



Final Study Work

from 10 February 2003 to 6 June 2003 at

caritas
L U X E M B O U R G

Development of an N-tier distributed application:



Robert Fisch
DIA 4 - ISINOME

7th July 2003



1. Overview

1. Overview
2. Acknowledgments
3. Caritas
4. The Project
 - 4.1. General presentation
 - 4.2. Technological choices
 - 4.3. Procedure
 - 4.4. Architecture
 - 4.5. Dynamic Code Download
 - 4.6. Security
5. Demonstration
6. Conclusion

Final Study Work - Robert Fisch - DIA 4 - ISINOME

Next



2. Acknowledgments

- » M. Erny Gillen,
president of Caritas Luxembourg,
who gave me the possibility of doing my final study work
- » M. Robert Urbé,
representative manager of the Caritas Foundation,
who was my tutor all along this time
- » M. Steffen Rothkugel,
professor at the IST,
who was my tutor
- » Everybody else who helped me during the last few months ...

Final Study Work - Robert Fisch - DIA 4 - ISINOME

Next



3. Caritas

- » Founded in 1932
- » Activity devoted to the assistance of children, families, seniors, refugees and homeless people in the Grand Duchy of Luxembourg
- » Co-founder of Caritas Internationalis, which counts today 158 Caritas members working in 198 countries or areas all over the world
- » The Foundation Caritas Luxembourg, member of the Caritas Confederation, is particularly active in the assistance to the refugees, the urgent humanitarian aid and the development co-operation.



4.1. Project \ General presentation

- » Development of a distributed client-server application
- » Dynamic code download and integration into the client
- » Easy to maintain structure
- » User-friendly and intuitive graphical interface
- » Setup of a new database structure
- » Fast and reliable client application
- » Stable and multi-threaded server
- » Windows 2000/XP compliant client and server
- » *Low cost solution*

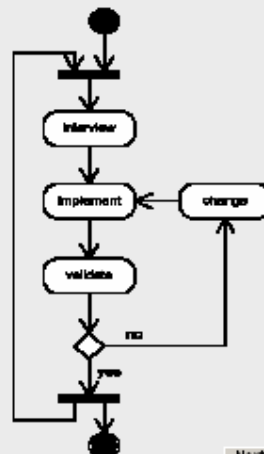


4.2. Project \ Procedure

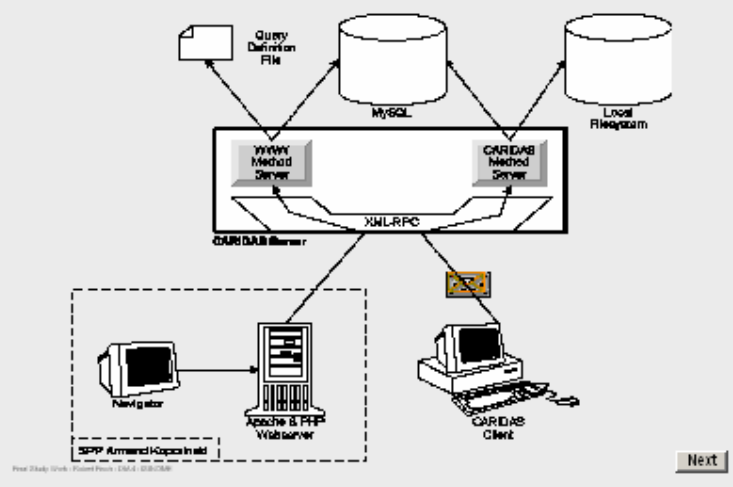
- » many different modules (sub-projects)
- => many different parallel mini-cycles

- » parallel cycles
- => freeze of one while working on another

- » one mini-cycle consists of
 - analysing the task
 - implementing a possible solution
 - letting validate this prototype
 - adjusting prototype if necessary



4.3. Project \ Architecture



4.4. Project \ Technological choices

- » Development environment: Delphi 6 Personal Edition, which is free for personal and non-commercial use, besides, all employees of Caritas are used to work with Windows
- » Database: MySQL 4.0.12, a very fast, multi-threaded, multi-user and robust database server, which is open-source software and can be accessed natively from within Delphi applications
- » Connection protocol: XML-RPC, which is an open-source, standardized and very fast protocol, for which there exist implementations in nearly every development language

Next

4.5. Project \ Dynamic code download

- » All modules, in form of DLL files, are located on the server
 - » After a user connected to the server and authenticated himself successfully, the server determines, based on the user's rights, a list of the modules the user may download
 - » The client downloads the modules and loads them dynamically
 - » The modules contain the presentation and some business logic
- => When a module is updated on the server, the changes are immediately visible to all users upon their next login!

Next



4.6. Project \ Security

- » The XML-RPC protocol does implement neither security nor sessions.
 - > SESSIONS: the server maintains a list of all connected users, thus he can store connection relevant data for each connected user
 - > SECURITY: I developped a security protocol, which is based upon the IDEA (synchronous) algorithm, a random session key (unidirectional) and a random generated piece of code.
- » The security protocol's first objective is to identify correctly the different users. It secondly aims to protect sensible data by encrypting it.



5. Conclusion

- » all objectives has been reached and the application already runs in it production environment
- » very fascinating, especially the research part for the dynamic code integration
- » I learned a lot in the following domains
 - > long term planning
 - > maintainability
 - > communication and interaction with application end-users
- » Contrary to my practical semester's experience, I can now imagine very well to work for a company



6. Demonstration

