New SE programs plan to avoid Wang Institute’s mistakes

Galen Gruman, Assistant Editor

Beset by huge financial shortfalls, the Wang Institute of Graduate Studies in Tyngsboro, Mass., closed Aug. 17 after founder and primary benefactor An Wang discontinued funding.

Wang’s closing prompted the Computer Society’s Technical Committee on Software Engineering to express concern about the potential effects on software engineering as an academic discipline, fearing the closure of the highly praised institute could make the start-up of similar programs elsewhere more difficult. “The Wang program was being used as a model. It’s really the momentum that was being established [that concerns us]. And they had a lot of weight,” explained committee chair Lorraine Duvall, a consultant in Rome, N.Y.

But Wang’s closure has not prevented several other software-engineering programs that have geared up in the past year. These programs may benefit from the lessons learned from Wang’s failure, lessons that may prevent similar shutdowns in the future, according to those involved in the new efforts at the Rochester Institute of Technology and George Mason University.

High costs blamed. Wang’s biggest problem was its costs, according to several people interviewed. “It was too expensive to run. It was nowhere near break-even,” said J. Joseph Meng, Boston University’s vice president for external programs. Boston University bought Wang Institute’s physical plant and will use it as an extension campus.

Wang had asked several universities to continue the software-engineering program, but none would because of its high cost, Meng said. “They were hurting for an academic institution to take on the [software-engineering] degree,” he explained, but “we’ve all concluded that the way it’s set up, it couldn’t have been run on a financially sound basis.”

The institute spent more than $100,000 per student but only received a $10,000 return, largely because of its five-to-one faculty-student ratio and small number of students, said James Palmer, director of the new Center for Software Systems Engineering at George Mason University. Wang’s losses are estimated in the tens of millions of dollars. Wang Institute would not confirm the amounts, but the estimates were made by Meng, Palmer, and others.

Despite its costs, the software-engineering program did not cause Wang Institute’s closure, said Susan Gerhart, a researcher at the MCC Software Technology Program in Austin, Texas, and a former Wang Institute professor. The program “was not set up to be an autonomous entity. If you separate it [from the rest of the institute], it’s no more expensive than any other academic program,” she said.

Once An Wang decided to stop funding the institute, it had no way to make up the lost income, Meng said. “They lost their sugar daddy,” said Wiley McKinzie, chairman of the Rochester Institute of Technology’s Computer Science and Technology School.

Before it closed, Wang Institute had 11 professors, 34 staff members, 28 full-time students, and 27 part-time students, a press release announcing the closure said. About 170 students have attended the institute since it was founded in 1981, the release said.

(Wang Institute officials did not return several phone calls to discuss the closure and spokesmen for Wang Corp., a separate firm headed by An Wang, said they could not speak for the institute.)

To avoid similar dependencies on one major source of income, the new programs have broadened their support, integrating them into existing schools rather than making them independent entities.

New programs. In the past year, three new programs have begun or started planning. George Mason University began a software-engineering program in December 1986. The Rochester Institute of Technology received approval in July from the New York state education board for its own program, which begins this month. And Wichita State University in Kansas is planning to start a software-engineering program in 1988 or 1989.

Seattle University’s Software Engineering Dept. is the lone survivor from three programs set up in the late 1970s and early 1980s (Wang Institute and Texas Christian University had the other two). It has survived because it was part of a larger computing program and because there are many large software firms in the Seattle area, said Ev Mills, the department’s chairman.

Software engineering also exists as tracks and concentrations at several universities.

George Mason University has hired four former Wang Institute professors — Hassan Gomaa, Sridhar Ragavan, Bo Sanden, and Dick Fairley (Wang’s software-engineering group chairman) — for its Center for Software Systems Engineering.

The university is making the most of its location in the “Ada belt” around Washington, DC, an area with perhaps the largest concentration of software development in the US, said George Mason’s Palmer. The center, begun with a $2.4 million grant from the state of Virginia, will offer six or seven courses on software engineering (four of which were offered this past year). It is also working with the Software Productivity Consortium in Reston, Va., a group of aerospace firms seeking to improve software development methods.

Like Seattle University, George Mason University serves mainly professionals from local industry: 85 percent of its masters students work for local software developers, Palmer said. Catering to these professionals, the program (part of its Information Technology and Engineering School) will have “some theory, some application,” Palmer said. “It’s not just applications, it’s not just theory,” he said. The program will be split between the Computer Science and the Information Systems and Systems Engineering departments, with the first concentrating on theory and the second on software-systems engineering.

