Arquitectura de Sistemas de SOftware

11th May 2009 • Personal notes allowed • Duration: 75 Minutes

> Read carefully the description of the software system below considering it as a concrete of application and then answer the questions that follow, **always justifying them succinctly and clearly**.
>
> In the answers, when useful, you may (and should) mention the bibliography or references that fundament the answers. When required, you should make explicit all the assumptions you did to answer the questions.

SIMPLEGARRA

Organizations often have very complex structures. They are usually made up of suborganizations, sometimes called departments, divisions, services, etc. For example, FEUP is part of U. Porto, which contains other schools (FBAUP, FLUP, etc). FEUP includes also other units, such as DEI, CICA, Biblioteca, etc.

An important fact of successful organizations is that they are alive, and that change very fast, in order to accommodate new challenges and support new services and processes.

To properly support these complex structures and their evolution, software systems must be very flexible. SimpleGarra is a (fictionary) branch of SiGARRA software system that is aimed at simplifying some of the processes involved at academic organizations.

Document management is just a type of service that must be very flexible to ensure its adequacy. Enrollments, course management, accounting, research activities, are some more examples.

After a detailed requirements analysis, we found the following as the more relevant in terms of architecture:

1. The system must run in a web browser.

2. The system must support the idea of an university, which may contain several schools and other units (institutes, companies, etc), eventually organized in other organizations.

3. The system must be able to manage several organizations at the same time.

4. An organization has a specific official image (logo, colors, description).

5. Each organization must have its own repository of information, ideally a database.

6. All organizations may share information with others, providing specific access-rights at the level of organization, unit/subunit, and person.

7. Organizations run business processes, which must be very flexible to edit and change using a model.

8. Organizations may share processes, too.

9. Processes may need to trigger notifications to persons, groups of persons, and responsible of organizations/units/services.

QUESTION 1: ARCHITECTURAL STYLES   35%

Consider the architectural styles studied. Answer and justify the following questions:

a. Which key architectural problems do you find as more relevant in this case?

b. Which style you consider to be the most appropriate to adopt as the **main style** for the SimpleGARRA? Characterize that style in terms of components, connectors and constraints, and instantiate them to the case. Relate them to the architectural problems above.

c. Which other styles you consider to be interesting to consider, in a complementary way? Justify with concrete situations of the case. Relate them to the architectural problems above.

QUESTION 2: DESIGN PATTERNS   65%

Considering the design patterns you studied, suggest and justify which you think are more appropriate to solve each one of the design problems below:

a. An organization may aggregate units, services and persons. Some organizations are part of others, but some have their own identity. How to represent these kinds of organizations to enable them to be considered in a similar way?

b. New units, services and processes can be dynamically added on the fly by simple configuration of the organization.

c. If the organization web site is offline, it must work in the intranet, at least, and synchronized only when online again.

d. Notifications to people must be propagated along a series of persons, services, units, and eventually organizations.

e. Processes are defined in a special natural language (called "Processês") that must be used to execute the processes and their activities at runtime.

f. Some people need to delegate some tasks to others, which may only have partial rights, not all.

The End.

QUESTION 1: ARCHITECTURAL STYLES   **35%**

Consider the architectural styles studied. Answer and justify the following questions:

a. Which key architectural problems do you find as more relevant in this case?

    Persistency, distribution, event notification, control-access, sharing, flexibility of processes.

b. Which style you consider to be the most appropriate to adopt as the **main style** for the SimpleGARRA? Characterize that style in terms of components, connectors and constraints, and instantiate them to the case. Relate them to the architectural problems above.

    Repository/Database. Components = tables, data. Connectors = relationships.

c. Which other styles you consider to be interesting to consider, in a complementary way? Justify with concrete situations of the case. Relate them to the architectural problems above.

    Layers, Blackboard, independent-processes, interpreter, object.-oriented.

QUESTION 2: DESIGN PATTERNS   **65%**

Considering the design patterns you studied, suggest and justify which you think are more appropriate to solve each one of the design problems below:

a. An organization may aggregate units, services and persons. Some organizations are part of others, but some have their own identity. How to represent these kinds of organizations to enable them to be considered in a similar way?

    Composite

b. New units, services and processes can be dynamically added on the fly by simple configuration of the organization.

    Interpreter+Template? Reflection? Factory Method, Builder

c. If the organization web site is offline, it must work in the intranet, at least, and synchronized only when online again.

    Proxy+Memento, Adapters, Observer,

d. Notifications to people must be propagated along a series of persons, services, units, and eventually organizations.

    Chain-of-responsility+publish-subscribe, observer.

e. Processes are defined in a special natural language (called "Processês") that must be used to execute the processes and their activities at runtime.

    Interpreter+Visitor.

f. Some people need to delegate some tasks to others, which may only have partial rights, not all.

    Proxy, chain of responsibility, command, decorator.