутникового канала обеспечивалась грантами РФФИ, тогда как техническая эксплуатация инфраструктуры сети ННЦ была обеспечена силами ИВТ.

По мере роста потребностей абонентов сети ННЦ было выявлено несколько «узких» мест. В первую очередь таковыми являлись недостаточные для современных широкополосных приложений емкости локальных подключений институтов (2–4 Мбит/сек) и хронический дефицит емкости внешних каналов.

Так, за период с 1995 по 2001 год емкость только спутникового подключения выросла более чем в 16 раз (с 64 до 1280 Кбит/сек).

Примерно в этот же период начались активные процессы интеграции сетей региональных научных центров СО РАН, которые до тех пор строились и развивались достаточно автономно и самобытно. Для иллюстрации этого обстоятельства достаточно сказать, что первые оптоволоконные линии связи на базе технологии FDDI были использованы в Иркутске, тогда как практически везде использовались в лучшем случае подключения 2 Мбит/сек. Развитие сетей научных центров происходило практически параллельно с развитием сетей вузов и тяготело к федеральным проектам RBNNet или RUNNet.

На основе опыта построения и результатов развития большой корпоративной сети ННЦ было принято решение объединения существующих сетей региональных научных центров СО РАН в единую корпоративную сеть передачи данных (СПД СО РАН). Была сформулирована концепция построения СПД в рамках СО РАН, принципы построения опорной сети, ряд рабочих документов, регламентирующих многие стороны использования сети.

В 1998 году возник проект создания альтернативной независимой сети суперкомпьютерных вычислений в пределах кампуса Академгородка, объединяющей потребителей услуг центра суперкомпьютерных вычислений. Однако в дальнейшем было признано нецелесообразным выделение такого обособленного сообщества и было принято решение приступить к широкой модернизации всей инфраструктуры.

В начале 2000-х была проведена значительная модернизация сети. В настоящее время сеть передачи данных СО РАН объединяет научные институты и организации СО РАН, СО РАМН, СО РАСХН, ГНЦ вирусологии и биотехнологии «Вектор», а также ряд других научных, учебных, медицинских организаций, учреждений культуры и социальной сферы. СПД СО РАН является одним из важнейших факторов, определяющих эффективность работы всего Отделения. Предоставляя услуги более 150 организациям-абонентам, обслуживая более 40000 пользователей только в ННЦ и порядка 30000 в региональных научных центрах, объединяющая около 15000 компьютеров, СПД СО РАН стала крупнейшей корпоративной научно-образовательной сетью России.

О ПЕРСПЕКТИВАХ РАЗВИТИЯ ИТ И ИНТЕРНЕТА В УЗБЕКИСТАНЕ

И.М. Шамсиева

Национальный университет Узбекистана

Ташкент, Узбекистан

i_shamsieva@rambler.ru

Сегодня, как и во всем мире, на ведущие позиции в Узбекистане выходят компьютерные и информационные технологии, развитие и модернизация сетей телекоммуникаций и передачи данных, доступа к Интернету. Прорыв в области развития ИТ очень важен для республики и поэтому является в настоящее время одним из приоритетных направлений государственной политики. Перспективы развития информационных технологий в стране определены постановлением президента Республики Узбекистан от 8 июля 2005 года «О дополнительных мерах по дальнейшему развитию информационно-коммуникационных технологий». В частности, документом утверждены целевые ориентиры развития сетей телекоммуникаций, передачи данных и применения ИКТ до 2010 года, программа применения ИКТ в деятельности органов государст-

венного управления и государственной власти на местах до 2010 года, программа формирования и развития Национальной информационно-поисковой системы.

В Узбекистане в целом ИТ и Интернет все глубже проникают в общество. Повышается интенсивность использования и рост опыта работы с новыми информационными технологиями. Если сейчас в республике Интернетом пользуются более 855 тысяч человек (около 33 пользователей на 1000 жителей), то уже к 2010 году число пользователей глобальной сети составит 3,4 млн. Ведется работа по увеличению скорости доступа сети к международным информационным сетям. Предполагается, что через четыре года показатель возрастет до 512 Мбит/сек. (на сегодняшний день – 143,104 Мбс).

До недавнего времени в национальном сегменте Интернета не было единой информационной сети, которая бы систематизировала информационные ресурсы образовательной и молодежной направленности и способствовала развитию просвещения, духовно-нравственного воспитания молодежи. Теперь в Узнете будет сформирована общественная образовательная сеть «ZiyoNET» (Ziyo (узб.) – просвещение) с единой информационно-ресурсной площадкой, объединяющей материалы, предназначенные для учебно-образовательных, научных, молодежных, детских учреждений и организаций. Уже запущена пилотная версия сайта образовательной сети ZiyoNET по адресу www.ziynet.uz.

Безусловно, в республике создаются благоприятные условия для более полного удовлетворения все возрастающих потребностей граждан в информации, но еще многое предстоит сделать. Недостаточным остается уровень информатизации библиотек и архивов, высших учебных заведений, использования ИТ для развития экономики, основанной на знаниях, и для улучшения инновационной деятельности. Все это имеет важное значение в национальном развитии страны и в создании благоприятной среды для интеграции в мировое информационное сообщество.

КОМПЛЕКС БЦВМ «АРГОН» ЗА СОРОК ЛЕТ – ДИНАМИКА РАЗВИТИЯ

В.И. Штейнберг

ФГУП «НИИ «Аргон»

Москва, РФ

argon@argon.ru

История БЦВМ комплекса «Аргон» во многих отношениях отражает развитие отечественной бортовой вычислительной техники в целом. НИИ «Аргон» в СССР был основным научным и конструкторским центром в этой области, выполнявшим в рамках Межведомственного координационного совета функции базового предприятия, ответственного за подготовку государственных программ по унифицированным межвидовым семействам БЦВМ и заданий смежным отраслям промышленности. Свыше сорока завершенных разработок института были внедрены в серийное производство и приняты в штатную эксплуатацию в сотнях АСУ ракетно-космических, авиационных и мобильных объектов.

Работы по созданию комплекса БЦВМ «Аргон» условно можно разделить на три основных этапа.

На первом этапе (1964 – середина 70-х гг.) разработаны 11 моделей БЦВМ. Они имели специальные наборы команд и строились на гибридных и первых сериях монолитных ИС. Основные конструктивные элементы: многослойные платы попарного прессования и послойного наращивания, пакет книжной конструкции.

Для второго этапа (середина 70-х – начало 90-х гг.) характерен переход к унифицированным архитектурам, ИС со средним и высоким уровнем интеграции и новой конструктивно-технологической базе. Были созданы два семейства БЦВМ: высокопроизводительных программно-совместимых с ЕС ЭВМ (базовые модели А-30, А-40, А-50) и управляющих с оригинальной архитектурой «ПОИСК». В качестве конструктивных решений в них использованы платы с металлизацией сквозных отверстий и модульное иерархическое построение аппаратуры.

На основе моделей А-30, А-40, А-50 создан целый ряд одно- и многомашинных бортовых вычислительных комплексов для авиационных, мобильных и стационарных объектов.

Для третьего этапа (90-е гг. с переходом в 2000-е гг.) разработаны и освоены технология высоконадежной малогабаритной аппаратуры и технология модульных открытых вычислительных средств.

Первая технология базируется на использовании сертифицированной высоконадежной элементной базы и базовой несущей конструкции пакетной сборки с оригинальными платами рельефного монтажа, имеющими ряд преимуществ перед традиционными многослойными платами. По данной технологии изготовлено более 150 штатных приборов для Международной космической станции и спутников связи пятого поколения.

Основу технологии модульных открытых вычислительных средств составляют конструктивы Евромеханики, магистраль VME. По данной технологии реализован ряд модификаций многофункционального бортового вычислительного комплекса с процессором класса Pentium и комплекс встраиваемых вычислительных модулей для связанных терминалов, реализованных на платформе микропроцессора ЦОС 1867 ВЦ1Ф (аналог TMS320 C30).

