The Task-Based Interface: Not Your Father’s Desktop

Greg Goth

The typical desktop computing environment hasn’t changed much since the mass adoption of the GUI. No matter what operating system your computer might feature, no matter what occupation you pursue, the files-and-folders paradigm is ubiquitous, accepted, and tolerated as so much digital dogma. But for a handful of researchers, this model isn’t quite right.

“When you walk into your office in the morning, you don’t want to tell your computer, ‘I want to run Word today,’” says Tom Dietterich, a professor in Oregon State University’s Department of Electrical Engineering and Computer Science and director of OSU’s intelligent systems research program. “You really want to say, ‘I want to work on this article I was writing; what Web pages was I looking at at the same time, and what emails did I send that were related to it?’”

The organization model Dietterich is describing is the task-based interface model, in which users can arrange their machine’s entire interface architecture to present only those applications and artifacts pertaining to a particular project. This model’s early adopters speak of it as something that will have to be pried from their cold, dead fingers before they give it up.

Two Prominent Projects

Perhaps the most interesting aspect of task-based interface design is the almost simultaneous development of two intriguing examples. Smart Desktop (www.smartdesktop.com), codeveloped by Dietterich and his OSU colleagues, is a general-purpose application spun off as a standalone company and ultimately purchased by storage giant EMC. Mylyn (www.eclipse.org/mylyn/index.php), released in 2005 as an open source application for developers who use the Eclipse platform, emerged from research at the University of British Columbia. (The application is named after myelin, the substance that speeds up neural messages.) That technology’s developers also formed a company that develops Tasktop (http://tasktop.com), a proprietary version for knowledge workers and software programmers. Tasktop was first released in February 2008.

Both could change the mass-market user interface. Conversely, the file-and-folder model’s inertia could help make them niche products. For example, Mylyn’s open source origins have helped it with Eclipse developers, but one dedicated Mylyn user says the open source ethos might also undercut Tasktop’s commercial potential.

“To be honest, there are only a few existing commercial successes around open source software, and only a few around tools in general,” says Bjorn Freeman-Benson, former Director of Committer Community for the Eclipse Foundation. “It’s a very hard business to be in. Now, I believe the Tasktop people, and perhaps the SmartDesktop people, are going to be financially successful. I believe they have a product people want and a model that will probably work out in terms of revenue. But in general, the whole software market is a pretty hard market to be in, particularly because of the whole open source thing.”

Despite Freeman-Benson’s cautions about the difficulty of creating a mass market for task-based interfaces, he also calls the technology behind Mylyn and Tasktop an epochal innovation.

“Once you use it, you’ll never go back,” he says, likening it to an addictive experience for developers.
I also believe it’s the same for anybody in a modern white-collar job. As far as I can tell, there’s only one current flaw in the system, and that is it’s still slightly too difficult for me to tell it I’m switching tasks. I’d like a big blue function key on the keyboard, but that’s a really minor complaint.”

Another veteran Eclipse user, Vancouver, B.C.-based developer David Green, concurs with Freeman-Benson.

“I’ve been around the technology world a long time,” says Green, who leads development as an Eclipse Mylyn project committer. “There’s always new stuff coming out, but it’s more of the same—and that’s not Mylyn. The technology behind Mylyn is truly innovative.”

Background
The genesis of Mylyn’s task-based interface came at one of the computing industry’s most fertile incubators, Xerox’s Palo Alto Research Center (PARC). In the late 1990s, Mik Kersten, Tasktop’s CEO and cofounder, was working there with Gregor Kiczales on tool support for aspect-oriented programming (AOP). Although Kersten says the Mylyn technology itself wasn’t being explored at PARC, one of the basic concepts behind it became clear to him at that point in AOP’s development.

“One of the core things was, no matter what hierarchy you made to organize your program or your information, there were going to be things that crosscut that hierarchy,” he says. “As with ADP, that concept is fundamental to the task-focused interface. No matter what structure you have in your software or desktop or some semantic listing of Web pages, to get anything done, you’re going to touch all sorts of different things in that structure, and you really need some way to unify those. We wanted to make that happen around tasks.”

Less is More
When Kersten first began working at PARC, he brought with him an attitude common among those who wish to provide users with a compelling product.

“I had very much the ‘more is more’ approach,” he says, “more features, to make it better, to make them more effective. Gregor would come sit behind me and say, ‘Less is more,’ and tell me to delete all this work I had just done. That didn’t make me happy at first, but then I realized how important that was, to make sure the tool was simple. This pervaded all the work we were doing. We knew people were getting overloaded with information in programming technology.”

Coincidentally, he says, other PARC researchers were working on visualization technology that Kersten felt dovetailed with tool work for AspectJ, a Java aspect-oriented extension. A key portion of this visualization research investigated degree-of-interest (DOI) trees.

