



“DSNet’s whole approach in the project was very impressive. They really did a very thorough job assessing the underlying problems with the eCommerce system architecture and transforming it into something that is maintainable and can accommodate future requirements.”
Andy Grove, CTO,
CodeFutures

FireStorm/DAO Saves DSNet Over 300 Hours on eCommerce Project

Oracle Database Administrators Manage Java Data Access Tier Generation

DSNet, the Ohio-based Web hosting services and consultancy provider, has completed a major overhaul of an online eCommerce Web site that handles \$10 million annual turnover and up to 400 concurrent users. The underlying data access architecture from the Oracle database was upgraded using FireStorm/DAO and a new merchandising platform was also implemented in half the time it normally would take due to the new data access architecture.

The Challenge

DSNet started the project with an eCommerce system that was over 7 years old containing in excess of 500,000 lines of code across the Web platform, merchandising system, feed management, and a number of other subsystems. Several consulting companies had been involved in intermittent periods, leading to a very chaotic architecture. “The chaos was almost exclusively centered on data access.” explains Cris Daniluk, Vice President of Technology, DSNet. The problems included:

- Database tables were significantly changed, but the underlying data access code was not restructured to avoid breaking code in other parts of the system. This led to the use of a patchwork of very awkward adaptor patterns.
- SQL was often updated to drop a field, but the field was not dropped from the corresponding Data Transfer Object (DTO). The result was that the Java data access tier developers continued to believe that the field was still in use.
- DTOs were used heavily to manage data. Each DTO contained methods to “load” itself from a ResultSet, save itself, and so on. Changes were frequently made to the database and not updated in all of the data access code. This was most notable when a field was removed.
- The lack of a unified structure made any caching strategy impossible.

“Basically, we needed to make significant changes to the underlying platform architecture before it was safe to add new features” explains Daniluk.

The Solution

Firestorm/DAO allowed DSNet to create a new data layer that accurately represented the domain model. Because the objects were self-managed, legacy save/update/fetch code was entirely eliminated from the DTOs. The data team, which was responsible for the databases, managed Firestorm/DAO, so that when they updated the database, they were simultaneously updating the entire code base. "Having database developers use FireStorm/DAO is very important organizationally, because it means that the data layers are always synchronized. It not only means that time is saved writing manual code, but also there's no time lag waiting for data tier developers to catch up with the database developers" notes Daniluk.



Some technical project highlights included:

- The centralized type-managed DTOs made centralized validation very simple and effective. By attaching a validation object to each DTO, DSNet quickly consolidated all of the business level validation.
- In order to continue supporting legacy code, DSNet was able to use the clean DAO code generated by Firestorm/DAO to develop a set of simple adapters, linking existing deprecated DTOs to new Firestorm/DAO DTOs.
- A caching system was developed to load the entire site product catalog periodically. Firestorm/DAO greatly reduced the time to develop this caching system

Choosing FireStorm/DAO

With data persistence as the key element of the project, DSNet was very careful with its choice of development tools. Daniluk was disappointed with the quality of many tools compared with FireStorm/DAO: "We found a lack of maturity in many other object relational mapping tools compared with FireStorm/DAO".

Providing a tool that the database developers could use was a key differentiator. Daniluk explains: "Hibernate, the only serious competitor, lacks a good modeling and schema importing tool. Their default strategy of using Xdoclet tags in code to generate the Hibernate schema is self-defeating. Firestorm/DAO's modeler allows the data team to manage the DAOs/DTOs".

DSnet also identified other technical reasons for choosing FireStorm/DAO. "One of the technical factors contributing to choosing FireStorm/DAO's was Oracle/PostageSQL sequence management." according to Daniluk.

Technical Benefits

The technical benefits identified by DSNet included:

- By ensuring the code matches the database, hard-to-find runtime SQLExceptions are converted to compile-time errors. This greatly increases code coverage and testability.
- FireStorm/DAO eliminates the need to write difficult JDBC code and ensures a consistent data access strategy. There are often consistency problems with JDBC, where developers have lots of flexibility, particularly to write bad SQL.
- FireStorm/DAO Architect Edition provides a framework for caching and other performance enhancing techniques.

Project Results

"Firestorm/DAO saved us about 300 hours of development in unifying the legacy DTOs, developing test harnesses, implementing a caching subsystem, and eliminating bugs caused by out-of-sync data access code" estimates Daniluk.

A 400% return on investment was achieved with FireStorm/DAO Architect Edition. So for DSNet, FireStorm/DAO Architect Edition paid for itself four times over in time-savings alone in the first three months of use. The longer term cost savings will be substantially more due to lower maintenance.

There are longer-term benefits to moving toward a standardized data access tier. As soon as the architecture had been updated, new functionality was added that tested the longer-term sustainability and maintainability of the new architecture. "The Firestorm/DAO-backed data access platform has undergone substantial development including a brand new merchandising platform, written in half the time it could have been without the data access platform provided through FireStorm/DAO" explains Daniluk.

Copyright © 2005 Code Futures Software Ltd. All warranties on this document are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

CodeFutures™ and FireStorm/DAO™ are trademarks of Code Futures Software, Ltd. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Other company, product, and service names mentioned in this document may be trademarks or service marks of others.

Contact Details

Web: www.codefutures.com

Email: sales@codefutures.com

CodeFutures Software
Interlocken Center
11001 W. 120th Avenue
Suite 400
Broomfield, CO 80021
Fax US: +1 303 484 5134
Phone US: +1 303 604 9669
Toll Free US: 866 604 9669

CodeFutures Software
Suite 32, Dorset House
Duke Street
Chelmsford, Essex
United Kingdom CM1 1TB
Fax UK: +44 845 280 1775
Phone UK: +44 777 56 88 801

CodeFutures Software
1 Parkgate Place
Parkgate Street
Dublin 8, Ireland
Fax Ireland: +353 1 6335084
Phone Ireland: +353 86 2607139