Приводятся основные технические характеристики моделей и их динамика с учетом развития компонентной базы и требований систем и объектов.

ОПЫТ ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ИДЕНТИФИКАЦИИ РЕЖИМОВ ТЕЧЕНИЯ ДВУХФАЗНЫХ ПОТОКОВ

В.Ф. Шуршев

Астраханский государственный технический университет
Астрахань, РФ
v.shurshev@astu.org

Представлен комплекс алгоритмического и программного обеспечения для идентификации режимов течения двухфазного потока, который является наиболее сложным из всех существующих потоков из-за постоянно меняющейся границы раздела фаз и сжимаемости паровой фазы. Для идентификации и согласования результатов идентификации по различным фазовым диаграммам, экспериментальным данным, знаний экспертов в программе используется разработанный метод дельта-окрестности, осуществляющий идентификацию режимов течения с использованием фазовых диаграмм режимов течения, для учета и классификации имеющихся экспериментальных данных заложен разработанный модифицированный метод К-ближайших соседей, схема согласования основана на правиле Шортлиффа. Данный комплекс используется при расчете и проектировании теплообменного оборудования. Программные продукты внедрены на ряде предприятий Южного федерального округа.

КОНЦЕПЦИЯ СОЗДАНИЯ ОБРАЗОВАТЕЛЬНОЙ СЕТИ МИНИСТЕРСТВА ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ТАТАРСТАН

А.У. Ярмухаметов

ОАО «International Computers Limited – Казанское производственное объединение вычислительных систем»
Казань, РФ

В докладе излагаются основные положения концепции создания образовательной сети Татарстана. В основу положены следующие принципы:

- 1) минимизация стоимости владения сетью как условие выживания образовательной сети в условиях остаточного бюджетного финансирования;
- 2) Интранет для школьников, Интернет для преподавателей;
- 3) принципиальное отсутствие коммерческой деятельности;
- 4) повышение надежности сети за счет дублирования ресурсного центра сети;
- 5) интравертность ресурсного центра, экстравертность портала.

Для начала несколько цифр:

- В Татарстане на один школьный компьютер приходится в среднем 24 обучаемых.
- К образовательной сети МОиН РТ подключено 1806 образовательных учреждений.
- По «техническим причинам» к сети не подключены 82 школы, что составляет менее 0,5 % от общего их числа.

Данный доклад дискуссионный. Я излагаю здесь свою точку зрения на проблему программы «подключения школ к Интернет». Правда, эта точка зрения получила конкретное воплощение при построении школьного сегмента научно-образовательной сети Татарстана. Научно-образовательная сеть имеет три сегмента: научно-исследовательский, использующий суперкомпьютеры (академические институты), вузовский, в котором работают преподаватели и студенты, и школьный сегмент. Мы будем рассматривать последний.

Какова **сверхзадача** школьного образования? – Сделать молодое поколение конкурентоспособным на рынке труда, а следовательно, богатым и здоровым.

Какова **задача** информатизации школ? – Дать все богатства, накопленные человечеством, как говорил вождь и учитель? Нет, задача состоит в том, чтобы научить, как эти знания получить и как их использовать. Другими словами, «дать удочку и обучить, как ею пользоваться».

Какова **цель** создания инфокоммуникационной образовательной сети Министерства образования? – Сделать упомянутую «удочку» качественной, доступной каждому ученику и педагогу.

Какой должна быть сеть? – Максималисты говорят: «Современные мощные компьютеры и широкополосный скоростной доступ в Интернет». Как вы думаете, что эффективней для обучения – работать один час в неделю на мощном компьютере или три часа в день на среднем при условии, что на них установлены одинаковые программы? Мощные компьютеры нужны для решения задач автопроектирования, решения криптографических задач и т.д. – в производстве и науке. Широкополосный доступ в Интернет обычно используется тинейджерами для загрузки музыкальных MP3-файлов, видеофайлов, трехмерных игр-стрелялок и т.п. Нужно ли это для получения навыков работы с компьютером? Наверно нужно, если не считать деньги.

Поэтому первый принцип создания школьной сети – «Принцип разумной достаточности». Или по другому: «Как при ограниченных средствах получить наивысший эффект».

Второй принцип: «Считай не стоимость закупки, а совокупную стоимость владения». Поясню простым примером. Практически каждый из нас имеет сотовый телефон. Стоимость плохенького телефона около ста долларов. При средней загрузке требуется оплатить 20 единиц в месяц, или 240 долларов в год. Совокупная стоимость владения телефоном в год составит 340 долларов. За пять лет (обычный срок для электроники) – 1300 долларов, что во много раз превышает стоимость самого аппарата.

Та же история и с Интернетом. Объясняю «на пальцах». Модем стоит менее 100 долларов, а оплата трафика оператору и контента провайдеру – тысячи долларов. Не секрет, что многие школы формально подключены к сети Интернет, т.к. не имеют возможности ею пользоваться по причине хронического безденежья. Мне известны случаи, когда провайдер бесплатно раздавал модемы школам и подключал к своей сети. Чем это заканчивалось? – Два – три месяца школы с энтузиазмом рылись в Интернет, после чего деньги заканчивались и проект благополучно умирал. Однако в официальных отчетах эти школы победно числились как «подключенные к Интернет».

Поэтому главная цель создания школьной сети – максимально сократить совокупную стоимость владения за счет экономии на доступ в Интернет.

Как это делается? Создается мощный ресурсный центр с большой базой знаний и к нему подключаются все школы. Ресурсный центр мы подключили к Интернет через SENet Tatarstan (Академсеть), что существенно снижает стоимость получения информации школами. Таким образом, не полторы тысячи школ самостоятельно «покупают» информацию в Интернет, а ресурсный центр один раз берет ее в свою базу знаний и бесплатно раздает всем школам. Налицо существенная экономия. Если некоторая информация стоит всего 1 доллар, то для 1500 школ она обойдется в 1500 долларов, а это уже 2 новых компьютера.

В базе знаний ресурсного центра размещаются учебные пособия, методические материалы, библиотеки и другая необходимая для учебного процесса информация. Более того, если преподаватель-предметник разработал новую эффективную методику, он может (и даже должен) разместить ее на ресурсном центре для всеобщего апробирования и внедрения.

Такая конфигурация школьной сети позволяет не только предоставить школам отобранные в море информации полезные данные, но и оградить от несанкционированного доступа к вредной для школьников информации, как это возможно при непосредственном подключении к Интернет.

Автоматически решаются некоторые вопросы информационной безопасности – в одной точке – на ресурсном центре фильтруются всем известные вирусы, компьютерные черви, трояны и другие вредоносные программы и спам.

Доступ преподавателей к информации, отсутствующей в базе знаний ресурсного центра (т.е. прямо в WWW), должен осуществляться по их заявке и фиксироваться.

Таким образом, школьная сеть – это ИНТРАНЕТ, закрытая сеть без непосредственного доступа школьников ко Всемирной Паутине.

Теперь о реализации концепции.

Концепция предусматривала три этапа:

1. Создание ресурсного центра и наполнение его контентом.
2. Подключение к нему районных компьютерных центров, обучение персонала центров. Отработка взаимодействия с центром. Пилотное подключение наиболее продвинутых школ.
3. Подключение всех школ.

Как это у нас бывает практически всегда, все было сделано наоборот.

В 2004 г. в соответствии с федеральной программой «Дети России» было выделено около 40 млн. руб. «на подключение школ к Интернету». Обращаю внимание на терминологическую тонкость: «подключение ШКОЛ», а не «образовательных учреждений». Дело в том, что ПТУ, колледжи, спецшколы для детей с ограниченными возможностями и т.п. официально «не школы». То есть «школы» есть подмножество множества «образовательные учреждения». Напомню, что в апреле разгорелся скандал, когда около 80 000 учителей не получили обещанных надбавок, потому что они работают не в «общеобразовательных школах», а, например, в кадетских корпусах. Таким образом, мы не имели права подключать к сети районные компьютерные центры, районные отделы образования и т.п. учреждения, которые и надо было подключать в первую очередь.

