

SPS Technologies

SPS architecture is based on N-tier model where Loose coupling is a key design issue to cope with future technology maturation. The figure below is a simple block diagram of the architecture showing the solution.

Lowering the coupling between the layers and optimizing interchanges are achieved by serializing data objects into binary or text streams, and, decoupling them from their original data sources. A data object can be stored in many ways in many containers, hence it is important not to directly include the mapping code in the data objects themselves. An abstract view of any ISFP module may be illustrated as follows:

In order to guarantee that tiers (basically the business logic and the access tiers) stay independent the whole modules are designed as factory patterns. A less abstract view illustrating the interaction with clients may be shown as follows in the following diagram:

One of our basic methodological decisions was to use Plain old java objects (POJO) rather than new evolving enterprise standards such as EJB's. EJB is not the best choice for fine-grained, in an object oriented development process. EJB is more expensive to code and more expensive to deploy than a POJO business interface and implementation. Irrelevant from an API perspective, most application code can be written without a dependency on the container. The quality of the container implementation and its capabilities towards applying declarative services to POJOs are both very important and very relevant. Refactoring, the disciplined technique for altering the structure of an existing code base to improve its design quality while not changing any of its functional characteristics, is a basic element is choosing any of the frameworks used in ISFP. The following diagrams sketch out the different frameworks used for implementing the different ISFP modules:

Object relation mapping is mainly handled in the persistence layer. Hibernate as an Open Source framework was chosen to implement the persistence tier. Hibernate is powerful, ultra-high performance object/relational persistence and query service for Java.

Hibernate allows to develop persistent classes following common Java idiom - including association, inheritance, polymorphism, composition, and the Java collections framework. Hibernate allows to express queries in its own portable SQL extension (HQL), as well as in native SQL, or with Java-based Criteria and Example objects, thus giving the chance of being completely independent of the DBMS. The Business Layer was implemented upon the spring framework.

Spring is the leading full-stack Java/J2EE application framework; Spring delivers significant benefits thus reducing development effort and costs while improving test coverage and quality. The two basic modules used were the core and the ORM spring packages.

The Core package is the most fundamental part of the framework and provides the Dependency Injection features allowing managing bean container functionality. The basic concept here is the Bean Factory, which provides a factory pattern removing the need for programmatic singletons and allowing decoupling the configuration and specification of dependencies from your actual program logic.

The ORM package is used to integrate with the object-relational mapping APIs, Hibernate, thus giving the chance to use Hibernate in combination with all the other features Spring offers, like simple declarative transaction management. There are literally several web development frameworks available, Struts was chosen since it's the most stable and provides some key benefits;

it's based on a Model MVC implementation, support for internationalization (I18N) a critical point since we develop multi language applications, includes a rich set of JSP tag libraries and contains many extension points for customization. Because data are heavily distributed and transferred between the different modules of the SPS system the 2MVC model is a key design issue in the developed system. The need for a highly interactive system managing heavy data transfers is managed through a set of AJAX library system developed in the company.

The AJAX scripts running on the client side offers a lot of quality added features to the SPS system. Both XML and JSON (JavaScript Object Notation) are used to interchange data between the client and the server tiers, JSON is a lightweight data-interchange format. Reports are developed using crystal reports, while an ad-hoc reporting tool is presented to provide users with the ability of constructing new reports in a fast and easy way. One of our application strong features is our graphic module; we use the latest technology to visualize some tasks on the web for the first time like berth planning and yard planning.

A different user interface library was also implemented using macromedia flash technology. These proprietary modules were designed to achieve more graphical features and better performance for special users with higher cost. The abstract architecture of one of the modules, viz. the Berth planning module, which is heavily based on a rich graphics interface, is shown in the second figure.