Your African solutions in databases, mobile and web applications

BenSino Information Technology Corporation

http://it.bensino-eg.com

EMS Case Study

Design Artifacts

Ralph D'Almeida

2013

Table of Contents

Overview

Comergence, the solidarity cooperative is a network of organizations and connivance of people who respond to the recognized need for an organization to coordinate and integrate the activities of existing organizations within African Immigrants Communities (CIIA in French).

I. Business needs

Each organization has its own event management application and avoiding redundant and conflicting events occurrence among all organizations' events necessitates tremendous efforts to cross-reference the events.

Live global statistical reports on many events created by different organizations cannot be gathered and analysed.

II. Solutions

EMS allows centralizing all organizations' events into one events management system accessible to each organization concurrently.

Any organization can login EMS to collect statistical data on events organized by another organization.

III. Benefits

Having one events management system for all organizations reduces conflicting events, improves their scheduling, and allows attendance monitoring.

Collecting statistical data on all events and attendees increase managers' performance regarding available capacity fulfillment and clients' fidelity discount.

Case Study

This section describes the software engineering solutions used to satisfy the above business needs. In other words, the design artifacts of EMS are highlighted here.

I. Review

Events Management System (EMS) is an enterprise web application for efficiently managing every organization's events in order to share attendance data among those organizations. Its goal is to provide useful data on events' attendance to managers while reducing the number of conflicting events. Basically, any organization can manage only their own events and associate clients to them, but the system will prevent creating two events with the same starting time unless explicitly specified otherwise. In addition only managers can perform statistical reports.

II. Design artifacts

All the artifacts used to engineer EMS are presented in this section from the requirements specifications to the implementation of the system.

A. Requirements

The solutions for centralizing all the organizations' events are detailed here in form of high-level technical requirements.

1. Stakeholders Summary

Name	Description	Responsibilities
Managers	*Organization's team and project managers*	*The manager performs statistical reports in order to ensure seats availability.*
Operators	*Staff members and receptionist*	*The operator registers clients and manages events.*
Clients	*Organization's members and events' attendees*	*The client attends an event.*

2. Needs and Features

Need	Priority	Features	Planned Release
The system should allow the managers to view events statistical data.	*High*	*View the average number of attendees per event*	*Version 1.0.0*
	High	*View event fulfillment percentage*	*Version 1.0.0*
	High	*Cancel events*	*Version 1.0.0*
The system should allow the operators to manage events	*High*	*Create events*	*Version 1.0.0*
	High	*Modify events*	*Version 1.0.0*
	High	*Search for a specific event*	*Version 1.0.0*
The system should allow the operators to register clients.	*High*	*Record clients' information*	*Version 1.0.0*
	High	*Associate clients with events*	*Version 1.0.0*
The system should allow the clients to view events' information	*Normal*	*View all events*	*Version 1.0.0*
	Normal	*View event's seats availability*	*Version 1.0.0*
	Normal	*View event's attendees*	*Version 1.0.0*

3. Use Case Model

a. Actors

Name	Description	Goals
Administrator	*The administrator (managers) enforces the system's policies and performs statistical reports*	*The administrator wants to cancel an event and analyze statistical data.*
Operators	*The operators can manage events*	*The operators want to create and update events, and associate clients to them.*
Clients	*The clients can only view events' information*	*The clients want to know which event to attend and if there are available seats for the event.*

