

Case Study: Implementation of a Robust .NET Architecture

The Company. This Client is a leading supplier of discovery, collaboration, and knowledge enterprise solutions, desktop software, scientific databases and consulting services to the pharmaceutical, biotechnology, and chemical industries. The Company provides: enterprise solutions, desktop software, scientific databases, and professional services for biotechnology, drug discovery and chemical research, including software, databases, and web sites which enable customers to create, analyze and communicate chemical, biological, and scientific information more effectively.

The Products. This Client's products are used primarily in the pharmaceutical, biotechnology and chemical industries, as well as in higher education and in academic and government research. The company's principal software is the de facto standard and primary

communication tool on the chemist's desktop. The enterprise version of this solution enables research information organizations to deploy application and information solutions using Internet, intranet, and extranet technologies. These solutions are now in use by companies such as Abbot Laboratories, Johnson & Johnson, Merck, etc.

The Challenge. Design and build an application framework granting support to all the client's life science software, and migrate its current applications to .NET technologies.

Considering that the application framework was for the most frequented web site and the most popular chemistry site on the internet, designing and building it was no small task.

The speed at which technology is advancing, means that it is hard to keep up to date, while not distracting attention from the existing tools, but still providing enhanced features and increasing performance, for the continuously increasing data.

As well as the framework, it was necessary to design and develop a practical, useful interface to ease consulting and updating the already voluminous chemical databases.

Further developing in the database environment was also necessary for the complete solution to fulfill the desired objectives.

"We set out less than one year ago with what was an impossible task, but thanks to your hard work and dedication it has been a great success."

The Solution. The chosen architecture was CSLA, which stands for Component-Based Scalable Logical Architecture. It is a multilayer, service-oriented, built on top of CSLA .NET business objects. This demanded research and training, while keeping with the project objectives and timelines.

The chemical finder interface was developed using C#, together with the components:

- Infragistics 2008 suite of controls
- Ultra WinGrid form Infragistics

- Greatls, which aids in form design

All of these components allow designing a more attractive interface, setting out available functions in a clear way, easily seen and used by the scientist who is daily interacting with them.

Developing in the database environment, as well as performing administration tasks and support, was carried out using PL-SQL, Oracle.

The Results. Our team along with the client's specialists, developed an architecture that allows modifying its application's look & feel and behaviors, without the need to recompile. Only the .xml files are modified. This architecture was built on top of CSLA Business Objects to get this framework's benefits. CSLA .NET simplifies and standardizes implementation of business logic, validation and authorization logic within your objects. The goal is to provide an easy and consistent coding pattern by which you can encapsulate all your business logic within your object-oriented business layer. The result is a business layer that can support numerous interface types, while remaining decoupled from any specific interface technology.

Today, the application framework interacts with other areas, such as, database administration and chemical finders, providing the chemical scientists with their already beloved tools, enhanced by the new developments and features.

The Technologies.

✓ Architecture:

- Multilayer, service-oriented, built on top of CSLA .NET business objects

✓ Languages:

- C#
- ASP .NET
- PL-SQL (Oracle)
- Python (addins)

✓ Components:

- Infragistics
- CSLA .NET framework (component scalable logical architecture)
- YUI (yahoo user interface)
- Microsoft Enterprise Library

“In addition to meeting the customer expectations, the work accomplished by this project team has made our new generation of .NET based products a reality. This has allowed the sales team to bring us new business which will surely keep us busy for the years to come...”

We invite you to learn how a relationship with Belatrix will give your company a distinctive advantage through low cost, disciplined, and high quality software development and quality assurance services.

You may contact us at:

businessdevelopment@belatrixsf.com.

Phone number: +1 (617) 608-1413 x2001