“The basic principle was, you took some function over the data—say, all the articles you had worked on in the last 10 years, and every document related to those and every bit of background research,” says Kersten. “In software development, it’s even more complex, because on a big system, you can have millions of lines of code. So it’s really like cutting millions of documents. It’s a structured data set with millions of things in it, and I was asking, how could we find just the subset of things that are relevant?”

After working with Kiczales and Charles Simonyi for a time at Simonyi’s Intentional Software, Kersten decided in 2003 to go back to the University of British Columbia for his graduate degrees and definitively tackle his problem with his thesis adviser, Gail Murphy. Rather than rely on heuristics or machine intelligence, Kersten says he and Murphy opted for a DOI approach that monitored and registered the user’s activity pattern, “because it’s my activity that determines the stuff I am going to.” Kersten and Murphy subsequently developed a prototype plug-in for the Eclipse platform and gave it to six IBM developers to try for a week late in 2004. Eclipse was perfect for the prototype, Kersten says. Another important lesson he took from PARC was that in introducing an idea as novel as Mylyn, “you don’t make people change their environment completely to prototype it out. We had the luxury of being able to add a new idea to a fully functioning developer’s tool.”

In their published results of the prototype study, Kersten and Murphy saw encouraging signs but also indications that the model needed modification (www.tasktop.com/docs/publications/2005-03-doi-aosd.pdf). The study’s subjects showed overall improvement in the time spent coding versus navigating (the edit ratio), but they didn’t spend the entire time on active Eclipse development. Kersten and Murphy reported, “The average edit ratio improvement across subjects of 15 percent is promising, but overly noisy. However, the single most active programmer (she actively worked in Eclipse for 19 hours during the week and accounted for 40 percent of the activity across both weeks) reported that she worked on the same task both weeks. Her edit ratio improved by 49 percent.”

“Gail, the study subjects, and I found the experiment very promising,” Kersten says. “But the study only lasted one week. If this model kept growing indefinitely, it would eventually get overloaded as well, since programmers tend to access so much diverse information over the course of a month. This is similar to how bookmarks or recently accessed lists tend to get overloaded over time.”

Kersten also says the most prominent feedback was that the study subjects wanted to be able to store the DOI model and access it later.

“We realized it wasn’t enough to have one model,” he says. “We realized that we had to chunk these DOI models by task. When all those ideas came together—the idea of DOI, of focusing on only the relevant information, relying on the static analysis and heuristics that are already being done by the tools, providing an affordance for task activation, and monitoring to create a per-task interest weighting for each element and relationship. At that point, we could see forward to the shape of the tool that we had to build.”
The key intuition, he says, is that for each task that someone works on, that person cares only about a subset of the system that’s related to that task. Windows and MacOS GUIs, as well as search-based navigation such as Google, always show slices of the whole system.

“At this point we realized that the next step in the evolution of the user interface was to focus on the task at hand,” says Kersten.

**A Two-Pronged Market Approach**

Kersten and Murphy envisioned building further on the Eclipse open source framework by writing connectors to popular desktop tools such as Outlook and collaborative-development-management suites. They felt that this approach would attract more developers while providing nondevelopers the same sort of task-focused desktop.

“Our trajectory has been, ‘Let’s make this extremely solid and permeate it among software developers and, in parallel, move away from [supporting] programmers in organizations to support their administrative staff and management,’” Murphy says, “because our real strength is helping groups collaborate. Today, the version of Tasktop I use has nothing to do with program development, because I don’t get to touch code much any more.”

The initial development of Mylyn and Tasktop was facilitated by an IBM fellowship, a Canadian Natural Sciences and Engineering Research Council grant, and prize money Kersten won in business-plan-writing contests. Now, Tasktop is a self-funding bootstrap operation (Murphy says there is one partner investor). It has three versions—a free starter version for Eclipse, a US$99 Pro version for Eclipse, and a $99 standalone Pro version for Windows or Linux. All three feature Gmail and IMAP (Internet Message Access Protocol) connectors. The purchased versions include full Outlook integration and scheduling via Google Calendar. The latest version also sports tightly integrated browsing with Firefox.

The technology’s devotees are spreading it organically. Freeman-Benson, for example, says Mylyn’s task-tracking aspects are so effective that he stipulates that contractors who equip Eclipse conferences use it in their reports to him.

“We can now interface with them in a distributed fashion,” he says. “It certainly makes the project management side much easier for me. I can look at the tasks outstanding and the tasks completed, and the communications channel works a lot better. Prior to using this distributed task tracking, it was difficult to manage all that, because everybody had their own internal app.”

Java framework vendor SpringSource has also adopted Mylyn for use in its tool suite. Christian Dupuis, who leads tool development at SpringSource, says the Mylyn foundation enables a task-focused tutorial for customers, which eliminates the need to download huge sample applications. The tutorial shows only those Java and Spring elements relevant to understanding the corresponding tutorial explanation.

Green, principal tools architect for Vancouver-based MAKE Technologies, says there are more than a dozen “very enthusiastic” Mylyn users at the company, which specializes in modernizing legacy applications. He says it greatly aids such tasks as dynamic domain diagramming.