b. Context Diagram

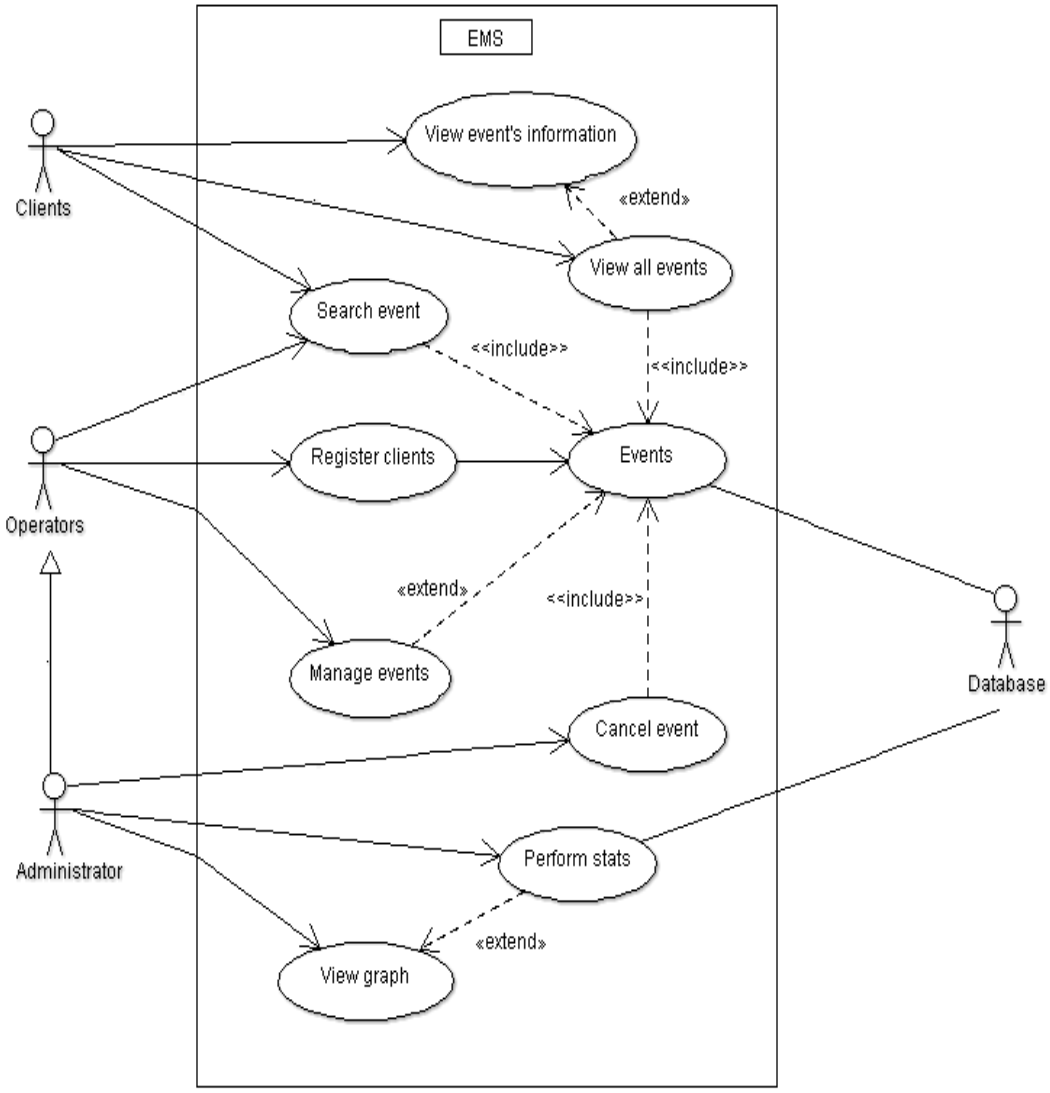


Figure 1: Use Case Diagram

c. Domain Model

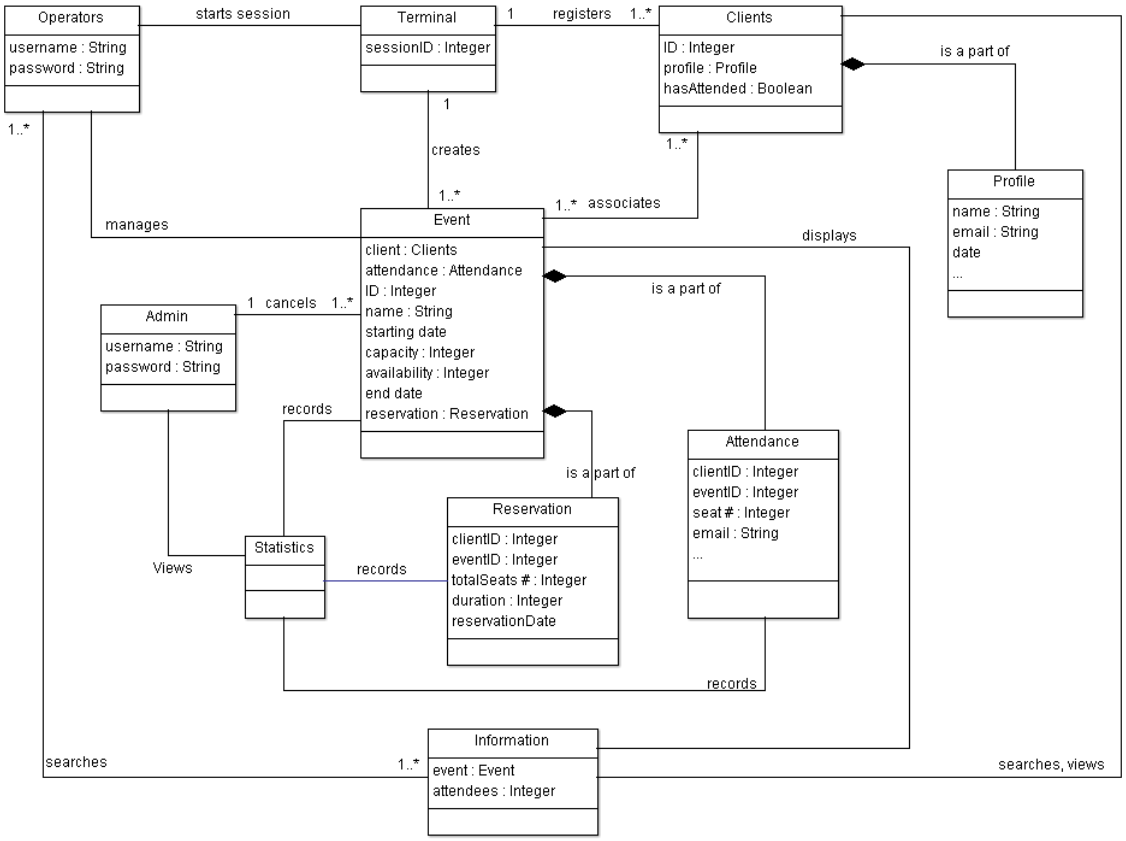


Figure 2: Domain Model Diagram

4. Supplementary Specifications

This section describes the non-functional and functional requirements not captured in the use case model. The quality, regulation, and standard adopted by the system are presented below.

a. Functionality

- There are Operators and Admin access level to the system. Operators can create and manage events as well as register clients and associate them to each event. Admin has the same access as Operators plus the ability to cancel events and perform statistical reports. Any staff member is considered as Operators while Managers are Admin. Clients are also Operators with the differences that they cannot manage events and can only view events' information.