Для создания ресурсного центра удалось получить финансирование из резервного фонда Президента РТ в объеме 8 млн. руб. В 2004 г. создана первая очередь ресурсного центра в Казани и подключено к нему 1300 школ республики.

На втором этапе в 2005 г. финансирование осуществлялось уже из бюджета Татарстана. Путем бюрократической эквилибристики удалось изменить формулировку названия проекта, и вместо школ к сети подключались уже «образовательные учреждения», т.е. удалось осуществить концепцию в полном объеме. Было подключено 500 объектов, в том числе районные компьютерные центры, интернаты, специализированные школы, некоторые ПТУ, техникумы. В городах и районных центрах подключение выполнялось по технологии ADSL, в деревнях – по коммутируемой связи. Соотношение примерно 1:1.

Существенно изменилась структура затрат. Подключение обошлось примерно в 9 млн. руб., а обслуживание ранее сформированного сегмента потребовало около 13 млн. руб. Несмотря на жесткую экономию, принятые структурные меры по минимизации затрат, по расчетам затраты на обслуживание сети составят около 20 млн. руб. в год, что вполне сопоставимо со стоимостью создания сети. Причем это без учета стоимости подготовки персонала на местах. Обучение персонала тоже является проблемой. Причем в самом неожиданном аспекте. Оказывается, чиновники от образования совершенно не заинтересованы в качественном (например, с получением сертификата Майкрософт или Интел по окончании курсов) обучении.

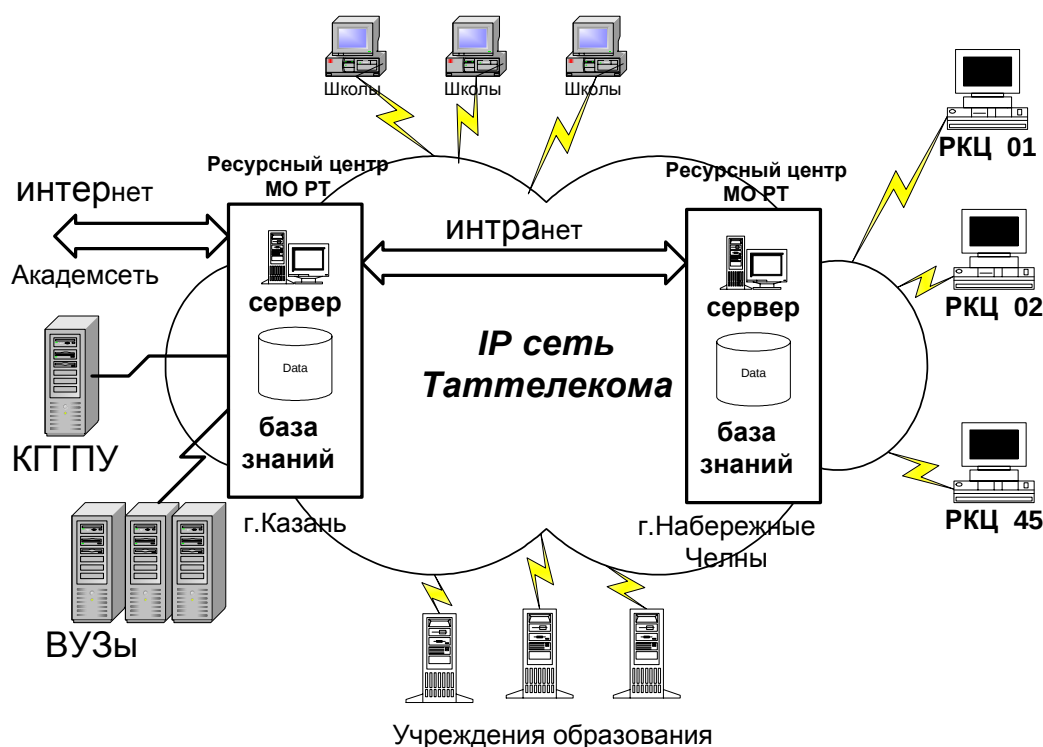
Мы рассматриваем созданную школьную сеть как фундамент для развития множества не только полезных, но и необходимых технологий. Одной из них является электронный документооборот министерства.

После преобразования Министерства образования в Министерство образования и науки к процессу наполнения образовательной сети подключается ряд вузов Казани, в том числе такие известные, как двухсотлетний Казанский госуниверситет, Казанский авиационный институт, академические институты Казанского филиала АН РФ и пр.

Мы видим ряд проблем, с которыми предстоит столкнуться при эксплуатации сети. Одна из них – весьма вероятная тенденция коммерциализации образовательной сети. Опасна «инициатива» снизу в виде попытки приторговывать полученным «халявным» Интернетом в форме создания Интернет-кафе, молодежных компьютерных клубов и т.д. К сожалению, менталитет таков, что многие просто не понимают, что это тривиальное воровство. Бизнес бизнесом, но он должен быть минимально цивилизованным. Необходим жесткий контроль на местах, и это решение проблемы.

Гораздо опаснее «инициатива» сверху. Подавляющее большинство чиновников просто не имеют представления, что для операторской и провайдерской деятельности необходимо получать лицензии, установить кассовые аппараты для сбора наличных денег, платить налоги, иметь проблемы с пожарниками, санинспекторами и прочими «контролирующими органами». Взвалить все эти проблемы на затюканного учителя информатики просто нереально, но ведь могут в приказном порядке!

Наличивается некая путаница в понятиях «портал» и «ресурсный центр». Я представляю ресурсный центр (РЦ) как образование интравертное, т.е. обращенное внутрь закрытой корпоративной школьной сети. Все члены этой сети могут пользоваться ресурсам РЦ в рамках предоставленных прав по льготным внутренним расценкам. Портал же – образование экстравертное, предназначенное для предоставления информации сторонним лицам. Более того, в рамках программы «Электронное правительство» регламентирован перечень сведений, которые любое министерство обязано выложить на своем портале. Портал также должен обеспечить абонентам РЦ доступ к ресурсам извне, например с домашнего компьютера. Базовые аппаратные средства (железо) при этом могут быть одни и те же как для портала, так и для ресурсного центра.



INFORMATION TECHNOLOGY POLICY FOR THE FORMER SOVIET NATIONS

A. Fitzpatrick

Federation of American Scientists

Falls Church, USA

achfitzpatrick@yahoo.com

No other technology is having as big an influence on our lives today as the vast and overwhelming array of technological artifacts, computers, networks, and means of communication that comprise information technology, or IT. From the Washington, DC political arena to governments worldwide, policy makers are struggling to keep up with, anticipate, and formulate the ways in which governments formulate public policy on science and technology.

Among the most challenging problems, the IT industry is growing so fast that governments are having an enormously difficult time projecting technological trajectories and trends, and have trouble establishing policies that conform to the pattern of enterprises in this industry.

How are these factors affecting IT growth, development, and investment in the former Soviet Union, where an educated workforce and new generation of talented individuals are becoming part of the global work force? How can Russia and the former Soviet nations best adapt to these trends and influence them in ways that benefit their own countries? Can lessons be learned from India's experience with the IT industry?

This panel will address these questions and a number of other related concerns in the realm of IT that are of particular interest to Russia and the former Soviet Union.

Confirmed Panelists are listed in the Co-author section below. Two other participants, one from private industry and one from India, have been invited but have not confirmed their participation on this panel.

COMPARATIVE COMPUTING EDUCATION: EAST AND WEST

J. Impagliazzo

Hofstra University

Hempstead, USA

john.impagliazzo@hofstra.edu

The developments of computing education in the East and West have exhibited differences in content and approach as they evolved over past decades. For example, the recent recommendations set forth in the «Applied Mathematics and Informatics» curriculum grew from former curricula structures; it promises to provide new directions toward the manner in which university computing education will take place within Russia and beyond. As another example, «Curriculum'68» that ACM published in 1968 had provided the framework for university computing education in the United States and beyond and has led to a series of other curricula recommendations over the years such as the «Computing Curricula 2001» report. Both approaches have similarities and differences, particularly in their philosophical purposes.

