Friedman Corporation uses Instantiations tool to rapidly reengineer its engineering software

SWT Designer works as a plugin to the Eclipse development platform to provide visual, drag-and-drop screen generation, automatically.

Overview

- ***Solution provider profile***
  Friedman Corporation
  - HQ: Chicago, Illinois & Warwickshire, UK, with offices across USA
  - Founded in 1980, develops enterprise-computing solutions for discrete to-order manufacturers
  - 1100+ user sites worldwide
- ***Challenge***
  - Transform 25 years of 5250 code to real-time, multidisplay GUIs
  - Capture & display 100s of pieces of data simultaneously for each of 70+ GUIs in just a few months
  - Leverage RPG stored procedures, SQL and DB2 data structures
- ***Solution Tools***
  - Instantiations WindowBuilder Pro & its SWT Designer GUI building tools
  - Eclipse platform (frameworks & patterns)
- ***Benefits***
  - Overwhelming market reception
  - Dramatic customer satisfaction
  - Industry-leading application design
- ***Tool provider profile***
  Instantiations, Inc.
  - Corporate HQ: Portland, Oregon, USA, with staff in Massachusetts, Pennsylvania, North Carolina, Ohio & Russia
  - 10-year tool provider for building & deploying high-performance Java & Smalltalk applications
  - Strong relationship with IBM & many high-profile clients & partners in the insurance, telecom, financial, ISV & professional-services industries
  - Two members of Instantiations senior management team coauthored the best-selling book entitled, *Eclipse: Building Commercial-Quality Plug-ins*
  - 2006 Eclipse Reader's Choice winner: Best Eclipse Product
  - International reputation for commercializing leading-edge software technologies & delivering high-performance software products & services

SWT Designer works as a plugin to the Eclipse development platform to provide visual, drag-and-drop screen generation, automatically

You might think it is complicated enough to build new graphical interfaces for 5250 screens that support standard business tasks such as order entry and purchasing. Yet, these types of applications typically involve only 10 or 20 data fields per screen. In contrast, Vadim Skarzhin, programming manager for Friedman Corporation, and his four-man programming team deal with hundreds of pieces of data per screen. Skarzhin's challenge perhaps set a new standard for preserving 25 years of traditional RPG business code and its supporting IBM® DB2® for i5/OS® database while producing near-instantaneous responsiveness in dynamic GUIs.

Friedman, a company that provides ERP solutions for cabinetry makers and other manufacturers in the dimensional product sector, needed graphical interfaces for reasons that are more substantive than user-friendliness. Engineering applications require a more sophisticated user interface to support immediate access to many dozens of views into the product data, including:

- Raw materials
- Finished dimensions
- Costing and pricing
- Build time
- Labor, materials and plant-floor availability
- Delivery considerations
- And other business rules

Traditional 5250 applications (with 24 rows of 80 characters) can only accomplish this type of processing with a series of sessions (see Figure 1). This is far too restrictive in the custom-cabinetry industry. The simultaneous presentation of one product perspective and its many views has become increasingly imperative.

Highly skilled cabinet designers and engineers need to use a sophisticated graphical interface to design, view and price customized cabinetry. After less than a year's development effort, Skarzhin's developers were able to deliver this imaginative, dynamic and powerful interface as one of Friedman's flagship offerings, Frontier® PCM WorkBench. PCM WorkBench, presents 70 unique "views" (see the "Eclipse" side bar) into hundreds of data points for an individual product — all from decades-old SQL queries into tables that have been maintained in DB2 for i5/OS on the IBM System i® platform.

How it all began

From the beginning, Skarzhin knew that the PCM WorkBench would have a more abstract nature than is the case for accounting applications, but business rules were

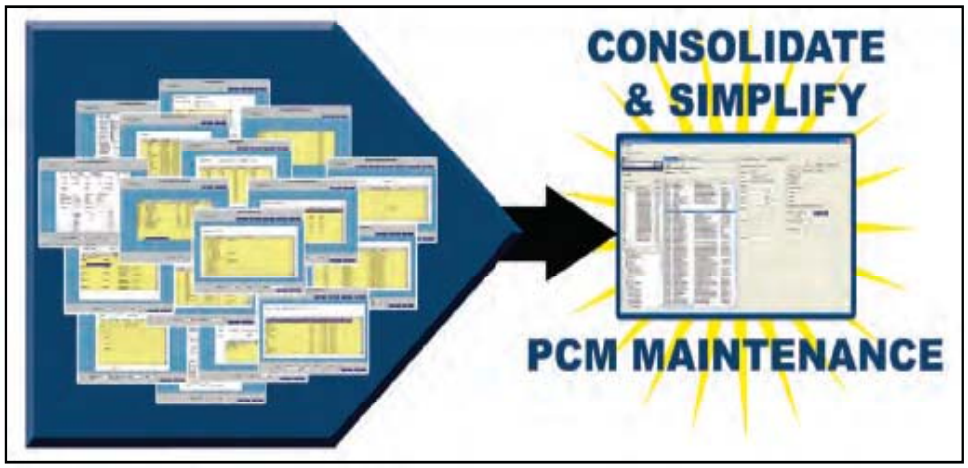

Figure 1: Consolidating dat from seventy 5250 screens into several rich-client perspectives.

still fundamental to the goal of this application development effort. For example, the business rules might involve the engineering complexities of choosing a left- or right-hinged cabinet door. The algorithms for configuring and pricing these complex products are similarly tricky, not only involving dimensions, but also wood choices, paint or stain finishes, quality of hardware and other considerations.