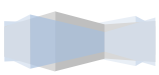
- A user is able to log out from the system at any time. Additionally, the user must be logged out when the application is closed.
- The system must be able to automatically update the events' attendance at a regular time basis.

b. Reliability

- The system will be available 24/7 unless it experiences downtime due to technical difficulties or for maintenance. Unforeseen downtimes should not exceed 1 day. Scheduled maintenance should not exceed 6 hours and should happen during overnight only.
- The Mean Time Between Failures (MTBF) is the sum of the availability periods divided by the number of observed failures. Assuming there are only 2 observed failures per month, the expected time will be 14 days.
- In case of a fault occurring in the system, the Mean Time To Repair (MTTR) or the longest down time the system will experience until repair will not exceed 24 hours.
- The system must always deliver reliable and accurate data.
- In term of maximum bugs or defect rate, the system defective rate will not be more than 2% of the KLOC. Major bugs will be fixed as soon as possible, significant ones in 3 days and minor ones in at most 24 hours. The maximum bugs allowed are 10 bugs/KLOC and a critical bug is considered as loss of data.

c. Performance

- The response time of the system when viewing statistical data should not exceed two seconds.
- The system will be able to handle a minimum of 1000 users logged in at a given time.
- The system shall provide access to the database with no more than 3 seconds latency.
- The system shall perform actions and have a user command response time of less than 2 seconds. It will display a notification detailing any error occurred and/or the action to be taken by the user. The length of time actions are completed is dependent on the complexity of the user's inputs.



d. Design Constraints

- The system will use the libraries GWT and SmartGWT, Java EE technology, Spring and Hibernate frameworks, Java programming language, SQL script, and datasource XML file.
- The system requires the following software tools: Eclipse Juno (4.2): Integrated Development Environment for Java EE, Apache Tomcat 7.0.35: Web Application server, and MySQL Community Server 5.5.29.
- The system will use a 3-tier architecture model where the front-end will integrate the Model View Controller architecture. The client's browser must allow the execution of JavaScript file.

B. Architecture

EMS is a distributed software that is designed by partitioning the stakeholders' concerns in order to avoid some of its integration problems. The RM-ODP (Reference Model for Open Distributed Processing) views model is chosen to fully satisfy each concern since it provides a set of viewpoints already proven useful in separating integration concerns. These viewpoints are described below:

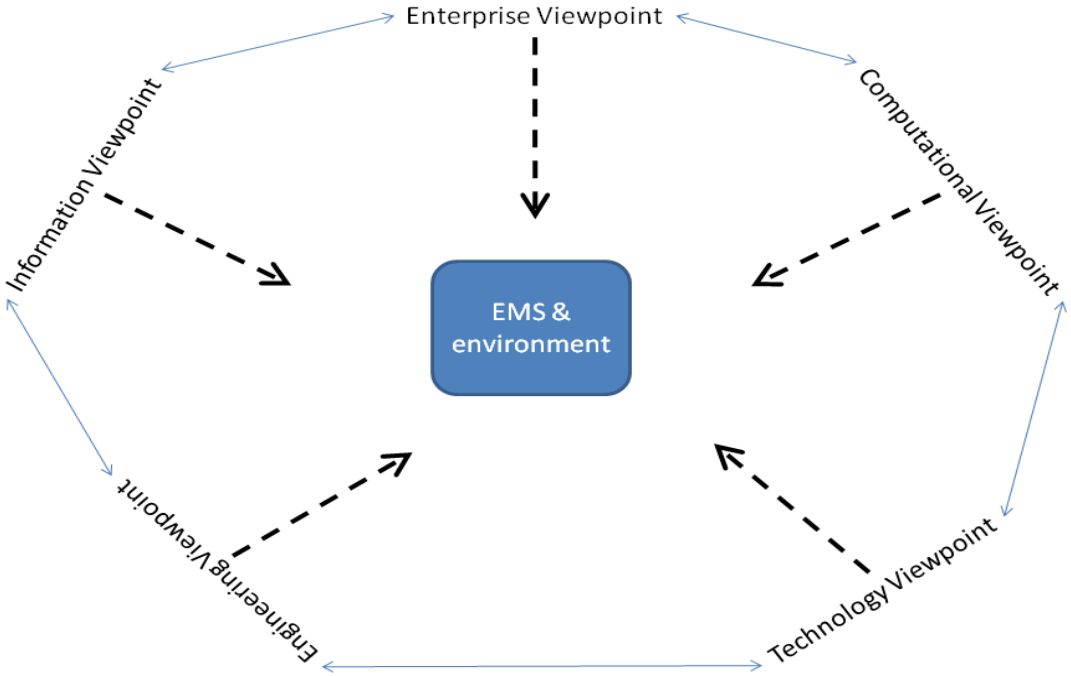


Figure 3: RM-ODP Architectural View

1. Enterprise Viewpoint

It describes the business requirements and how to meet them, but without having to worry about other system considerations, such as software architecture, computational processes, or the technology used to implement the system. It then represents the business aspects of the system and provides the basis for checking conformance of EMS implementation. It is concerned with the purpose, scope and policies governing the activities of EMS within Comergence. The purpose is defined by the specified behaviour of the system and the policies capture further restrictions of that behaviour. The system is modelled by many enterprise objects that model its environment, and by the roles in which these objects are involved.

a. Community

It is specified in a contract and is a configuration of enterprise objects, formed to meet an objective. The community is modeled here as an organizational unit. For instance, each organization that constitutes Comergence is represented as one community. The interactions between communities are achieved through interactions between enterprise objects.

b. Scope

It is the behavior (collection of actions with constraints on their occurrence) that EMS is expected to exhibit. The scope of Comergence community is comprised of:

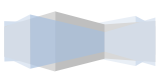
- To coordinate other organizations
- To produce reports

c. Objective

It captures what the EMS community is for. An example is the execution of each organization's activities as a sub-community of EMS community.

d. Enterprise Object

It is a member of one or more communities that models an entity (actor roles). It may also participate in actions by filling artefact roles. Sometimes, it is useful to model the states of the enterprise objects because they may help establishing the correspondence with information object (Information Viewpoint). The following are considered as enterprise objects:



- Staff
- Members
- Clients
- Seats
- Reservation
- Organization unit system administrator

e. Role

It identifies a specific behaviour of an enterprise object in a community. We have:

- Inclients
- Outclients
- Receptionist
- Manager of organization's department
- Organization unit administration system

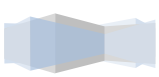
The above examples are based on type of interactions expected in EMS community.

f. Policy

It is a set of rules related to a particular purpose and is used where the desired behaviour of EMS changed to meet particular circumstances. As an example, "Seat Reservation Policies" include reservation duration policy and number of seats reserved policy. Specifically, there are prescribed limits on the number of seats that can be reserved for an event mentioned in number of seats reserved policy. Members may reserve a maximum of 10% of the total number of seats per event, while clients can only reserve 5% of the total number of seats.

g. Process

It is a collection of steps taking place in a prescribed manner and leading to the accomplishment of some results. It models sequences of actions, carried out by one or more enterprise objects and is expressed here in form of activity diagrams. For instance, the enterprise object "Clients" carries out the sequence of actions "Go to reception", "Validate information", "Select events", "Pay bill" modeled as processes within the activity diagram.



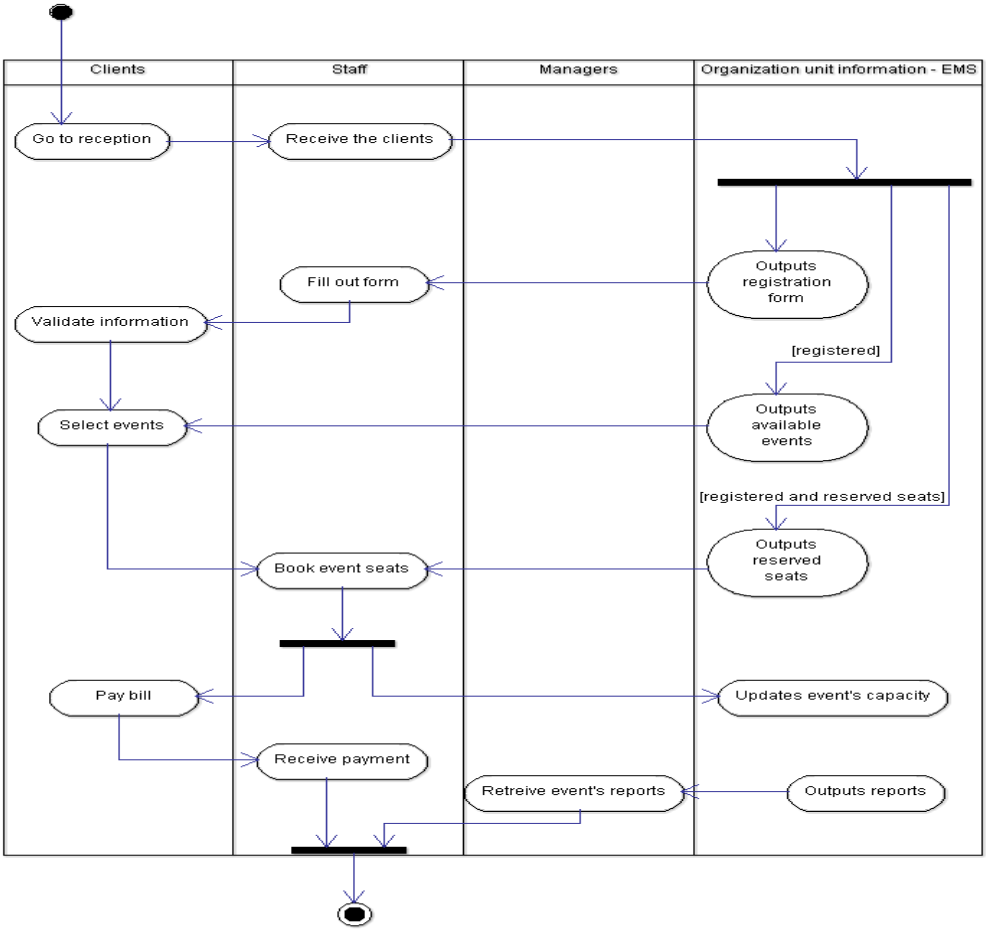


Figure 4: Enterprise viewpoint process activity diagram

2. Information Viewpoint

Representing the information system aspects of EMS, it is concerned with the type of information handled by the system and constraints on the use and interpretation of that information. The information specification of EMS is expressed by one model that contains a set of packages expressing the invariant, dynamic, and static schemata of the system. From the objects and roles identified in the enterprise specification, the following information objects types: “Buyers”, “Seats”, and “Members” describe the information stored and handled by EMS about them. Moreover, “Calendar” and “Reservation” objects respectively model the passage of time and the relationships between “Buyers” and “Seats”. The object types of the information viewpoint specification are described here in form of class diagram. The attributes of each class define the information captured by this viewpoint specification.