This abstract proposes a panel discussion on the perspectives of computing education, particularly as it applies to undergraduate university education. The panel will consist of experts in the field from the Former Soviet Union, from Western Europe, and from the United States. The conference International Program Committee will select at least two experts to represent «Eastern» advancements and perspectives in the area. The panel chair, who has helped author two of the current curricula reports from ACM and the IEEE Computer Society, will be one of the panelists. He will select two other individuals to characterize «Western» developments and viewpoints on the subject.

The panel should be approximately ninety minutes in length and each panelist will make an eight-minute presentation on his or her focus area. Additionally, panelists will provide to the audience written summaries of their comments and possibly provide related material. As full audience participation is critical to the success of this panel proposal, questions and comments will occupy the remaining time (approximately thirty-five minutes). The audience should anticipate a frank and vibrant discussion on this important topic.

SOME RESULTS OF THE COOPERATION BETWEEN SOVIETS AND CUBANS ORGANIZATION FROM 1973 TO 1989

Tomas Lopez Jimenez

University of Information Sciences

C. Habana, Cuba

tomas@uci.cu

In the 1972 year Cuba became member of the COMECON. The Soviet Union organized the cooperation between the COMECON countries in the Computing Sciences and Industries fields by means of the International Commission for the Computing Techniques (MPK po BT).

From the 1973 year the Cuban ICID developed some joint projects with the Soviet INEUM and other Soviet institutions. In the present paper are described the main developed projects that were part of the SM-MCE (Small Systems of Electronics Computing Machines – Minicomputers Systems). Some minicomputers (SM -3 and SM-4), video displays and keyboards. Also are included some software products, specially a Cuban developed COBOL compiler for the Soviets SM – 3 and SM -4 that were supplied to Finlandia.

THE IMPORTANCE OF ORAL HISTORY IN RESEARCHING THE DEVELOPMENT OF COMPUTERS IN THE SOVIET UNION

F. Nebeker

IEEE History Center, Rutgers University

New Brunswick NJ, USA

f.nebeker@ieee.org

In European universities of the 19th century, the documentary tradition of historical scholarship became established. It required that historians cite all sources and use only original documents as sources. In the last half century, the historical community has recognized that for many types of history a different type of documentation is indispensable: the personal testimony of participants. By recording the testimony, producing a transcript, and making the transcript available to scholars, practitioners of this new type of history, called oral history, turn eyewitness accounts into citable documents.

In the case of the development of computers in the Former Soviet Union (FSU), oral history has only begun to take hold as a viable method of documentation. Such an activity would provide an essential supplement to the record provided by paper documents, particularly since libraries can preserve this activity in electronic form. For some computer projects, paper documents have not survived or are not presently accessible – or cannot be documented in paper such as with interactive software. Even in cases where there is abundant documentation, it is usually the case the important aspects of the work, notably the thoughts, feelings, and motivations of the authors of those documents, have not been captured in the written documents.

In this work, the authors will discuss, on the basis of extensive experience with oral-history projects, how oral history might be used in building a broader and richer picture of the development of computing in the FSU.

INRIA – AKADEMGORODOK, A HISTORY OF THE SOVIET-FRENCH COOPERATION IN THE FIELD OF COMPUTER SCIENCE

K. Tatarchenko

Paris IV, la Sorbonne

Paris, France

ktatarchenko@yandex.ru

Thanks to the archives at the Institut National de Recherche en Informatique et Automatique (INRIA), I was able to lay the groundwork for the work of my research, the fruits of which were developed in a Master's Thesis defended at the Sorbonne in July 2005: «L'INRIA – Akademgorodok, une histoire de coopération franco-sovietique dans le domaine d'informatique». The object of my research belongs to multiple institutional intersections and seeks to provide a social and intellectual history of computer science, a field in perpetual evolution. My work seeks to account for the political and institutional stakes that accompanied the development of science as a discipline and activity.

In 1958, as an alternative to the «closed cities» working for the military, N. Khrushchev supported the construction of Akademgorodok, a sort of scientific community based upon Francis Bacon's «New Atlantis». The Computing Center of Akademgorodok, although displaced in the middle of the Siberian forests, nevertheless attempted to unite the best Soviet specialists, (such as Lyapunov, Ershov, Yanenko and many others) and to earn an international reputation as quickly as possible. The community, composed of both scientifically motivated connections and personal friendships, brought together two eminent mathematicians: G. I. Marchuk, director of the Computing Center, and J-L Lions, director of LABORIA at the heart of the INRIA. The reciprocal interest in encouraging their teams to work together in digital technologies and applied mathematics encountered favorable circumstances.

In March 1966, France left the leadership of the United Nations and, three months later, General de Gaulle undertook a triumphant official visit to the Soviet Union. The bilateral agreements that followed the arrival of General de Gaulle provided the administrative and financial framework for scientific cooperation in the domain of computer science, which developed according to the structures of these agreements throughout the following decades up until 1993.

This history of cooperation in the domaine of computing occuring at the intersection of personal ambitions, academic institutions and political pressures provides the context of rethinking the place of science in the History of the Cold War.

ECOMICS OF INFORMATION TECHNOLOGY: OPERATIONS, INVESTMENTS, AND STRATEGY

R. Teller

Booz Allen Hamilton

Falls Church, VA, USA

rtteller@gmail.com

The economics of information technology is an enabler for companies to achieve rapid growth, better utilization of informatics, and can achieve a company's business strategy. Without an understanding of the informatics economics, companies stagnate and waste computer money and resources, and fall behind their competition.

PRINCIPAL ASPECTS OF THE MINIMAL-TIME SYSTEM DESIGN METHODOLOGY

A. Zemliak

Puebla Autonomous University

Puebla, Mexico

azemliak@fcfm.buap.mx

One of the main problems of a large system design is the excessive computer time that is necessary to achieve the optimal point of the design process. This problem has a great significance at least for the VLSI electronic circuit design. A system design methodology includes two main parts as a rule: the block of analysis of mathematical model of the system and optimization procedure that achieves the optimal point of the design process objective function. This is a traditional design approach for the system design. We call it as a Traditional Design Strategy (TDS). In the limits of this conception it is possible to change optimization strategy and use the different analysis methods. There are some powerful methods that reduce the necessary time for the circuit analysis by means of the special sparse matrix techniques [1] or by the partitioning of a circuit matrix [2]–[3].

Another formulation of the circuit optimization problem was developed in heuristic level some decades ago [4]. This idea was based on the Kirchhoff laws ignoring for all the circuit or for the circuit part. The special cost function is minimized instead of the circuit equation solving. This idea was developed in practical aspect for the microwave circuit optimization [5] and for the synthesis of high-performance analog circuits [6] in extremely case, when the total system model was eliminated. The last idea that excludes completely the Kirchhoff laws can be named as the Modified Traditional Design Strategy (MTDS).

The generalized theory for the system design on the basis of the control theory formulation was elaborated in some previous works [7]–[8]. This approach serves for the time-optimal design algorithm definition. The General Design Strategy (GDS) can be defined and includes a set of the different design strategies. The TDS and the MTDS are two extreme cases of the GDS only. The GDS combines all the possibilities for different design strategies. This approach serves for the minimal-time design algorithm definition. On the bounds of this conception, the traditional design strategy is only a one representative of the enormous set of different design strategies. As shown in [8] the potential computer time gain that can be obtained by the new design problem formulation increases when the size and complexity of the system increase but it is realized only in case when we have the algorithm for the optimal trajectories systematical construction. We can define the formulation of the intrinsic properties and special restrictions of the optimal design trajectory as one of the first problems that needs to be solved for the optimal algorithm construction.

