J2EE for the Public Administration: A Success Story

Maurizio Lancia
Italian National Research Council
Piazzale Aldo Moro 7
00185 Rome - Italy
+39 0649931
maurizio.lancia@cedrc.cnr.it

Paola Garzenini
Italian National Research Council
Piazzale Aldo Moro 7
00185 Rome - Italy
+39 0649931
paola.garzenini@cedrc.cnr.it

Alessio Marchetti
Italian National Research Council
Piazzale Aldo Moro 7
00185 Rome - Italy
+39 0649931
alessio.marchetti@cedrc.cnr.it

Roberto Puccinelli
Italian National Research Council
Piazzale Aldo Moro 7
00185 Rome - Italy
+39 0649931
roberto.puccinelli@cedrc.cnr.it

ABSTRACT
Our poster shows how J2EE technologies, well-known design patterns and standard methodologies have successfully been applied in building a complete, robust and well-documented accounting application. Our experience demonstrates that those three factors allow for easy maintenance and reuse of components. The application, whose development was initially outsourced, is now maintained by a CNR internal group and other Public Administrations have shown interest in adopting some of its components.

The poster tells, in brief, the story of the application from the planning phase to the latest developments.

Categories and Subject Descriptors
J.1 [*Administrative Data Processing*]: Government, D.1.5 [*Programming Techniques*]: Object-oriented Programming.

General Terms
Design, Documentation, Management.

Keywords
Accounting, J2EE, component-based development, Public Administration, software maintenance and reusability, design patterns.

1. Introduction
CNR, the Italian National Research Council, is an institution which employs about *8000* people, with a *750 million euro annual budget*. At the end of the year 2000, CNR started planning the development of a completely new accounting system, aimed at providing a more effective management of its financial resources.

OOPSLA'04, Oct. 24-28 2004, Vancouver, British Columbia, Canada.
ACM 1-58113-833-4/04/0010.

Design and development of the system were outsourced but CNR kept the high level control over the entire development cycle, defining standards and methodologies.

2. Choices
2.1 Technology platform
One of the first steps was choosing the technology platform. Given the importance of the accounting system, it was crucial to build on a platform which allowed for rapid implementation of enterprise level applications. For the same reason, it was deemed advisable to adopt an open standard platform, to avoid tying CNR to a particular vendor. Besides, in order to ensure code re-usability, support to component-based development was considered a key requisite.

At that time, J2EE was the only platform meeting all the requirements listed above and, given the outcome of the project, it turned out to be the right choice.

2.2 Methodologies, documentation and tools
In order to ensure high code quality, CNR provided guidelines for the development process and defined the items to be delivered, in terms of software and documentation. In particular, the following documentation items were required: component specifications (interfaces and operations), use case diagrams, architectural diagrams (component diagrams and class diagrams), APIs in javadoc format, data dictionary, naming conventions, test case description, "use case/test case" matrix. UML was adopted as the standard notation. Finally, CNR asked the outsourcer to use specific modelling tools and code generators to speed up component development and documentation production.

3. Design, implementation and deployment
3.1 Design pattern based architecture
CIR (Contabilità Integrata per la Ricerca = Integrated Accounting for Research), the new accounting system, was designed by the outsourcer using well-known design patterns. User interaction, for instance, is managed by an implementation of the "*Model View Controller"* pattern. A single servlet, in collaboration with helper classes, acts as the *controller*. JSP pages are responsible for

rendering the user interface (the *view*). The *model* is based on two types of classes, respectively representing *behaviour* and *data*.

It is worth mentioning that, for performance reasons, no entity bean has been developed. Complex interactions with the underlying data base are managed via stored procedures.

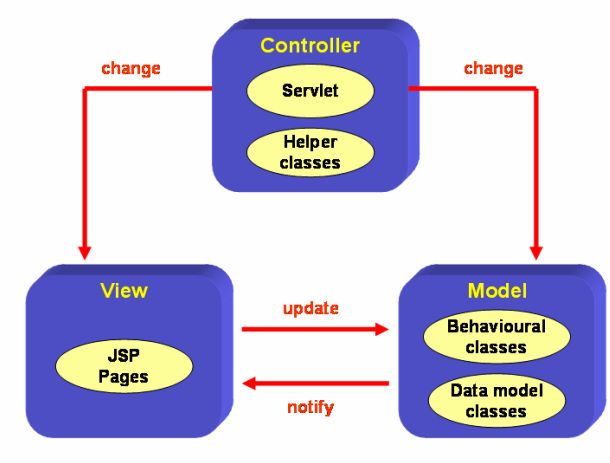

Figure 1: CIR's MVC-based architecture

3.2 Some figures

After two years of development and one year of production, CIR is now a complete, well documented and robust accounting application which supports the centralized management of CNR's financial resources. Besides, the application allows to produce reports in PDF format. The layout of those reports is defined by means of customizable templates, whereas actual data are fetched from the data base.

Here below we provide some figures that give an idea of the size of the application:

- about 500,000 lines of code
- about 10,000 FPs
- 106 EJB components
- 337 data base tables
- about 1500 users

The following languages were used to develop the application:

- Java
- javascript
- XML
- PL/SQL

It is worth mentioning that the ease of integration with other sub-systems, provided by J2EE, places CIR at the core of CNR's ERP system.

3.3 Hosting environment

The application is deployed on a couple of application servers in load-balancing configuration, whereas the data base is managed by a cluster of two machines in "standard hardware failover" configuration, both connected to a shared disk array. PDF reports are generated and stored in the DB by a couple of so called "print servers", which pull the job requests from a common queue.

4. Maintenance and future developments

4.1 Make or buy?

At the end of 2003, CNR was faced with a choice regarding the maintenance and the possible enhancements of the accounting application. There were mainly two alternatives:

- set up an internal development group, made up of CNR staff;
- outsource maintenance to a third party.

After an exhaustive analysis, *CNR chose to set up an internal maintenance group*. This was possible because of three key factors:

- CIR's architecture is based on well known design patterns, thus easily understandable;
- CIR was developed using open standard technologies;
- CIR is well documented.

An upgrade to EJB v. 1.3 has already been accomplished by the maintenance group and two major releases are planned for October and November 2004, to make the application compliant with the new CNR's regulation.

4.2 Reusing components

CIR's architecture is modular and the interest shown by other Italian Public Administrations in customizing it to their own needs indicates a significant potential for components' reuse.

5. Conclusions and future directions

The story of our accounting application shows that open standard platforms, component-based development and high quality documentation standards are rewarding choices in the long term. CIR's clean architecture and exhaustive documentation enabled CNR to undertake the maintenance with a reasonable effort and to plan for new developments. The interest shown by other Italian Public Administrations indicates that CIR's components have a significant potential for reuse.

6. For more information

Informative website:

http://www.cir.rm.cnr.it

The website is currently in Italian but we are going to set up an English version.

Contact:

roberto.puccinelli@cedrc.cnr.it

7. References

[1] Gamma, E., Helm, R., Johnson, R., Vlissides, J.. Design Patterns Elements of Reusable Object-Orientated Software. *Addison-Wesley, 1995*

8. Acknowledgements

We would like to thank Aldo Stentella for his help in implementing the graphic layout.