George Mason has 750 masters students and 200 doctoral students, about

Continued on p. 90
OS/2 to open new capabilities for PC developers, users

With an installed base of about 6 million machines, MS-DOS has become business’s standard PC operating system. Its creator, Microsoft Corp. of Bellevue, Wash., has now developed its replacement, OS/2, as part of a joint development with IBM. The new OS/2 is aimed at the Intel 80286 processor, which allows multitasking through a protected mode and faster processing. It also removes the 640K-byte limit on addressable memory.

In an interview with Assistant Editor Galen Gruman, Mark Mackaman, Microsoft’s OS/2 product manager, discussed the new operating system.

Q: With OS/2 soon to be available, what about the millions of MS-DOS users?
A: The two systems will coexist simultaneously. We don’t see OS/2 becoming dominant for two to four years. But it will eventually be dominant in the business operating environment.

Q: Because of IBM’s proprietary MCA bus architecture, will you have to abandon support of current hardware?
A: No. At this point, we’re using the IBM PC AT as our standard. It’s on every desk in business. PS/2 is, for now, nonstandard. It would be crazy to use the PS/2 hardware as the standard and require developers to adapt the operating system for the AT. If PS/2 becomes a standard, we will change the OS/2 developers’ kit to use the PS/2 hardware as the base configuration.

Q: There has been criticism that OS/2’s compatibility box will run only one DOS application at a time and that it works slowly. Why the limit?
A: I don’t understand why that’s a criticism. Under DOS, you can only run one program now, so why require that OS/2 handle more than one?

Q: Perhaps people want to take advantage of OS/2’s advanced multitasking features without giving up their current software investment.
A: Of course. Maybe they also want more memory than 640K in the compatibility box. But then they really want a DOS Plus. Our intent is to provide a clone of the DOS environment that you can use from OS/2. If someone is going to use only DOS applications, then don’t buy OS/2.

Q: If you’re limited to one DOS program, what about the terminate-and-stay-resident programs like Sidekick?
A: Whatever you can do in DOS now, you’ll be able to do in the compatibility box. But you really don’t need TSRs because OS/2 has multitasking. You can just call up a program in another window. We put in the hooks for TSRs [in MS-DOS] because people wanted some sort of multitasking ability in DOS — but DOS wasn’t really designed for it. I’ve been surprised how creative people have been with them.

Q: What should people be warned about when running DOS programs in the compatibility box?
A: When you leave the box, the DOS program stops. It’s in suspended animation. It doesn’t run in the background, even though when you go to the box, your OS/2 programs continue to run in the 80286’s protected mode. This could cause problems with time-sensitive applications. But we had to do that because DOS writes directly to the screen and we couldn’t tell it to stop writing while it was in background.

Q: Why will OS/2 benefit developers?
A: DOS has reached its limits, and OS/2 removes the 640K RAM and multitasking barriers.

Q: Do you think developers need much more than 640K, or are they just using as much as is available?
A: A lot of programs are already up against the 640K ceiling. Look at Symphony; it has a lot of capabilities. They’re not just wasting memory. We can’t extend DOS because that would knock a lot of programs out; there wouldn’t be enough RAM left. OS/2 will let them address 16M bytes of RAM, and with its virtual-memory management, the amount of memory is unlimited.

Q: It sounds like the opposite of a RAM disk. Instead of using memory as a disk, you use a disk as memory.
A: Yes. And it can handle up to 512M bytes per application, theoretically.

Q: Theoretically?
A: Well, we don’t have any programs that big to test! But I’m sure someone will come up with one. Look at AI. Some programs take a megabyte just for the code. Some companies, like Gold Hill, have come up with some tricks to make it work on an AT, but 640K just isn’t enough. I think that’s why AI hasn’t done any better.

Q: Lately, with the new Mac II and the PS/2 systems, it seems that Apple is getting more like IBM and IBM is getting more like Apple. Do you see the two approaches merging?
A: The idea of a common user interface makes sense. Common interfaces make training a lot easier. People now have to learn different interfaces for different products. And it makes sense to have a standard programming interface.

I like to use the car analogy. All the car makers put the gas pedal on the right and the brake in the middle. If there’s a pedal to the left of the brake, you know it’s a clutch. “P” means park and “D” means drive. But you can tell a Chevrolet from a Ford. Each car has a different style even though they follow the same conventions.

Q: I suppose a developer could ignore the tools provided, and it would work in the environment but would look any way he wanted.
A: They can ignore all our tools and design the interface from scratch. But we hope they don’t.

Q: You’re not restricting the developer from doing what he wants as much as you’re providing standards. Is that the difference?
A: We provide the tools the developers use. We hope they will put a command bar at the top and have pull-down menus because that seems logical, but they can put them at the bottom and make them go up. They can do it any way they want.