Manually creating and maintaining such diagrams, Green recently wrote on his blog, isn’t difficult but is time consuming (http://greensopinion.blogspot.com/2008/12/mylyn-context-driven-domain-diagram.html). He recently investigated creating a way to have the company’s model-driven domain tooling create and maintain a domain diagram based on the task-related domain elements in the active Mylyn context.

“There would be no need to create or maintain such a diagram manually,” he wrote. “It would always exist, and it would always show only those items that are relevant to the currently active task. Suddenly, purpose-specific diagrams become implied, easy, expected.”

Green hopes that the enthusiasm for Mylyn among those developers can translate to wider acceptance of task-focused interfaces and Tasktop.

“There are some core champions here for the Tasktop product. We’re looking at gaining and demonstrating successes in a smaller part of the company and then using those to drive adoption in the rest of the company,” he says.

**Smart Desktop**

While Kersten and Murphy were leveraging fellowship, grant, and prize money to perfect Kersten’s vision, Dietterich and his OSU colleague Jon Herlocker (who came up with the idea for Smart Desktop’s core technology) were receiving grants from the US National Science Foundation and DARPA, which was exploring a technology called CALO (Cognitive Assistant That Learns and Organizes).

Whereas Kersten and Murphy built the Mylyn user community around Eclipse developers, Herlocker and Dietterich immediately began developing their technology (called Task Tracer while under the OSU auspices) for knowledge workers—specifically, for the mass market that uses Microsoft Office and Outlook. In this approach, they also diverged from DARPA’s original concept of CALO as an all-inclusive cognitively advanced desktop.

“But we really felt if we were going to be practically successful, we had to live where the users already did, and we set it up so they would have to change their work style as little as possible,” Dietterich says.

“We just modified the Outlook and Windows and Office UIs in various ways to make them task-aware, project-centered,” he adds. “You can think of what we’re doing as a tagging approach.”

In particular, Herlocker and his colleagues added what they call “search folders” to Outlook, which basically contain search results. According to Herlocker, “In a search query you can say ‘Show me everything that has this tag on it.’ Those [folders] are maintained over time. The
system automatically creates one folder for each project, and the folder contains all the emails related to that project. It’s not a folder in the traditional sense. It doesn’t matter what real folder the email lives in.” Dietterich says he’s so enmeshed in the technology that he jokingly told Herlocker, “You better make sure this succeeds. Otherwise, I’m going to be paying students forever to make sure it keeps working. I am totally dependent on it. I have four-and-a-half years of history in the database with this thing. It becomes such an important tool to you; the longer you do it the more benefits accrue. Once a customer buys into this, they’ll be hooked for life. I don’t know if it’s monetizable, but it certainly gives me great value as an individual.”

Handicapping the Future
SmartDesktop’s future is unclear. The technology’s ownership is twice removed from the OSU lab (Dietterich says he no longer has a financial stake in it). The technology, still licensed by OSU, was purchased by Pi Corporation in 2006. In turn, EMC bought Pi and consolidated it with another subsidiary, Mozy, in 2008, renaming the new entity Decho (for digital echo). However, a Decho spokesman says SmartDesktop will no longer be a discrete product: “Following Pi’s acquisition by EMC, Decho will not release SmartDesktop as a separate product; instead they will incorporate the functions into several products planned for release over the coming months.”

The commercial road for Mylyn and Tasktop seems more visible, if not entirely clear; its creators, for example, are still in control of the code. Kersten retains title as the Mylyn Eclipse project lead as well as Tasktop CEO. Murphy is Tasktop’s chair and chief operating officer. Kersten says users of Mylyn or Tasktop, on the basis of Evans Data Corporation estimates, could number around 500,000. With connectors to Bugzilla, Jira, and Rally, the technology is rapidly gaining “eyeball share” among developers. Not only does the buzz around Mylyn and Tasktop ensure they are mentioned in the same conversation, but the improvement of the open source Mylyn framework also facilitates improvements in Tasktop where applicable.

Murphy is optimistic about Tasktop’s future. “Although Tasktop is still considered a small company, the output of open source and commercial products is amazing,” she says. “We’re two years old. We have 10 people and nobody has left the company. I think that’s a success. You build a strong business one step at a time. I think transfer success started with the open source project being so successful. People are saying it’s the best thing since code completion, and it’s a technology that can do more than support programmers. We have nonprogrammers who have been using it for two years.

“We have two challenges. We are continuing to ask our end-user base to refine and make things simpler, and then I think it’s a lot of hard work to get the word out there. And in today’s marketplace, I think that’s become challenging because there are so many different channels available. We have found that the new word-of-mouth type channels, including the blogosphere and micro-blogging, have been helping us get the word out on the productivity benefits with our technology.”

Is your career foundation solid?
Get the building blocks you need.
Take your career to the next level in software development, systems design, and engineering.
Our experts. Your future.
www.computer.org/buildyourcareer