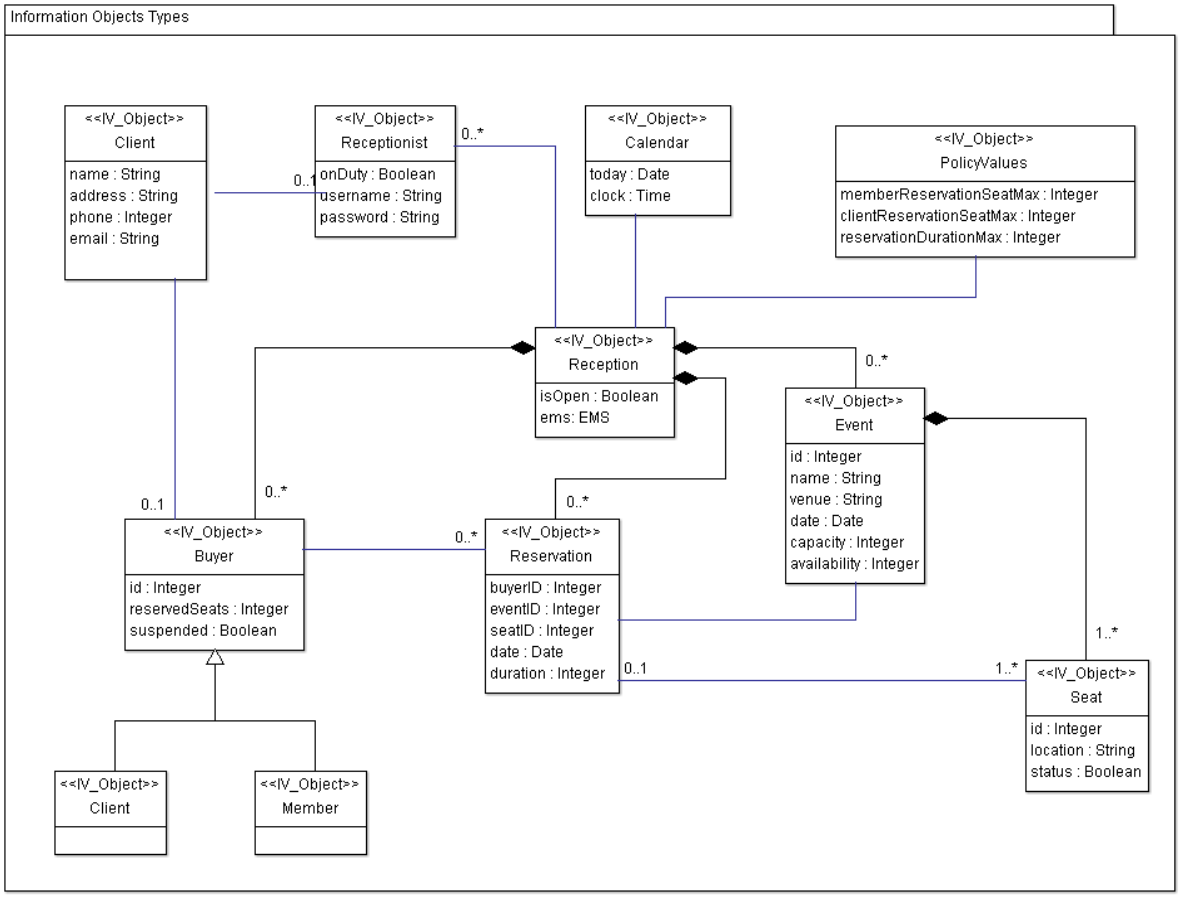
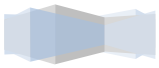


Figure 5: Information viewpoint (Object types) class diagram

The action types of the information viewpoint specification are derived from the processes and interactions defined in the enterprise viewpoint specification. They are described using a package that expresses the invariant schema specifying the action types supported by the information objects of EMS. That is, they are represented by signals which will trigger the state changes in the state machines of the information objects.

The correspondences between the Enterprise and Information specifications are expressed in a package that contains a top-level correspondence which links these two types of objects. This correspondence can be broken down into a set of correspondences which establish particular details of it.



3. Computational Viewpoint

EMS design aspects are highlighted here. It is concerned with the functional decomposition of the system into a set of objects that interact at interfaces – enabling system distribution. A computational specification defines units of function as computational objects and the interactions among those computational objects without considering their distribution over networks and nodes. These computational objects and interactions provide the software architecture of EMS. The functionality specified in the enterprise and information viewpoints identifies the basic operations that are grouped into interfaces. The viewpoint is described here in form of component diagram. Each computational object is expressed as a component. Object interfaces are expressed as component ports and interface signatures are expressed as interfaces.