Problem formulation

The design process for any analog system design can be defined as the problem of the cost function $C(X)$ minimization ($X \in R^N$) with the system of constraints. It is supposed that the minimum of the cost function $C(X)$ achieves all design objects and the system of constraints is the mathematical model of the electronic circuit. It is supposed also that the circuit model can be described as the system of nonlinear equations:

$$g_j(X) = 0 \quad j = 1, 2, \dots, M \quad (1)$$

The vector X can be separated in two parts: $X = (X', X'')$. The vector $X' \in R^K$ can be named as the vector of independent variables where K is the number of independent variables and the vector $X'' \in R^M$, is the vector of dependent variables, where $N = K + M$.

The optimization process for the cost function $C(X)$ minimization for two-step procedure can be defined in general case as following vector equation:

$$X^{s+1} = X^s + t_s \cdot H^s \quad (2)$$

with constraints (1), where s is the iterations number, t_s is the iteration parameter, $t_s \hat{\mathbf{R}}^1$, H is the direction of the cost function $C(X)$ decreasing. The vector H is the function of $C(X)$.

The specific character of the design process for the electronic systems consists in fact that it is not necessary to fulfill the conditions (1) for all steps of the optimization process. It is quite enough to fulfill these conditions for the final point of the design process.

The problem (1), (2) can be redefined in the form when there is no difference between independent and dependent variables. All components of the vector X can be defined as independent. This is the main idea for the penalty function method application. In this case the vector function H is the function of the cost function $C(X)$ and the additional penalty function $\varphi(X)$: $H^s = f(C(X^s), j(X^s))$. The penalty function structure includes all equations of the system (1) and can be defined for example as:

$$j(X^s) = \frac{1}{e} \sum_{i=1}^M g_i^2(X^s). \quad (3)$$

In this case we define the design problem as the unconstrained optimization (2) in the space R^N without any additional system but for the other type of the objective function $F(X)$. This function can be defined for example as an additive function: $F(X) = C(X) + j(X)$. In this case we reach the minimum of the initial cost function $C(X)$ and comply with the system (1) in the final point of the optimization process (in the minimal point of the function $F(X)$). This method is named as MTDS. This method produces another design strategy and another trajectory line in the space R^N .

On the other hand it is possible to generalize the idea of additional penalty function application if to make up the penalty function as one part of the system (1) only and the other part of this system is defined as the constraints. The penalty function includes first Z items only, $j(X^s) = \frac{1}{e} \sum_{i=1}^Z g_i^2(X^s)$ where $Z \in [0, M]$ and $M - Z$ equations make up one modification of the system (1):

$$g_j(X) = 0 \quad j = Z + 1, Z + 2, \dots, M. \quad (4)$$

It is clear that each new value of the parameter Z produces a new design strategy and a new trajectory line. This idea can be generalized more in case when the penalty function $j(X)$ includes Z arbitrary equations from the system (1). The total number of different design strategies is equal to 2^M if $Z \in [0, M]$. All these strategies compose a structural basis for GDS. These strategies exist inside the same optimization procedure. The optimization procedure is realized in the space R^{K+Z} . It is clear that these different strategies have various computer times because they have a different operations number. It is appropriate in this case to define the problem of an optimal design strategy search that has the minimal computer time. Here and further the optimality is defined as the computer time minimization.

However the most general approach can be constructed on the basis of the design problem formulation as the problem of optimal control. It is possible to define a design strategy by equations (2), (4) with a variable value of

the parameter Z during the all optimization process. It means that we can change the number of independent variables and the number of the terms of the penalty function in each point of the optimization procedure. It is convenient to introduce in consideration a vector of the special control functions $U = (u_1, u_2, \dots, u_m)$ for this aim, where $u_j \in \Omega$; $\Omega = \{0; 1\}$. The sense of the control function u_j is next: equation number j is presented in the system (4) and the term $g_j^2(X)$ is removed from the right part of the formula (3) when $u_j = 0$, and on the contrary, the equation number j is removed from the system (4) and is presented in the right part of the formula (3) when $u_j = 1$. The next formulas are defined for the model of the system and for the penalty function:

$$(1 - u_j)g_j(X) = 0 \quad j = 1, 2, \dots, M; \quad (5)$$

$$\varphi(X^s) = \frac{1}{\varepsilon} \sum_{j=1}^M u_j \cdot g_j^2(X^s). \quad (6)$$

All the control variables u_j are the functions of the current point of the optimization process. The vector of the directional movement H is the function of the vectors X and U in this case: $H = f(X, U)$. The total number of the different design strategies which are produced inside the same optimization procedure is practically infinite. Among all these strategies exist one or few optimal strategies that achieve the design objects for the minimum computer time. So, the problem of the optimal design strategy search is formulated now as a typical minimal-time problem of the control theory. The main problem of this definition is unknown dependency of all control functions.

Minimal-Time System Design Problem Formulation

It is possible to determine the problem of any analog system design as the problem of the optimal control. The principal system of equations can be defined in continuous or discrete form and for example for continuous form consists of following two systems of equations:

$$\frac{dx_i}{dt} = f_i(X, U), \quad i = 0, 1, \dots, N; \quad (7)$$

$$(1 - u_j)g_j(X) = 0, \quad j = 1, 2, \dots, M; \quad (8)$$

where $N=K+M$, x_0 is the additional variable, U is the vector of control variables, $U = (u_1, u_2, \dots, u_M)$, $u_j \in \Omega$; $\Omega = \{0; 1\}$.

The functions of the right hand part of the system (7) depend on optimization method and can be determined for example for the gradient method as:

$$f_i(X, U) = -\frac{\delta}{\delta x_i} F(X, U) \quad i = 1, 2, \dots, K; \quad (9)$$

$$f_i(X, U) = -u_{i-K} \frac{\delta}{\delta x_i} F(X, U) + \frac{(1 - u_{i-K})}{t_s} \{-x_i^s + \eta_i(X)\} \\ i = K + 1, K + 2, \dots, N; \quad (9')$$

where $F(X,U) = C(X) + \frac{1}{\varepsilon} \sum_{j=1}^M u_j g_j^2(X)$; the operator $\frac{d}{dx_i}$ hear and below means $\frac{d}{dx_i} j(X) = \frac{\eta_j(X)}{\eta_{x_i}} + \sum_{p=K+1}^{K+M} \frac{\eta_j(X)}{\eta_{x_p}} \frac{\eta_{x_p}}{\eta_{x_i}}$, x_i' is equal to $x_i(t-dt)$; $\eta_i(X)$ is the implicit function ($x_i = h_i(X)$) that is determined by the system (8).

The control variables u_j have the time dependency in general case. The equation number j is removed from (8) and the dependent variable x_{K+j} is transformed to the independent when $u_j=1$. This independent parameter is defined by the formulas (7), (9). In this case there is no difference between formulas (9) and (9'). On the other hand, the equation (7) with the right part (9') is transformed to the identity $\frac{dx_i}{dt} = \frac{dx_i}{dt}$, when $u_j=0$, because $h_i(X) - x_i' = x_i(t) - x_i(t-dt) = dx_i$. It means that at this time moment the parameter x_i is dependent one and the current value of this parameter can be obtained from the system (8) directly. This transformation of the vectors X' and X'' can be done at any time moment. The function $f_0(X,U)$ is determined as the necessary time for one step of the system (7) integration. This function depends on the concrete design strategy. The additional variable x_0 is determined as the total computer time T for the system design. In this case we determine the problem of the time-optimal system design as the classical problem of the functional minimization of the optimal control theory. To minimize the total design computer time it is necessary to find the optimal behavior of the control functions u_j during the design process. The functions $f_i(X,U)$ are piecewise continued because the control functions u_j are discontinuous.

Now the analog system design process is formulated as a dynamical controllable system. We need to find the special conditions to minimize the transition time for this dynamical system.

Examples

Some electronic circuits have been analyzed to demonstrate a new system design approach based on the optimal control theory. The circuit for three transistor cell amplifier shows in Fig. 1.

