



“There was no way of knowing that, less than a year later, businesses would be buying NeXT computers in order to run Improv.”

How a skunkworks project at Lotus turned into NeXT's breakthrough application.

Everybody called them “those hacks from Lotus.” They would come to the NeXT user group meetings, smile, listen, and refuse to talk about what they were working on. People assumed it would take Lotus Development Corporation “years” to develop something “uninteresting” for the NeXT, says Paul Kleppner, a Lotus “hack” and principal engineer on the Improv project. • “We couldn’t say anything,” says Lynda Urgotis, the senior technical writer who designed Improv’s hypertext documentation system. “We simply stood there and took it [thinking] ‘We’ll get the last laugh.’” • Of course, there was no way that the people at the user group meetings could have known how powerful Lotus’ secret next-generation spreadsheet would be, or how successful. They had no way of knowing that, less than a year later, businesses would be buying NeXT computers in order to run Improv. • After all, they had never heard about Fluffy Bunny.

BY SIMSON L. GARFINKEL

next generation

The idea that eventually grew into Improv started in 1986, when Lotus Development Corporation took an empty library on the first floor of its building at 161 First Street in Cambridge, Massachusetts, and set up the Advanced Technology Group (ATG).

ATG's charter was to come up with new ideas that Lotus might be able to turn into products. There were three projects and three researchers: a new development environment, an artificial intelligence system that never panned out, and "Modeler"—the seed of Improv.

One of the hardest things about using a computerized spreadsheet like Lotus's 1-2-3 is laying out the initial model: How to arrange the rows? The columns? It is simple enough to start entering numbers and formulas in cells; it is another thing entirely to create a structure that you can grow with.

Enter Pito Salas, a bright developer in the ATG. Salas studied a variety of complicated models that had been built with conventional spreadsheets and discovered that most of them employed a few basic patterns. If a spreadsheet program could be taught to understand those patterns, he realized, it would be much easier to use.

Within a few months, Salas had come up with the fundamental idea at the core of Improv: that the raw data in a spreadsheet, the way that the user views the data, and the formulas used to perform calculations can all be separated from one another. The formulas should be general, so that the user can type something like, "PROFIT = PRICE - COST" and have the spreadsheet calculate every PROFIT cell from its corresponding PRICE and COST cells. The user should be able to rearrange the views to highlight the relevant information and relations. The data itself should be stored in

a multidimensional data structure. A slick interface should sit on top, to make it easy to get information in and out.

By the end of the summer, Salas had a demonstration program working on an IBM PC. It was really more a slide show than a computer program, admits Salas: The math was all faked. The project stalled at this phase until February 1987, when Salas hired Glenn Edelson, a hot programmer whose job it was to take Salas's ideas for general formulas and see if they could be implemented in C++.

"I had described all those [ideas] with hand waving," recalls Salas. "I had no clue how to build it. Glenn came in and we started working."

Things clicked. It soon became obvious that the technology for calculating general formulas could work and that multidimensionality was a powerful way to model complicated financial problems. In the spring of 1987, Salas hired Bonnie Sullivan to write a specification for the project and Jeff Anderholm to see if there was really a market need for a "high-end financial modeling application."

That summer, Modeler went on tour. The "road show," as Salas called it, took the Modeler demo program to six of Lotus's key customer accounts. At each stop they spent a day with a few analysts who did high-end financial modeling.

"We had a paradigm shift on that trip," recalls Salas. "It was really cool. We said that these ideas might be turned into a high-end financial modeling package of interest to perhaps ten percent of your company, and they said 'No, no, no! What you have is the next generation spreadsheet—something that will be used by half of our company!'"

Despite the enthusiastic comments, Lotus still took more than a year to decide to turn Salas's ideas into a product. The decision was

thought we had a high-end financial modeling package, but customers said, 'No. It's the next-generation spreadsheet.'"

finally made in September 1988 to go ahead with the "Back Bay" project (named after a Boston neighborhood known for its townhouses, boutiques, and restaurants). After experimenting with interfaces and a database engine under DOS and the Macintosh operating system, the group decided that the product would be based on OS/2 and Microsoft's Presentation Manager. They even picked a mascot—Fluffy Bunny—and started up an underground newsletter, "Fluffy Bunny Goes to Back Bay."