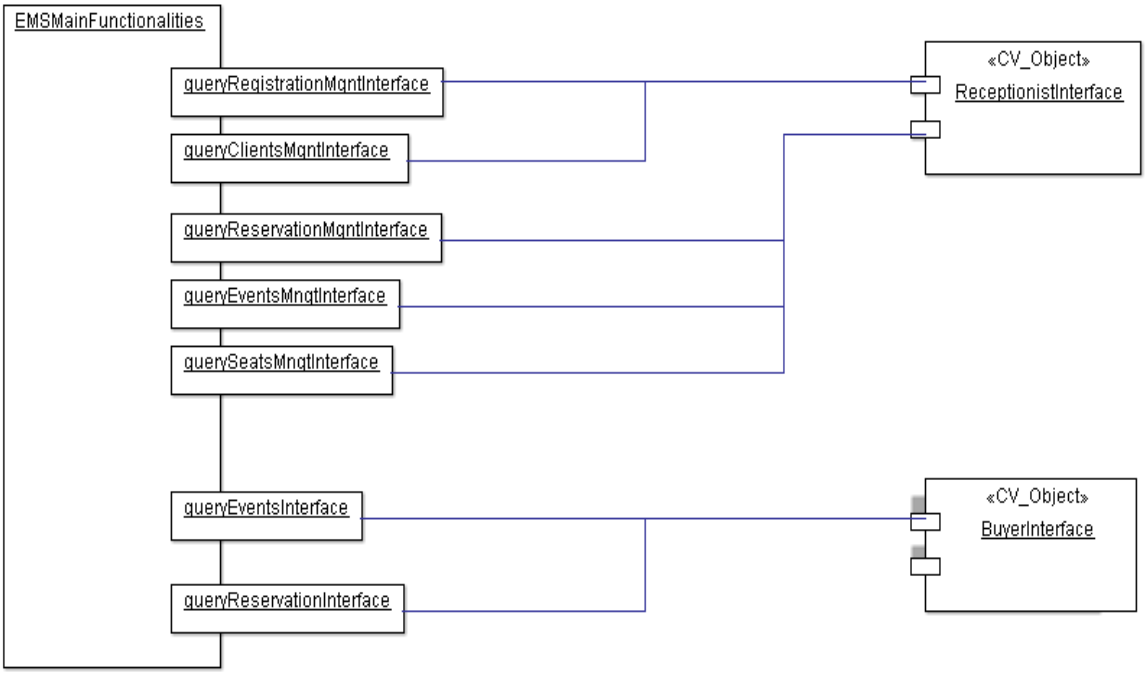


Figure 6: Computational viewpoint – EMS component diagram

The behaviour of the elements of a computational specification also needs to be specified. State machines can be used to express the internal behaviour of computational elements, but interaction diagrams are more appropriate when messages and interaction protocols are the focus of design. Hence, the sequence diagram below shows the interactions that occur between the components of the computational specification during the reservation process.

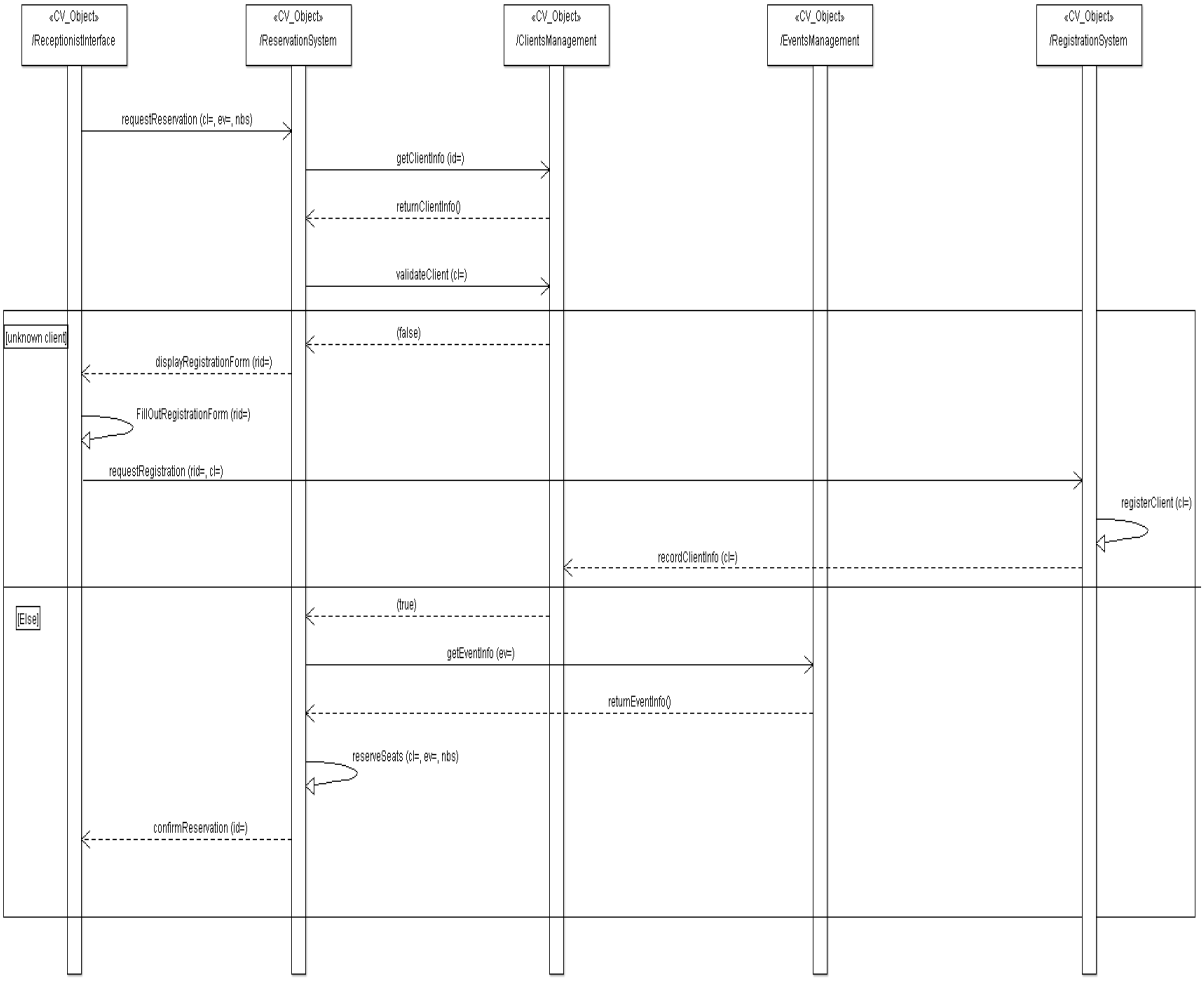


Figure 7: Computational viewpoint - interaction diagram for reservation process

The correspondences between the Enterprise and Computational as well as between the Information and Computational specifications are expressed in packages that consistently link these types of objects in pairs.