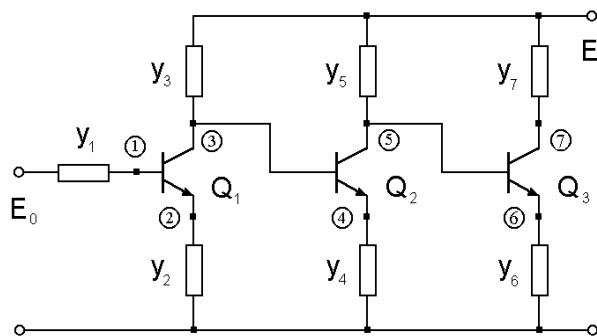


Fig. 1. Circuit topology for three-cell transistor amplifier

Table 1. Data of the three-transistor cell circuit analysis

N	Control functions vector U (u1, u2, u3, u4, u5, u6, u7)	Gradient method		DFP method	
		Iterations number	Total design time (sec)	Iterations number	Total design time (sec)
1	(0 0 0 0 0 0 0)	6379	321,09	854	64,47
2	(0 0 1 0 1 0 1)	922	54,53	764	52,29
3	(0 0 1 0 1 1 0)	1667	80,71	650	46,13
4	(0 0 1 0 1 1 1)	767	35,35	426	22,68
5	(0 0 1 1 1 0 0)	3024	159,67	940	52,71
6	(0 0 1 1 1 0 1)	823	37,73	177	7,71
7	(0 0 1 1 1 1 0)	3068	86,87	450	14,56
8	(0 0 1 1 1 1 1)	553	15,75	170	6,93
9	(0 1 1 0 1 0 1)	465	10,01	101	2,66
10	(0 1 1 0 1 1 0)	1157	31,92	111	3,85
11	(0 1 1 0 1 1 1)	501	8,82	124	2,66
12	(0 1 1 1 1 0 0)	2643	72,66	314	9,24
13	(0 1 1 1 1 0 1)	507	9,24	170	4,62
14	(0 1 1 1 1 1 0)	3070	67,27	423	12,25
15	(1 0 1 0 1 0 1)	1345	28,07	397	16,94
16	(1 0 1 0 1 1 1)	615	10,01	191	4,62
17	(1 0 1 1 1 0 1)	699	10,71	197	4,97
18	(1 0 1 1 1 1 1)	366	4,97	103	1,96
19	(1 1 1 0 1 0 1)	789	10,43	201	4,97
20	(1 1 1 0 1 1 0)	3893	61,53	1158	18,06
21	(1 1 1 0 1 1 1)	749	7,71	148	2,11
22	(1 1 1 1 1 0 0)	4325	90,72	945	19,18
23	(1 1 1 1 1 0 1)	796	8,47	133	2,31
24	(1 1 1 1 1 1 0)	2149	29,26	1104	13,44
25	(1 1 1 1 1 1 1)	2031	5,67	180	0,77

Table 2. Data of the optimal design strategie

N	Method	Optimal control functions vector U (u1, u2, u3, u4, u5, u6, u7)	Iterations number	Switch points	Total design time (sec)	Computer time gain
1	Gradient method	(1111111); (1111101)	363	350	1,127	285
2	DFP method	(1111111); (1110111)	69	66	0,322	200

The Ebers-Moll static model of the transistor has been used. The analyzed circuit has seven independent variables $y_1, y_2, y_3, y_4, y_5, y_6, y_7$ as admittance ($K=7$) and seven dependent variables $V_1, V_2, V_3, V_4, V_5, V_6, V_7$ as nodal voltages ($M=7$). The results of the analysis of the traditional approach (strategy 1) and some other strategies that have the computer time less than the traditional strategy are given in Table 1. The optimal strategies from this table (number 18 and 25 for two optimization procedures respectively) are not optimal in general and the data for the time-optimal strategies are given in Table 2. The time gain of the optimal design strategy with respect to the traditional strategy is equal to 285 for the gradient method and 200 for the DFP method. However, the potential time gain is realized when the algorithm for the optimal trajectories systematic construction is found.

Optimal Algorithm Structure Study

To obtain the optimal sequence of the switch points during the design process, we need to define a special criterion that permits to find the optimal control vector U . The problem of the minimal time strategy searching is connected with the more general problem of the stability of each trajectory. There is a well known idea to study of any dynamic process stability properties by means of the Lyapunov direct method. We have been defined the system design algorithm as the dynamic controllable process. In this case we can study the stability of each trajectory and the design process transit time properties on the basis of the Lyapunov direct method. We propose now to use a Lyapunov function of the design process for the optimal algorithm structure revelation, in particular for the optimal switch points searching. There is a freedom of the Lyapunov function choice because of a non-unique form of this function. Let us define the Lyapunov function of the design process (7)–(8) by the next expression:

$$V(X,U) = \sum_i \left(\frac{\partial F(X,U)}{\partial x_i} \right)^2, \quad (10)$$

where $F(X,U)$ is the generalized objective function of the optimization procedure. The function (10) satisfies all of the conditions of the standard Lyapunov function definition. This function is used for the analysis of the design trajectory behavior with the different switch points. We can define the system design process as a dynamic transition process that provides the stationary point during some time. The problem of the time-optimal design algorithm construction is the problem of the transition process searching with the minimal transition time. There is a well-known idea [9] to minimize the transition process time by means of the special choice of the right hand part of the principal system of equations, in our case the form of the vector function $H(X,U)$. By this conception it is necessary to change the functions $H(X,U)$ to obtain the maximum speed of the Lyapunov function decreasing (the maximum of $-dV/dt$) at each point of the process. Unfortunately the direct using of this idea does not serve well for the time-optimal design algorithm construction. It occurs because the change of the design strategy produces not only continuous design trajectories (when we change the strategy $u_j = 0, \forall_j$ to the strategy $u_j = 1, \forall_j$ for instance) but non-continuous trajectories too (in opposite case). Non-continues trajectories had never been appeared in control theory for the objects that are described by the differential equations, but this is the ordinary case for the design process on the basis of the described design theory. In this case we need to correct the idea to maximize $-dV/dt$ at each point of the design process. We define another principle: it is necessary to obtain the maximum speed of the Lyapunov function decreasing for that trajectory part which lies after the switching point. In this case the trajectories with the different switching points are compared to obtain the maximum value of $-dV/dt$.

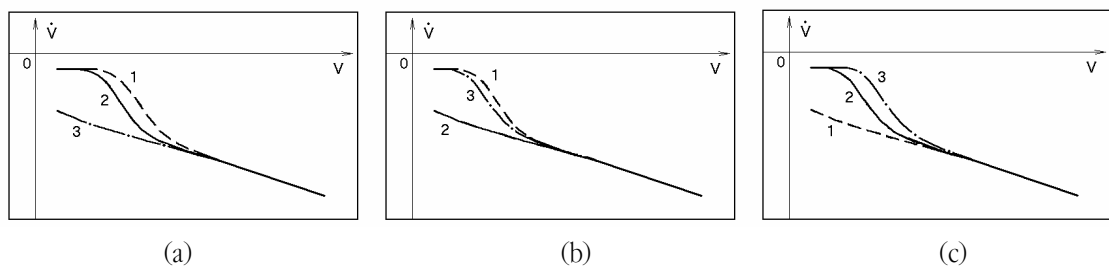


Fig. 2. Time derivative of Lyapunov function behavior for three switching points 1, 2, 3 consecutive integration steps before (a) in (b), and after (c) the optimal point

Technically this idea is realized by comparing some probes with the different switching points and selecting the one of them that provides the maximum of $-dV/dt$ after the switching. Numerical results prove this idea. The behavior of the function dV/dt for the circuit in Fig. 1 is shown in Fig. 2 for three neighbor switching points 1, 2 and 3 that correspond to the three consecutive integration steps before (a), in (b) and after (c) the optimal point.

The optimal switch point corresponds to the curve 3 of Fig. 2 (a), or curve 2 of Fig. 2 (b), or curve 1 of Fig. 2 (c). It is clear that this point corresponds to the maximum negative value of function dV/dt and at the same time corresponds to the minimum value of the total design steps. In this case the optimal switch points can be found and serve as the basis to the time-optimal algorithm construction.