In 2005, Skarzhin was a one-man development team in regard to modernizing Friedman's code, and IBM WebSphere® Java-based technologies were his development tools of choice, although he did use some add-on products to help with the Java coding. These products included the Java-based Eclipse frameworks and patterns, which are functional building blocks that accelerate the software-development process (see the "Eclipse" side bar). Skarzhin also used WindowBuilder™ Pro from Instantiations. In regard to WindowBuilder Pro, he explains that this development tool set (and one of its development components, SWT Designer™) reduced the company's programming efforts by more than 90 percent. In fact, he has been so pleased with SWT Designer that he contacted Instantiations to volunteer as a testimonial candidate. Instantiations was thrilled to hear the success Freidman Corp experienced with SWT Designer, especially considering the fact that Skarzhin never even asked for assistance in learning the product. Stated another way, the development tools that are packaged with WindowBuilder Pro are not only powerful, they are also so intuitive to use that Skarzhin required no ramp-up help from the people at Instantiations. Any tool provider or developer reading this story will immediately be impressed with this phenomenal level of built-in, self-learning usability for a development environment.

Java and Eclipse or .Net technologies?

To be fair, Skarzhin is a self-starter by nature. Early on, with help from Florence DeBello, Friedman's R&D director, he determined that Java™ and the Eclipse development platform were the optimal technology foundation for modernizing Friedman's vast collection of RPG applications. There were several reasons for this. First, Friedman's applications represented hundreds of thousands of lines of IBM Integrated Language Environment® (ILE) RPG code that was well-proven and had already been separated into stored procedures and categorized by business rules. Likewise, the DB2 tables had already been enhanced to include as much of the data rules as possible. It was no more plausible that Friedman walk away from its well-structured legacy applications than it was to assume that the company could survive by clinging to 5250-based user interfaces. The application logic had to stay while the green screens had to go.

There were additional considerations for continuing to develop and deliver on the System i platform on which the RPG code ran. As is true for all IBM systems, the System i platform offers many development tools that are founded on Java and the Eclipse development platform. As such, applications created with these tools are portable across all Java-based environments, including the Linux® and Apple Macintosh operating systems, which are very popular in the engineering and design industries where Friedman finds most of its customers.

The other alternative would have been to move toward a Microsoft® .Net development environment with its platform-specific technologies and tools. However, this option required convoluted methodologies for supporting Linux and Mac users and, importantly, did not readily provide a means of reusing the application code in which Friedman was heavily invested.

"Instantiations' WindowBuilder Pro, coupled with the high levels of functionality incorporated into DB2 for i5/OS, has allowed us to produce engineering-quality graphical interfaces that run on the Linux and Macintosh operating systems, while still using mature RPG code and data structures that are literally decades old."

–Vadim Skarzhin, Data Processing Manager, Friedman Corporation.

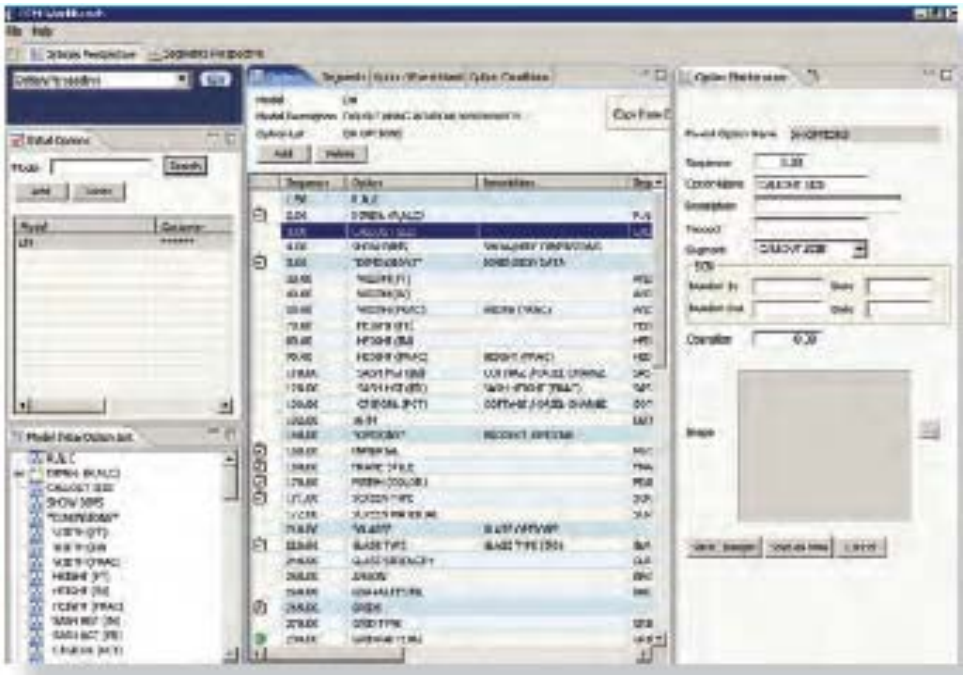

Figure 2: This perspective shows four product views. Additional views are readily available by clicking other tabs.