4. Engineering Viewpoint

Also known as the solution types and distribution, it is concerned with the infrastructure required to support EMS distribution. Its specification defines the structure of node (nucleus, capsule, cluster, engineering objects), channel, and their management functions. The computational objects defined earlier are supported by corresponding basic engineering objects, which are deployed within clusters on nodes, and by the

engineering infrastructure using nodes, nucleus, capsules, capsule managers, clusters, cluster managers, and channels. Here, 3-tier and MVC architectural style are used to define a physical deployment model.

The node configuration is composed of Presentation_Tier, Business_Tier, and Data_Tier. A request from Presentation_Tier node is sent to a server node, which serves as the Interaction_Server and Business_Server that also handles functional requests. Finally, data persistence is taken care of by another server node, which serves as the Data_Server.

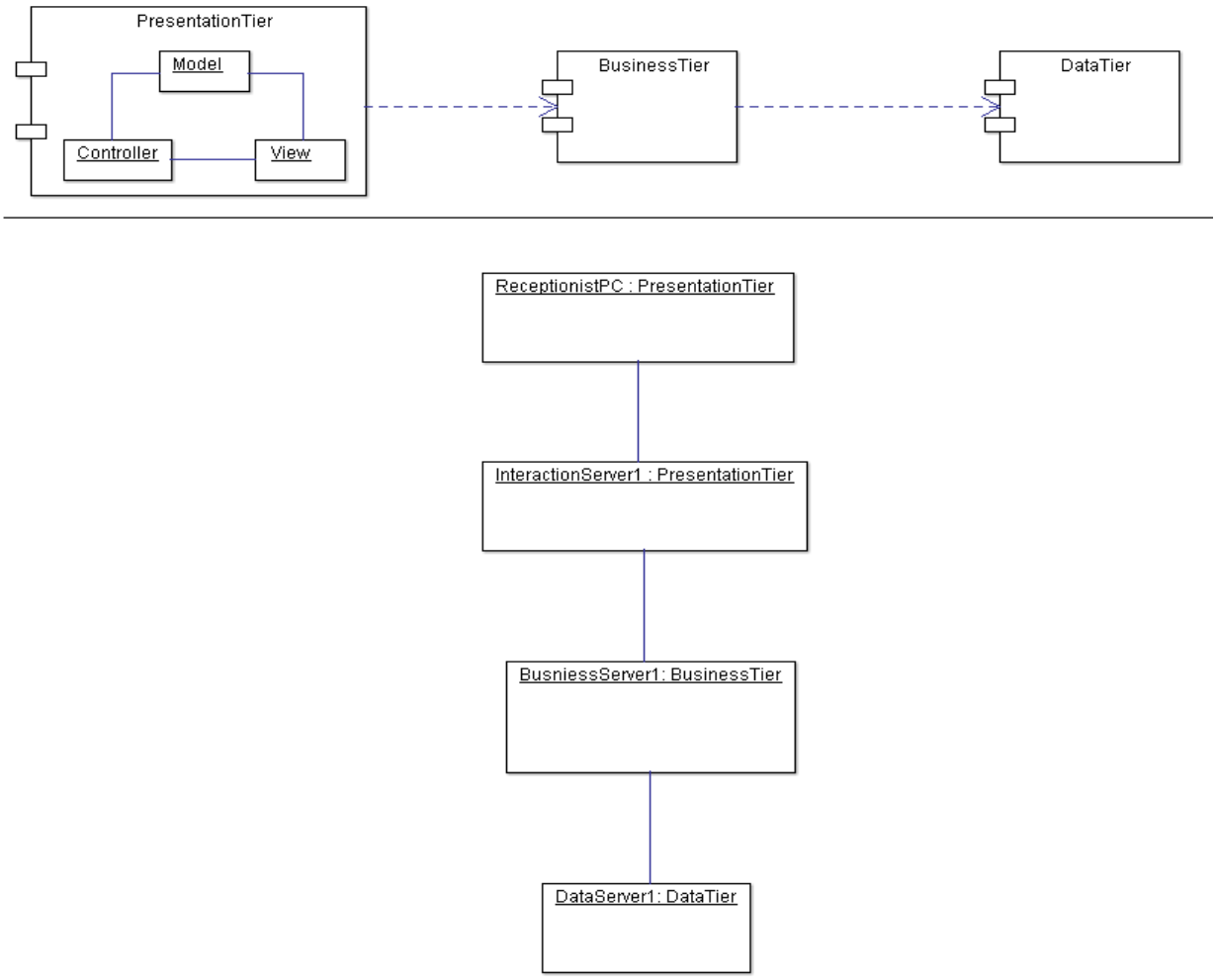


Figure 8: Engineering viewpoint - EMS node configuration

The structure of each node is composed of the node itself, nucleus, capsules, capsule manager, clusters, cluster manager, basic engineering objects (BEOs),

engineering objects, stubs, binders, protocol objects, and interceptors. In this example, BEOs are deployed as follows:

- BEOs for graphical user interface to access the system are deployed on the ReceptionistPC,
- BEOs to support MVC and n-tier architectural style are deployed on the Interaction_Server and the Business_Server,
- BEOs for application specific computational objects are deployed on the Business_Server and the Data_Server.