Conclusions

The traditional approach for the analog circuit design is not time-optimal. The problem of the optimum algorithm construction can be solved more adequately on the basis of the optimal control theory application. The time-optimal design algorithm is formulated as the problem of a functional optimization of the optimal control theory. In this case it is necessary to select one optimal trajectory from the quasi-infinite number of the different design strategies, which are produced. This approach can reduce considerably the total computer time for the system design. The above-described approach gives the possibility to find the time-optimal algorithm as the approximate solution of a typical problem of the control theory. The optimal position of the control vector switch points can be obtained by means of the time derivative analysis of the special Lyapunov function of the system design process.

Acknowledgement

This work was supported by the Mexican National Council of Science and Technology CONACYT, under project SEP-2004-C01-46510.

References

- [1] Osterby O. and Zlatev Z. Direct Methods for Sparse Matrices. Springer-Verlag, N.Y., 1983.
- [2] Wu F.F. Solution of Large-Scale Networks by Tearing // IEEE Trans. Circuits Syst. Vol. CAS-23, No. 12. 1976. P. 706–713.
- [3] Sangiovanni-Vincentelli A., Chen L.K., Chua L.O. An Efficient Cluster Algorithm for Tearing Large-Scale Networks // IEEE Trans. Circuits Syst. Vol. CAS-24, No. 12. 1977. P. 709–717.
- [4] Kashirsky I.S., Trokhimenko Y.K. The Generalized Optimization of Electronic Circuits. Kiev, 1979. (in Russian).
- [5] Rizzoli V., Costanzo A., Cecchetti C. Numerical Optimization of Broadband Nonlinear Microwave Circuits // IEEE MTT-S Int. Symp. Vol. 1. 1990. P. 335–338.
- [6] Ochotta E.S., Rutenbar R.A., Carley L.R. Synthesis of High-Performance Analog Circuits in ASTRX/OBLX // IEEE Trans. on CAD. Vol. 15, No. 3. 1996. P. 273–294.
- [7] Zemliak A.M. Analog System Design Problem Formulation by Optimum Control Theory // IEICE Transactions on Fundamen. Vol. E84-A, No. 8. 2001. P. 2029–2041.
- [8] Zemliak A.M. Acceleration Effect of System Design Process // IEICE Transactions on Fundamen. Vol. E85-A, No. 7. 2002. P. 1751–1759.
- [9] Barbashin E.A. Introduction to the Stability Theory. Moscow, 1967. (in Russian).

ИНДЕКС ФАМИЛИЙ АВТОРОВ СТАТЕЙ

Б

Белов С.Д.	137
Бурдонов И.Б.	124

Г

Галушкин А.И.	10
Гладких Б.А.	12
Городняя Л.В.	12, 117
Грошева М.В.	19
Губкина И.М.	113

Д–И

Денисов А.А.	18
Ефимов Г.Б.	19
Завгородний В.И.	20
Зенин В.Н.	27
Зуева Е.Ю.	31
Иванников В.П.	124
Игнатова И.Г.	32

К

Казанцев П.А.	10
Каймин В.А.	33
Карпова В.Б.	34
Касьянов В.Н.	35
Кириянов М.В.	41
Ковшов А.М.	42
Косачев А.С.	124
Коцобан Н.В.	49
Крайнева И.А.	12, 50, 94
Краснов А.Я.	56
Кузнецов Д.Н.	126
Кузнецов С.Д.	124
Кулаков К.А.	58
Куперштох Н.А.	94
Куцос О.И.	66

Л

Лазарев В.А.	27
Липаев В.В.	67
Лисовская И.И.	80
Лисовский И.М.	67, 80
Лукин Н.А.	87

М

Марчук А.Г.	88, 94
Марчук А.Г.	117
Матюхин Б.Н.	99
Мирошников Г.Г.	100

Н–О

Недоря А.Е.	126
Никульцев В.С.	137
Никульцева Е.В.	137
Оганян Г.А.	102
Остапенко Г.П.	10

П

Парахина О.В.	106
Парышев С.Е.	106
Петров А.О.	27
Позднеев Б.М.	112
Поляк Ю.Е.	113

Р–С

Ретинская И.В.	113
Ретинский В.С.	113
Рогов А.А.	115
Самсонов В.А.	19
Смирнов В.И.	116
Соколов А.В.	117

Т

Тарасов Е.В.	126
--------------	-----

Тихонова Т.И.	117
Томилин А.Н.	124
Трахтенберг А.Д.	125
Турсунбай кызы Ырысгуль	125

Ф–Ч

Филиппов В.Э.	126
Филиппова М.Я.	126
Хетагуров Я.А.	133
Черемных Н.А.	50
Черемных Н.А.	94
Чечель Н.В.	136
Чубаров Л.Б.	137

Ш–Я

Шамсиева И.М.	139
Штейнберг В.И.	140
Шуршев В.Ф.	141
Ярмухаметов А.У.	142

А–К

Fitzpatrick A.	145
Hryachev V.V.	135
Idrissov R.I.	33
Impagliazzo J.	146
Jimenez Tomas Lopez	147
Klepper L.Ya.	41, 42, 101
Konashev M.B.	43

М–Z

Molchanova E.V.	41, 42, 101
Nebeker F.	147
Tatarchenko K.	148
Teller R.	148
Zavgorodniy V.I.	26
Zemliak A.	149
Zgurovskiy M.	26

ИНДЕКС НАИМЕНОВАНИЙ ОРГАНИЗАЦИЙ

Американский благотворительный фонд поддержки информатизации образования и науки	113
Астраханский государственный технический университет	100, 141
ГУЗ «Республиканская больница им. В.А. Баранова»	49
Институт вычислительных технологий СО РАН	137
Институт истории СО РАН	94
Институт машиноведения УрО РАН	87
Институт прикладной математики им. М.В. Келдыша РАН	19, 31
Институт систем информатики им. А.П. Ершова СО РАН	12, 35, 50, 88, 94, 117, 125, 126
Институт системного программирования РАН	67, 124
Институт точной механики и вычислительной техники им. С.А. Лебедева РАН	34, 67, 80
Институт точной механики и вычислительной техники РАН	116
Институт философии и права УрО РАН	125
Карельский научный центр РАН	117
МГТУ им. Н.Э. Баумана	18

Московская финансово-юридическая академия	56
Московский государственный институт радиотехники, электроники и автоматики (технический университет)	102
Московский государственный институт электронной техники	32
МГТУ «Станкин»	106, 112
Московский государственный университет приборостроения и информатики	99
Научно-исследовательский центр электронной вычислительной техники	41
Научный центр нейрокомпьютеров НИИ автоматической аппаратуры	10
Национальный университет Узбекистана	139
ОАО «International Computers Limited – Казанское производственное объединение вычислительных систем»	142
ОАО «НИИ вычислительных комплексов им. М.А. Карцева»	27
ОНУ им. Мечникова	66
ООО «Аплер Софтвер»	106

Петрозаводский государственный университет	58, 115	Federation of American Scientists	145
Политехнический музей.....	136	Hofstra University	146
Санкт-Петербургский государственный университет	42	IEEE History Center, Rutgers University.....	147
Томский государственный университет	12	ISI SB RAN	26
ФГУП «НИИ "Аргон"».....	140	National Technical University of Ukraine	26
Финансовая академия при Правительстве РФ	20	Novosibirsk State University.....	33
Центральный экономико-математический институт РАН.....	113	Paris IV, la Sorbonne	148
Booz Allen Hamilton.....	148	Puebla Autonomous University.....	149
Central Economic-mathematical Institute, RAS.....	41, 42	Saint-Petersburg Branch, Institute of the History of Science and Technology, RAS.....	43
		University of Information Sciences	147
		World Distributed University.....	33
		Yaroslavskiy State University	135

СОДЕРЖАНИЕ

А.И. Галушкин, Г.П. Остапенко, П.А. Казанцев

<i>О разработках нейрокомпьютеров в России</i>	10
---	-----------

Б.А. Гладких

<i>Должен ли программист изучать историю вычислительной техники и информатики?</i>	12
---	-----------