Q: As someone who came from mainframes, I find graphics interfaces cumbersome for some applications, like word processing and spreadsheets — although I can see how they help new users.
A: I had the same reaction when I started using graphics environments. But they do more than help new users. I really think they open a whole new level to the user if they’re presented right. Look at word processing. You can see the type on the screen. You can get your headlines to appear like you want, not wait until it’s printed. That opens up what you can do.

Well, the same will happen for other applications. Someone will find ways to use the new capabilities to bring the work to a higher level. We’re in a Model T era now. Remember, the industry just started in the late 1970s.
a quarter of whom are enrolled in software-engineering courses, Palmer said. In fact, the program “has increased our total enrollment by a significant amount,” he said. Enrollment has tripled since September 1985.

The Rochester Institute of Technology is also following the industry-oriented model. “We are forming a local software-engineering consortium. We intend to become a major academic player on the software-engineering field,” said computer science chair McKinzie.

The software-engineering program (formally called Software Development and Management because using the name “software engineering” would require that students get an engineering license from New York state), will be housed in the Applied Computer Studies Dept., a division of the Computer Science and Technology School. It will blend courses from the software-engineering program, Computer Science and Technology School, and Business School, said Guy Johnson, chairman of the Applied Computer Studies Dept.

“We’re concentrating on the actual problems of software development,” Johnson said. But the program intends to avoid being dependent on a main benefactor as the Wang Institute was, he said. “We hope for cooperation from local industries to put together projects,” Johnson said. The university’s long-standing relationship with local industry should help this effort, he said, pointing out that “we’ve always had [industry] advisory boards in our departments.”

But the program was set up to work without industry money, McKinzie said. Funds from the consortium will expand the program, not underwrite the basics, he said.

Software engineering’s role. The rise of software engineering in academia has raised questions about the discipline’s role in education. While a degree called “software engineering” is not necessary, the types of courses that were taught at Wang Institute are needed, said the Computer Society technical committee’s Duvall.

“I think the need is very great,” she said. When she was hiring programmers for a large company several years ago, “I didn’t think that traditional computer-science graduates had the engineering background. I wanted problem-solving abilities and methods of approaching solutions,” Duvall said.

“Software engineering has not prospered in academia” despite the field’s growth, said Seattle University’s Mills. This may be partly caused by a rivalry between computer scientists and software engineers, he said, a rivalry much like that between computer scientists and electrical engineers. “Computer-science programs are still trying to establish as a standard what kind of person ought to come out of them,” Mills said.

“There is a substantial management component [in software engineering] since a large part of the problem is project management,” MCC’s Gerhart said. “That upsets some computer-science departments.”

The problem with software engineering is the question “Where does it fit in?” Gerhart said. It is a conflict between fundamental knowledge (computer science) and applying that knowledge (software engineering), she said. But, she added, the two can coexist: “England has been very successful in taking some of the pure, theoretical approaches and making them work in applications. They [the two disciplines] are much more integrated in the universities there.”

Palmer acknowledged potential rivalry between computer scientists and software engineers — “that’s a normal concern,” he said — but said the programs at the university are “complementary by design” and should minimize such rivalry.

Johnson also said he is not concerned about a rivalry between computer science and software engineering: “People in computer science recognize that this is a reasonable academic field.” However, McKinzie said that some of the acceptance may be because software engineering is an area that government and industry will spend money on, not because it is accepted as part of computer science.

“Industry and government will probably have to set up separate research and educational plants from computer-science academia,” McKinzie said. Several such programs do exist: the Software Productivity Consortium, MCC, the Software Engineering Institute, and the Defense Dept.’s STARS program. And the new software-engineering programs are oriented more toward industry’s needs to retain professionals than to teaching undergraduates.

At Seattle University, the rivalry is muted because computer science is an undergraduate degree and software engineering a masters degree, Mills said. Plus, he said, most of the software-engineering students are professionals from area firms like Boeing who are going back to school to sharpen their skills. They are not competing for the same students that the computer-science program is.

Reversing enrollment declines. When set up as a program for professionals, software engineering can reverse declining computer-science enrollments, Palmer and McKinzie said.

In fact, one reason that the Rochester Institute of Technology started its software-engineering program was to recruit students for the Computer Science and Technology School, which has seen steadily declining enrollment for several years, McKinzie said, largely because traditional computer science is not sufficiently oriented to applications. Johnson agreed.

The solution to declining enrollments is to form joint programs with other disciplines, McKinzie said. “Right now, I think that’s a good thing to do. But eventually it [computer science] will integrate itself into all other disciplines,” he said, so these joint programs will no longer be needed.
NSF panel recommends funds for visual tools

A report to the National Science Foundation recommends that federal agencies spend $70 million to $230 million annually on the development of "visualization" hardware and software for advanced scientific and engineering research.