Green light

A month later, in October, Steve Jobs came to Lotus to show off his new computer. After the talk, Lotus's top management gave Jobs a private show-and-tell of their most interesting products then under development.

Salas showed Jobs a clunky, character-based spreadsheet, but all of the future elements were there. Formulas were entered at the bottom of the spreadsheet rather than integrated into the cells; it was multidimensional; and the user could instantly call up different views of the same data set.

Immediately, Jobs wanted Back Bay for the NeXT.

"He kept getting more excited. He was the most excited person in the room," remembers Salas.

Back Bay fit right in with Jobs's vision, says Allen Olsen, ATG's development manager. "Right from the start, he was looking for something new. It might have been better financially for his company to get 1-2-3, but that would have compromised his vision. [Back Bay] was attractive because it was a new kind of spreadsheet."

Likewise, the Back Bay group was excited by the NeXT. But when Jobs went back to California, the Back Bay group continued working on the OS/2 product—or at least they tried. Writing applications for the buggy initial releases of OS/2 and

Presentation Manager was no picnic. And the allure of Jobs's black cube kept creeping back into their minds.

The decision to switch Back Bay to the NeXT came four months later. Ed Belove, the vice president of advanced technology, left a Post-It stuck on Salas's computer: "Pito—Come see me as soon as you get back. This is very important.—Ed."

"I think that [Lotus CEO] Jim Manzi really believed in us," recalls Jobs. "When the Improv team recommended to proceed on the NeXT (after their failure on OS/2 and Presentation Manager), Jim gave the green light."

There were both marketing and technical reasons for moving Back Bay to the NeXT, says Frank King, vice president of Lotus's software business group. By putting Back Bay on a computer platform that couldn't run 1-2-3, Lotus wouldn't have the problem of explaining to customers which spreadsheet to use for which application. There wouldn't be any marketing conflicts with Lotus's cash cow.

But King also wanted his engineers to experiment with the NeXT development system. "I have a lot of respect for Steve and the technology he has developed," says King. "I wanted to see if the things he was saying about NeXTstep were true."

The switch also gave the Back Bay group the freedom to rethink some of the basic ideas about the user interface. Rather than try to port the code written for Presentation Manager to the NeXT, the group resolved to rewrite the entire user interface from scratch. "We knew that we had a lot of work to do in the UI area, and it seemed that it was quick to do [on the NeXT]," recalls Olsen.

A few days after the decision, Steve Jobs sent a huge bouquet of flowers to Cambridge. "It was like

he was wooing us," says Lynda. "It must have been three feet tall!"

Perhaps more important, NeXT sent technical wizard Bruce Blumberg to Lotus, where he gave the Back Bay developers a special one-week class.

Fluffy Bunny was off and running.

Direct manipulation

The best part about developing a user interface with Interface Builder wasn't the ability to design a single interface quickly; it was the ability to try out a lot of them. "It let us develop a lot of bad UIs," says Salas. "We could have an idea, and in an hour we would have it on the screen."

For example, an early version of Improv's Item Dispenser was simulated with Interface Builder and multiple windows. Other times, developers used the program merely as an electronic blackboard to sketch out ideas before writing the code to make the ideas work.

Perhaps the single most important and unique aspect of Improv's user interface is its Category Tiles, which the program uses to represent different dimensions of a multidimensional data set. But in January 1989, Back Bay didn't have Category Tiles. Instead, all of the view rearrangement was done with menu commands with names like Promote, Demote, and Perspective.

"It had all of these arbitrary restrictions," says Kleppner. Then the group hit upon the idea of using icons. "We realized that if we represented these things as icons, all these manipulations could be represented by moving icons from one place to another."

The Category Tiles also "broke through a lot of barriers in understanding for new users," says Salas. They let users visualize and directly manipulate the dimensions of the multidimensional database.

But where should the icons for the categories go? After trying a lot

of different ideas, the developers decided to put them in their own window.

"And then Steve Jobs came," says Kleppner, remembering Jobs's visit in April 1989. "He said, 'Okay, who's the guy who came up with the picture? It's very important that he be here.'"

Jobs then said that the category manipulation had to be more direct. "You have to be able to touch the categories and move them around. Having them off in a separate window is too removed," Kleppner remembers Jobs saying. "He didn't even want to have the tiles; he wanted to just move them around. He's a fanatic for direct manipulation, and it really shows."