Л.В. Городняя, И.А. Крайнева

<i>«Школьная» – пакет прикладных программ автоматизации учебного процесса</i>	12
--	-----------

А.А. Денисов

<i>Исследование архитектурных решений ЕС ЭВМ с точки зрения современных технологий</i>	18
---	-----------

Г.Б. Ефимов, М.В. Грошева, В.А. Самсонов

<i>История использования аналитических вычислений в задачах механики</i>	19
---	-----------

В.И. Завгородний

<i>Перспективы создания защищенной ЭВМ</i>	20
---	-----------

V.I. Zavgorodniy

<i>The AIST project</i>	26
--------------------------------------	-----------

M. Zgurovskiy

<i>Development of educational and research segment of information society in Ukraine</i>	26
---	-----------

В.Н. Зенин, В.А. Лазарев, А.О. Петров

<i>ПЭВМ АГАТ – первый массовый персональный компьютер СССР</i>	27
---	-----------

Е.Ю. Зуева

<i>Работы по машинному зрению в ИМП им. М.В. Келдыша</i>	31
---	-----------

И.Г. Игнатова

<i>Новые тенденции в информационных технологиях – переход к компетентному подходу</i>	32
--	-----------

R.I. Idrissov	
<i>Russian supercomputers software</i>	33
В.А. Каймин	
<i>Технологии Линукс и Open Source в образовании</i>	33
В.Б. Карпова	
<i>История создания БЭСМ АН СССР</i>	34
В.Н. Касьянов	
<i>А.П. Ершов и графы в программировании</i>	35
М.В. Кирьянов	
<i>Интеграция наследованного программного обеспечения в современные IT-технологии</i>	41
L.Ya. Klepper, E.V. Molchanova	
<i>Application of computer technology for optimal radiotherapy planning of malignant tumours</i>	41
L.Ya. Klepper, E.V. Molchanova	
<i>The program complex for local adjustment of parameters of radiological models</i>	42
А.М. Ковшов	
<i>Распознавание текстовых изображений – проект «Открытый код»</i>	42
М.В. Konashev	
<i>Personal computer and personal freedom in «information society»</i>	43
Н.В. Коцобан	
<i>Развитие информационных технологий в Республиканской больнице им. В.А. Баранова Республики Карелия</i>	49
И.А. Крайнева, Н.А. Черемных	
<i>Академик А.П. Ершов и его архив</i>	50
А.Я. Краснов	
<i>Предыстория современного этапа информатизации образования России</i>	56
К.А. Кулаков	
<i>Генерация систем неотрицательных линейных диофантовых уравнений</i>	58
О.И. Куцос	
<i>Применение языка тернарного описания в двуязычном машинном переводе</i>	66
В.В. Липаев	
<i>История развития отечественного программирования для специализированных ЭВМ</i>	67
И.М. Лисовский	
<i>Сергей Алексеевич ЛЕБЕДЕВ – создатель первой в континентальной Европе и в Советском Союзе цифровой электронной вычислительной машины (МЭСМ)</i>	67
И.М. Лисовский, И.И. Лисовская	
<i>Специальные вычислительные машины, используемые в системах наведения С200 и С300</i>	80
Н.А. Лукин	
<i>Функционально-ориентированные процессоры для систем реального времени</i>	87
А.Г. Марчук	
<i>Временный научно-технический коллектив (ВНТК) «СТАРТ»</i>	88

А.Г. Марчук, И.А. Крайнева, Н.А. Черемных, Н.А. Куперштох <i>Сибирская школа информатики академика А.П. Ершова как научно-образовательный феномен</i>	94
Б. Н. Матюхин <i>Из истории развития САПР в СССР</i>	99
Г.Г. Мирошников <i>Особенности проведения реинжиниринга бизнес-процессов в телекоммуникационных компаниях</i>	100
Е.V. Molchanova, L.Ya. Klepper <i>Mathematical modelling and modern computer technologies in planning radiotherapy of the cancer of the bladder</i>	101
Г.А. Оганян <i>Семейство малых ЭВМ «Наури»</i>	102
О.В. Парахина <i>Эволюция преподавания компьютерных дисциплин в московских вузах за 40 лет</i>	106
С.Е. Парышев <i>Разработка программного комплекса, предназначенного для реализации объектно-ориентированных бизнес-систем, использующих реляционную базу данных</i>	106
Б.М. Позднеев <i>Перспективы стандартизации информационно-коммуникационных технологий в образовании, науке и госсекторе</i>	112
Ю.Е. Поляк <i>Проблемы формирования виртуальных компьютерных музеев</i>	113
И.В. Ретинская, В.С. Ретинский, И.М. Губкина <i>Компьютерные технологии обучения в технических вузах с 70-х годов прошлого века до наших дней</i>	113
А.А. Рогов <i>Использование информационных технологий при управлении исследовательской деятельностью школьников</i>	115
В.И. Смирнов <i>Разработка ЭВМ БЭСМ-6</i>	116
А.В. Соколов <i>Опыт обучения информатике студентов математического факультета Петрозаводского университета</i>	117
А.Г. Марчук, Т.И. Тихонова, Л.В. Городняя <i>Новосибирская школа юных программистов</i>	117
И.Б. Бурдонов, В.П. Иванников, А.С. Косачев, С.Д. Кузнецов, А.Н. Томилин <i>Операционная система многомашинного информационно-вычислительного комплекса АС-6</i>	124
А.Д. Трахтенберг <i>«ОГАС» В.М. Глушкова: социальные и культурные пределы технического воображения</i>	125
Турсунбай кызы Ырысгуль <i>О раскраске графов</i>	125

М.Я. Филиппова, Д.Н. Кузнецов, А.Е. Недоря, Е.В. Тарасов, В.Э. Филиппов	
Кронос: семейство процессоров для языков высокого уровня	126
Я.А. Хетагуров	
Некоторые вопросы философии развития вычислительных средств	133
V.V. Hryachev	
PICLAB – The laboratory of nonlinear image processing algorithms	135
Н.В. Чечель	
Научно-исследовательская работа в области истории вычислительной техники в Политехническом музее	136
Л.Б. Чубаров, В.С. Никульцев, С.Д. Белов, Е.В. Никульцева	
История и перспективы сети передачи данных СО РАН	137
И.М. Шамсиева	
О перспективах развития IT и Интернета в Узбекистане	139
В.И. Штейнберг	
Комплекс БЦВМ «Аргон» за сорок лет – динамика развития	140
В.Ф. Шуршев	
Опыт использования информационных технологий для идентификации режимов течения двухфазных потоков	141
А.У. Ярмухаметов	
Концепция создания образовательной сети министерства образования и науки Республики Татарстан	142
A. Fitzpatrick	
Information Technology Policy for the Former Soviet Nations	145
J. Impagliazzo	
Comparative Computing Education: East and West	146
Tomas Lopez Jimenez	
Some results of the cooperation between Soviets and Cubans organization from 1973 to 1989	147
F. Nebeker	
The Importance of Oral History in Researching the Development of Computers in the Soviet Union	147
K. Tatarchenko	
INRIA – Akademgorodok, A history of the Soviet-French cooperation in the field of computer science	148
R. Teller	
Economics of Information Technology: Operations, Investments, and Strategy	148
A. Zemliak	
Principal Aspects of the Minimal-Time System Design Methodology	149
Индекс фамилий авторов статей	156
Индекс наименований организаций	157

Научное издание

SORUCOM.2006

**РАЗВИТИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
В РОССИИ И СТРАНАХ БЫВШЕГО СССР:
ИСТОРИЯ И ПЕРСПЕКТИВЫ**

Материалы международной конференции

В двух частях

Часть 2

Редактор И.И. Куроптева. Компьютерная верстка – И.Г. Лежнев

Подписано в печать 22.06.06. Формат А4.

Бумага офсетная. Тираж 200 экз.

Государственное образовательное учреждение
высшего профессионального образования
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
185910, г. Петрозаводск, пр. Ленина, 33