The report, prepared by an NSF-organized panel on graphics, image processing, and workstations, paints a picture of scientists and researchers who are overwhelmed by floods of data. As a consequence, much of the information is wasted because it cannot be absorbed.

Part of the solution, according to the panel, lies in interactive, graphical tools that can "transform numbers into pictures, enabling scientists to observe, interact with, and manipulate the data computers generate."

The term "visualization" is meant to encompass the converging areas of computer graphics and image processing and to apply the technology to scientific computing, particularly at the five NSF-funded supercomputer centers.

In presenting the report at ACM's SIGGraph conference July 29 in Anaheim, Calif., panel chair Bruce McCormick of Texas A&M University said the allocation of funds at the supercomputer centers is biased against visualization technology. McCormick said about 98 percent of all runs at the supercomputer centers are for debugging, which visualization could reduce by helping researchers spot errors.

The panel recommends that scientists and engineers team up with visualization researchers to solve representational problems.

The report, "Visualization in Scientific Computing," and an accompanying videotape are available from ACM, PO Box 64145, Baltimore, MD 21264.

Programmers' tax status up for debate

The Ways and Means committees in the US Senate and House of Representatives later this month will debate whether a part of the 1986 Tax Reform Act that changed the status of independent programmers, systems analysts, and engineers, as well as designers and draftsmen, should be repealed.

Under Section 1706 of the reform law, it is harder for a firm to claim that long-term, independent, technical-services contractors are not actually de facto employees. Section 1706 removed such contractors from safe-harbor provisions in Section 530 of the 1978 Revenue Act that gave independent contractors in several fields, including computer engineering, the benefit of the doubt if the Internal Revenue Service disputed their independent contractor status. Technical-services contractors are now subject to the same rules as other independent contractors.

The safe-harbor provisions benefited firms that hired independent contractors for extended periods because it freed them from the responsibility of withholding income and Social Security taxes from the contractors' earnings. It also freed them from providing benefits such as insurance and vacations.

Sen. Patrick Moynihan (D-N.Y.), who sponsored Section 1706, has argued that it is fairer if technical-services contractors are treated under the same common laws as other independent contractors. Joe Gale, a Moynihan aide, also said the safe-harbor provisions had been applied unevenly.

Several industry groups, including the Information Industry Association, the Data Processing Management Association, and the Computer Society, oppose Section 1706 because they say it discourages large firms from contracting work to independent clients. A July 8 report from the IEEE US Activities Board said Section 1706 and subsequent IRS guidelines prompted Apple, AT&T, Boeing, Ford Aerospace, IBM/CA, Lockheed, and Westinghouse Defense Systems to sever direct-consulting relations with technical-services contractors.

The Computer Society has estimated that 10 percent of all programmers are independent contractors.

The IEEE USAB argued that the the IRS criteria "are more appropriate for clerical and manufacturing-services workers than for highly skilled professionals involved in design, research, and development tasks. No matter how fairly they are applied, the common-law tests cannot fairly discriminate between employees and consultants." Computer Society President Roy Russo has urged Congress to postpone Section 1706's implementation and define the differences between employees and independent contractors for all professions.

NEWS BRIEFS

US to review program-display copyright laws. The US Copyright Office has asked for comments as part of public hearings aimed at redefining copyright regulations for program menus and display.

Under current policies, the Copyright Office "does not register separately textual screen displays, reasoning that there is no authorship in ideas or in the format or arrangement of text, and that any literary authorship in the screen display would presumably be covered by the underlying computer program — itself a literary work," said a July 29 Copyright Office notice.

But recent court decisions have suggested that program displays must be copyrighted separately from the source code (Soft News, July, p. 106, and March, pp. 90-92). Earlier court decisions had held that program display was not separately copyrightable; several cases are still pending.

Because of the conflicting decisions and the apparent judicial trend toward accepting separate copyrights on screen display, the Copyright Office has decided to reevaluate its policy. Written comments should be submitted by Oct. 9. For information, contact Dorothy Schrader, general counsel, Copyright Office, Library of Congress, Washington, DC 20559; (202) 287-8380.

Real-time operating systems said to be in infancy. Despite the recent industry interest in distributed real-time systems, such systems remain a research direction still in its infancy, according to participants at the Fourth IEEE Workshop on Real-Time Operating Systems. The field lacks concepts and experience, they commented. About 50 people attended the July 2-3 workshop in Cambridge, Mass.

The presentations revealed a wide gap between academic and industrial/military development of real-time operating systems. Software concepts developed in commercial environments were very different from those developed in academia. This gap prompted many speakers to defend their basic approaches during their presentations.

— Horst Wedde,
Wayne State University