Jobs didn't have an answer, says Kleppner, but "one of the benefits of that [meeting] was that we junked the idea of the extra panel" and put the Category Tiles on the worksheet itself.

Conflict resolution

In a way, Jobs himself became an icon on the Improv development landscape. "He would come in periodically, tell us what he liked and what he didn't like," remembers Kleppner. "We didn't do everything he wanted, but he pushed us in the right direction a lot of the time."

For example, in one issue of "Fluffy Bunny Goes to Back Bay," Fluffy Bunny is warned: "Sometimes formulas are friendly, but sometimes they are Not Very Nice."

Formula conflicts were the problem. Since Improv has general formulas, it's possible that a single cell can be specified by more than one formula. A business's total profit over two years could be calculated either by subtracting the total price from the total cost or by summing the profits for each year.

The Back Bay group thought that the possibility of formula conflicts was a real weakness in the

[CONTINUED ON PAGE 79]

putting
Back Bay
on a
computer
platform that
couldn't run
1-2-3, Lotus
avoided
marketing
conflicts
with its
cash cow.

[CONTINUED FROM PAGE 35]

program. There seemed to be no way to automatically resolve them. When Improv detected a conflict, it invalidated both formulas and complained to the user. Yes, sometimes formulas were not very nice.

"Jobs said, 'This is not a weakness, it is a great feature!'" remembers Kleppner. "With 1-2-3, you can copy and blow away a formula." The fact that Improv could automatically detect conflicts and tell the user about them was a major step forward.

As the product matured, many commands held over from the OS/2 version were eliminated. Meanwhile, the Help system and the Presentation Builder were developed as separate application programs (the Help system was later integrated into Improv). Forty people, including 13 developers, were working on the project. All of the pieces were falling into place.

Back in Redwood City, California, NeXT was struggling to get the NeXTstep 2.0 operating system completed. Originally, the project was called "Warp 9," because that was how fast the NeXT development team had to get it out. As the summer of 1990 progressed, the Back Bay group received a variety of NeXTstep pre-releases: Warp 2, Warp 3, Warp 3i, and finally Magenta. "Not only was it a new version; it had a new numbering scheme!" exclaims Kleppner.

Other changes—as well as bugs introduced in intermediate releases—were far more serious. One bug in the memory allocation scheme caused the malloc() function—and all of Improv—to run incredibly slowly. Another bug with the Mail application prevented people from sending documents.

Then, a few weeks before the rollout of the new NeXT machines, the NeXTstep text object changed completely—a real problem for Improv's Presentation Builder. "We

had spent a fair amount of time optimizing the old one," says Olsen. Days before the announcement, nobody knew if Improv's graphs would have text on them or not.

By introduction day, they did.

During the following months, the newly named Improv group shipped out their beta release of the program. The final version started shipping in February 1991.

Once again, Steve Jobs sent flowers to the developers. Only this time, it was truckloads. A dozen large bouquets came, along with many small ones. "There was one bouquet that was five feet tall—the size of a short doc writer," says Urgotis.

What's next?

Probably the most-asked question about Improv is whether Lotus plans to develop versions for other platforms.

"We are all mindful of the higher revenue opportunities in the [Microsoft] Windows environment," says King, hinting that the development group is already fairly far along the preliminary design stages of the project.

But designing a user interface with Windows development tools is a problem, says King. "There is no doubt in my mind that if we had started to do Improv on Windows, we wouldn't be as far along as we are now. NeXTstep is



It's not the typical Lotus, where we sit in our ivory tower and run ads. We need the feedback."

the most productive development environment for graphical applications, bar none."

Meanwhile, the Improv group is willing to talk only about support of the existing NeXT product. Foremost on the agenda is listening to users' reactions. Lotus has set up a hotline for direct user input, something that is very unusual for Lotus. The company also produces a user newsletter and is conducting extensive interviews to find out what users like about the program and what they want changed.

"It's not the typical Lotus, where we sit in our ivory tower and run ads," says Anderholm. "We need the feedback."

Already, users have made it very clear that Improv needs macros and an Undo capability. But more than that, says Anderholm, "We have to figure out what we can add to the product to help people learn it, because you have to unlearn what you know about conventional spreadsheets like Excel and 1-2-3.

"Our challenge is to convince people that the benefits of this new spreadsheet are worth the cost of switching." ♦

SIMSON L. GARFINKEL
is a Senior Editor at NeXTWORLD.