Development efforts hinge on the right tool

Even with the extensive use of frameworks and patterns, the graphical coding went slowly. The first release of the PCM WorkBench offering would need to provide about seven perspectives and between 70 and 80 views — and each view might contain hundreds of pieces of data. The sheer volume of screens and the data that had to be presented on those screens required meticulous effort on the part of the Friedman developers. It was only possible to develop one or two views of a particular perspective in a week's time.

Skarzhin needed a visual development tool so that his team of developers could verify the accuracy of their Java coding in real time. That is when a product from Instantiations, called SWT Designer (a component within the WindowBuilder Pro development toolset), became attractive. In the autumn of 2005, Skarzhin's team began investigating SWT Designer (SWT stands for Standard Widget Toolkit) and immediately saw the potential for huge gains in productivity. Instead of producing two views in a week, they created five or six views in a single day.

A multitiered, multitechnology solution

Today, Friedman's application model is three-tiered. The user interface tier and the middle tier are both Java-driven. Skarzhin attributes SWT Designer with having been used for between 60 and 70 percent of the total development effort for these two tiers because of its integration as a plugin with the Eclipse frameworks, making the development effort more visual and much quicker.

The middle tier drives the interaction between the user interface and the data. For example, when presenting data to the client, the Java code generated by SWT Designer sorts and performs lookups across multiple products. The data, which resides on a third tier, is maintained in DB2 and is accessed and updated via SQL statements and RPG stored procedures that are encapsulated in APIs. Some of the logic that originally resided in the display files (the data definition specifications, or DDS) is also wrappered into APIs that interact with DB2.

In spite of this level of application modernization, the extensive reuse of code has not adulterated the original RPG code. Skarzhin is proud to point out that the business-

The Eclipse Development Environment

Eclipse is an open-source Java development platform that is comprised of extensible frameworks, as well as tools and runtimes for building, deploying and managing software. Originally developed and donated to the open community by IBM, the Eclipse platform is extended, complemented and supported by major technology vendors, innovative start-ups, universities, research institutions and individuals. Its popularity has exploded in the last five years to become a world-class Java development environment that has an 80-percent share of the Java integrated development environment (IDE) market.

The Eclipse frameworks provide groups of Java patterns, which are groups of reusable software components that can help speed the process of developing applications. Java and Eclipse vernacular refers to perspectives (which the .Net world calls "windows"). Within a perspective, there can be many views, which refers to the various arrangements of data within a window as its dimensions are reduced or expanded .

Note: Two key members if the Instantiations management team, Eric Clayberg and Dan Rubel, are the coauthors of the widely praised best-seller *Eclipse: Building Commercial-Quality Plug-Ins*.

For more information

Contact your IBM sales representative, Friedman Corporation (www.friedmancorp.com), Instantiations, Inc. (www.instantiations.com) or visit IBM at: **ibm.com**/server

logic code, which drives the users' perspectives and views and also updates the DB2 tables, is still maintained by the same SQL queries and 5250 character-based screens and programs that have been in place for years. And, as is a paramount benefit of all object-based applications, changing the source code at the stored-procedure level changes it instantly throughout the entire stream of application references.

Why SWT Designer?

Brian MacDonald is the Marketing Programs Manager at Instantiations. He was able to shed some light on the extreme popularity (and success record) of SWT Designer for similar development efforts. This visual tool is a plugin to the Eclipse platform and is very useful when there is a need to rapidly develop screens that must present large quantities of data elements, which, otherwise, would be unwieldy and time-consuming to code in a nonvisual environment — its drag-and-drop feature aids in placing fields precisely on the screen. For this reason, MacDonald is very familiar with (and proud of) comments that allude to 10-minute screen builds instead of two hours. After the developer places the data on the screen, SWT Designer automatically generates the Java code (using Eclipse patterns). The resulting GUI layer comes out looking nice on its first iteration. The generated Java code is also absent any overhead code from the generating tool that would slow down its runtime performance.

Skarzhin agrees with MacDonald's description of the product, with one addendum, "An important differentiation between Instantiations and other Eclipse tool providers is that Instantiations' expertise focuses solely on Eclipse and Java. Coupled with their two folks who 'wrote the book on Eclipse,' and their intimate relationship with the WebSphere and Rational® people at IBM, this is a tremendous advantage."