

## Implementing a Client Server 3 – Tier Project using Rose 98

### Authors

J. Ramesh – Project Manager – Starling, IBS  
Vinoos Sebastian – Software Engineer – Starling, IBS

### The Company

Located in the beautiful environment of Technopark, at Trivandrum, the state capital of Kerala, International Business Services Group (IBS) is a company established in India to provide offshore software development services to large international clients, particularly in the Middle East, Europe and North America. IBS is promoted by a group with extensive experience and knowledge of Airline business and world class software development services. IBS undertakes software production as a managed service to its world-wide customers, cost effectively and to quality.

The Swissair Group of Switzerland is a very large, high quality organisation and is one of the leading software providers for the air transportation industry. IBS undertakes off shore development of small, medium and large size projects for Atraxis - the IT provider for the Swissair Group. All projects involve cutting edge technology and the use of Java, C++, VC++ Oracle 8.0, DB2 etc. on platforms such as Window NT 4.0 and AIX 4.3. Team sizes range from 4 to 30.

The Airline industry is one of the most important IT driven industries with computer technology being used in all sectors – reservation, passenger and cargo handling, ground operations, flight control, staff rostering etc. Airline related software projects are very complex. This calls for a thorough understanding of the underlying business requirements, designing and documenting them and providing flexibility in meeting ever changing processes arising out of competition and increasing business demands. Airline software also requires to be maintained over long periods running into many years. Typically this often results in software maintenance issues due to the non-availability of the original designs. Therefore a good methodology for analysis and design is essential. With these considerations in mind, we decided to adopt UML as the language supporting OMT methodology for the analysis and design for all our large projects. A number of tools in the market were evaluated and Rose 98 was selected because

- of its support of UML 1.1
- features of code generation facility for Java, C++ and Oracle 8.0.
- Good product support in India.

# IBS

---

## The Project: STARLING

STARLING is a [Staff Rostering and Labour Reporting](#) application capable of handling staff rostering requirements at airports. Though it is being developed for Airline industry it has potential to be sold as a product to any industry where rostering is applicable. It was decided to develop the application using 3-tier client server architecture with Java as the front end, C++ on Unix as the application layer and Oracle 8.0 as the database. The roadmap for STARLING is to enhance to a web enabled CORBA compliant system with a rich set of features satisfying diverse customers in the Airline industry. It was decided to build the system using the best tools available because of the complexity involved and a need to be flexible enough to make improvements / changes as required. The issue of maintaining the system over a period of 5-6 years was also considered.

The project was planned to have a longer design and analysis phase wherein all requirements would be captured and documented to aid the later phases of the project.

## Requirements and Use Cases

The project started off with the study of the business requirements. Use cases were created to capture the requirements purely from the user's perspective, without being influenced by implementation details of the development platform chosen. The team initially drew the use case diagram in Rose and documented the various scenarios / flows as per guidelines given by Rational Rose handbooks. Later the format had to be modified to suit the customer needs. This is where the flexibility of Rose methodology came in handy. The format quickly underwent changes to include critical validations and exceptions that are to be handled by the system. The use case view in Rose and the ease with which use cases could be drawn enabled the team to depict the complex functionality in a simple and easy-to-understand manner. While this enabled the dissemination of the business knowledge within the team, it also helped in assuring the client that their requirements were fully understood. The use case documents that were created in MSWord could be attached to the corresponding use case. Thus we were able to place all the analysis and design models and documents in a single Rose model repository, thus eliminating the search for information from various sources.

Another Rational product, SODA was used to create a comprehensive set of documents every time a review with the client took place. This reduced our documentation efforts and the time required in reviewing the design.

## Design and Class Diagrams

Once our understanding of the requirements had to a good shape, we went ahead with our design and started identifying classes, attributes and methods in the class diagrams. This was followed by drawing sequence diagrams and collaboration diagram. This helped in validating the use cases and classes and formed the foundation for coding. The iterative nature of the design process advocated in Rose helped us to try out various

# IBS

---

design options to satisfy requirements of flexibility, technical limitations, response time etc. The round trip engineering feature available in Rose for Java and C++ enabled us to reduce the coding time required and at the same time allowed us to design in small steps, thus ensuring consistency.

Since the Rose model was an integrated store of the requirements and the design (Use cases and class diagrams respectively), transmission by mail and review by our client was easier and faster. This is very important in the context of our being an off shore development centre.

The support provided by Rose for JDK 1.1.4 enabled us to plan out the sequence of operations possible in each screen. An issue, which arose, was the decision to create the screen for the GUI using the Visual Café IDE. This led to the issue of including the screen classes into the Rose model since all the business classes were already present in the model. The reverse engineering feature of Rose came in handy for this. We reverse engineered the Java files created in Visual Café and included them into the model. By using the JDK1.1.4 framework and the associated availability of Java class methods, comprehensive sequence diagrams could be created. This enabled the team in planning out the screen functionality without having to do screen development through pure JDK which would have been more time consuming than with using Visual Café.

C++ being well supported in Rose 98, the team involved in developing the application layer of the system could easily develop the class diagram, generate code, develop it further and reverse engineer so that the source code and the design were always in agreement.

Rose support for Oracle 8.0 was also extensively used in the development of the database design.

SQL generation from the class diagrams was a useful feature cutting down the time required in creating the scripts in our complex database design.

A comprehensive reverse engineering of the project database schema was carried out and it made the database design more comprehensive. The changes made on the tables were therefore reflected in the design immediately.

A more involved PL/SQL or Java/C++ programming interface with Rose may be a useful feature for developers, avoiding the need to go out of Rose to write code.

## Rose in a Team Environment

The very important issue of managing a large team of more than 30 people working simultaneously on the design of a single system is taxing for any Project manager. The concept of controlled units in Rose ensured that team could work on different parts of the same model simultaneously and at the same time gain access to any module when required.

# IBS

---

## Integration with Configuration Management System

The need for configuration management and version control of units forced us to look into the integration of Rose 98 and Intersolv's PVCS. Since Rose supports nearly all version control systems supporting the Microsoft SCC API, we were easily able to check in and check out the model units directly from the Rose window itself. We did face some problems but Rational Rose was always there to support us quickly with periodic updates of patches.

## Summary

Rose has been used quite extensively in the analysis and design phases of the project and more importantly has been used in the development of all three layers of the system. It has been used end-to-end in the prototype developed for Starling.

The team will be continuing to use Rose for further phases of the project when component diagrams and deployment diagrams are to be developed. We believe that the existence of Rose model will help us in making improvements/enhancements of the product in coming years. The use of Rose for our project has also triggered interest in other project teams to look at the methodology and effectiveness of the tool more closely.

We feel confident in saying that we have a good team of developers who can use Rose effectively for a complete life cycle project of any complexity.

## Quotes: -

J. Ramesh (Project Manager – Starling) – “Worldwide, good software design and documentation has proved to be critical for success of large projects, especially those undergoing Year 2000 conversions. I believe that with tools like Rose 98, our product design and source code will be in step right through the various stages of the evolution of Starling .”