

0.1 Practicum OOP – CORBA opdrachten

Samengesteld door Alessio Raedelli

Difficulties are assigned on a scale from 1 (easiest) to 4 (most difficult), and depends only on the difficulty of the CORBA dependent design and implementation, not on the user interfaces. In all this assignments think with care about the security side! CORBA as it is now does not offer any warranty about who's talking with a server!

* Games:

- multiplayer games
Difficulty: 1
(E.g. chess or draughts)
- client/server games
Difficulty: 1
E.g. muds, monopoly, risk

* Office:

- messaging system
 - 1) bulletin board
Difficulty: 4
The system should support:
 - * public areas (everyone can post and read).
 - * restricted areas (some group of persons can post and everyone can read).
 - * private areas (some group of persons can post and read messages).
 - * ...
 Optionals:
 - * interface with an existing e-mail system to achieve a 'mailing-list' effect.
 - 2) personal mail system
Difficulty: 3
The system should support the Message Distribution Agent (MDA) part of a mail-system. Look at procmail for an implementation which uses standard technologies.
 - * arrival of new messages
 - * retrieval of new messages
 - * forwarding of new messages
 Optional:
 - * support for the POP3 protocol
- distributed TODO list
Difficulty: 1
The system should support:
 - * a public TODO area.
 - * a private TODO area for each person (think about security, and how to implement it).
 - * adding an item to the public area.
 - * assigning an item to a private area (optional: restrict this operation to some users)
 - * items priorities

- Optionals:
 - * ...
- distributed appointments list
 - Difficulty: 2
 - The system should support:
 - * a calender viewer
 - * insert appointments in the calender
 - * warn the user of an appointment
 - Optionals:
 - * view and instert appointments in someone else
(think about security and how to implement it)
 - * interface to an e-mail system as early-warning system
 - For a working exaple have a look at SunSoft's dtcm (to be found in /usr/dt/bin/dtcm), or MicroSoft's Schedule+
- conferencing tools
 - Difficulty: 2
 - Think about chatting tools (something like the Internet Relay Chat), mbome alike tools.
- distributed drawboard
 - Difficulty: 1
- * Software Development
 - bug tracking
 - Difficulty: 4
 - The system should support:
 - * submitting a new bug
 - * assign a bug to an programmer
 - * assign a priority to a bug
 - * add some not to a bug-submission
 - * add a solution to a bug-submission
 - Optional:
 - * notify some user-group of any modification to the submission.
 - For working examples of commercial products see:
 - * PureSoftware's PureDDTS
 - * GNU gnats
 - distributing editing
 - Difficulty: 3
 - Write a source management system, possibly based on top of some revision control system (RCS, SCCS). For a working system look at SunSoft's workshop (/opt/SUNWspro/bin/workshop)
- * System Management
 - system's failure track system
 - Difficulty: 3
 - The system should support:
 - * early warn of a full filesystem
 - * early warn of full swapspace
 - * centralized administration
 - * multiple host support