In general, BEOs are deployed on various nodes and within clusters in capsules. They usual use channels to interact with each other.

The channels used here are as follow: one between ReceptionistPC and Interaction_Server, one between Interaction_Server and Business_Server, and one between Business_Server and Data_Server. The first channel for instance comprises a stub, a binder, and a protocol object for the ReceptionistPC, and a stub, a binder, and a protocol object for the Interaction_Server.

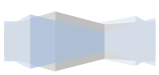
The functions are expressed by activities and actions. They are represented in terms of interfaces that specify their signatures, and the engineering objects that provide them.

The correspondences between the Enterprise and Engineering as well as between the Computational and Engineering specifications are expressed in packages that consistently link them.

5. Technology Viewpoint

It describes the implementation of EMS and is concerned with the choice of technology to support its distribution. Its specification defines technology objects (hardware, software, and network products) that implement standards as its templates, implementation as a process of instantiation, and IXIT as implementation extra information for testing. The viewpoint is described here in form of deployment diagram.

The node configuration upper diagram on one hand shows the different technology objects types that will be used in EMS deployment architecture and how they can be connected among themselves. The lower diagram on the other hand describes the actual system.



The node structure contains technology objects that implement the corresponding engineering objects.

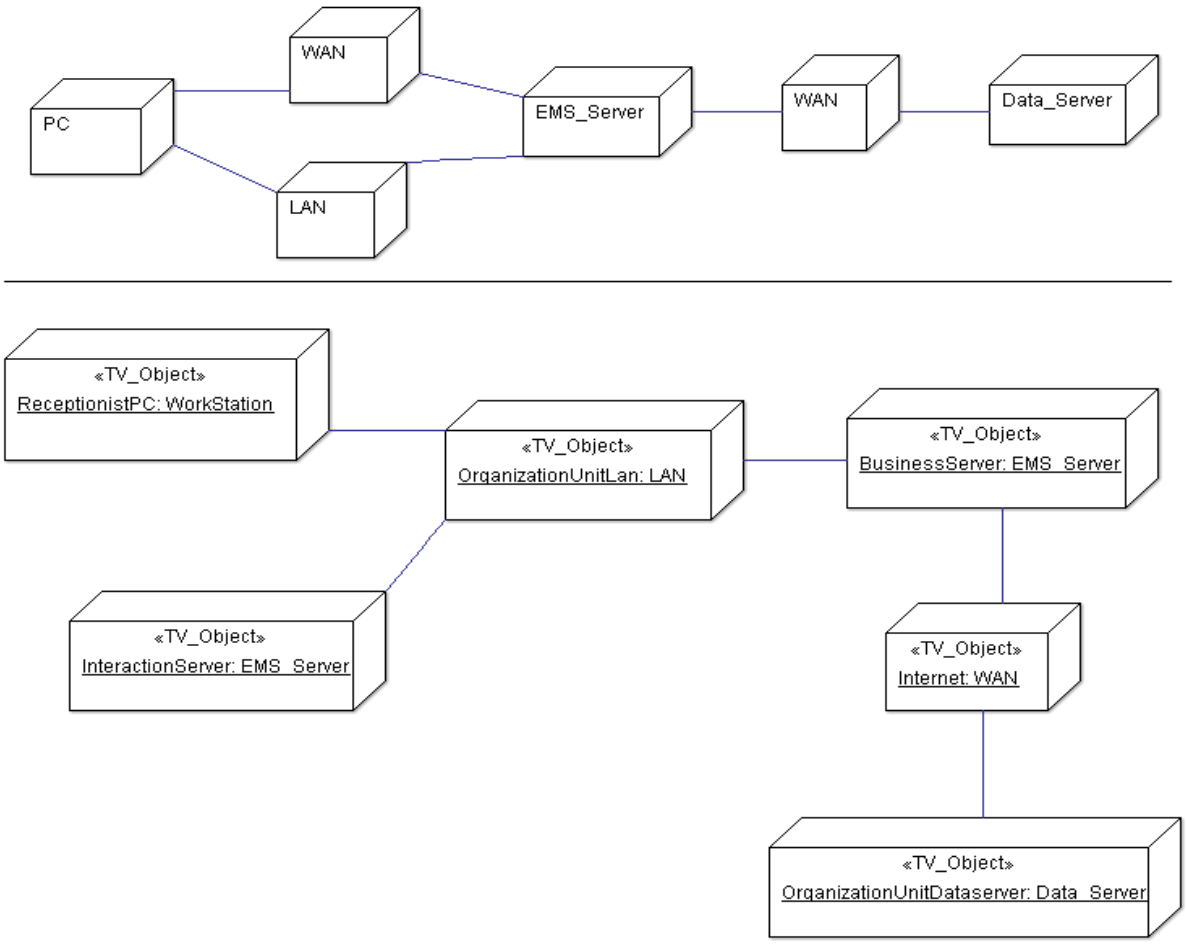


Figure 9: Technology Viewpoint - EMS node configuration

IXIT means that additional information for specification or standard are defined when performing a test against EMS implementation. In this respect, IXIT can be attached to any technology objects for interaction with a user, other technology objects in the same node, and other technology objects in other nodes.

The correspondences between the Engineering and Technology specifications are expressed in a package that contains a top-level correspondence which links these two types of objects.

6. Additional Viewpoint (data view)

This view is described in form of er-diagram and represents the inputs and outputs of the data in EMS. It links the five viewpoints depicted above in the sense that they

define the need for the data, the elements of the data, the elements of the system that manipulate the data, the behavior of these elements, how the data is distributed, and how the data is managed.

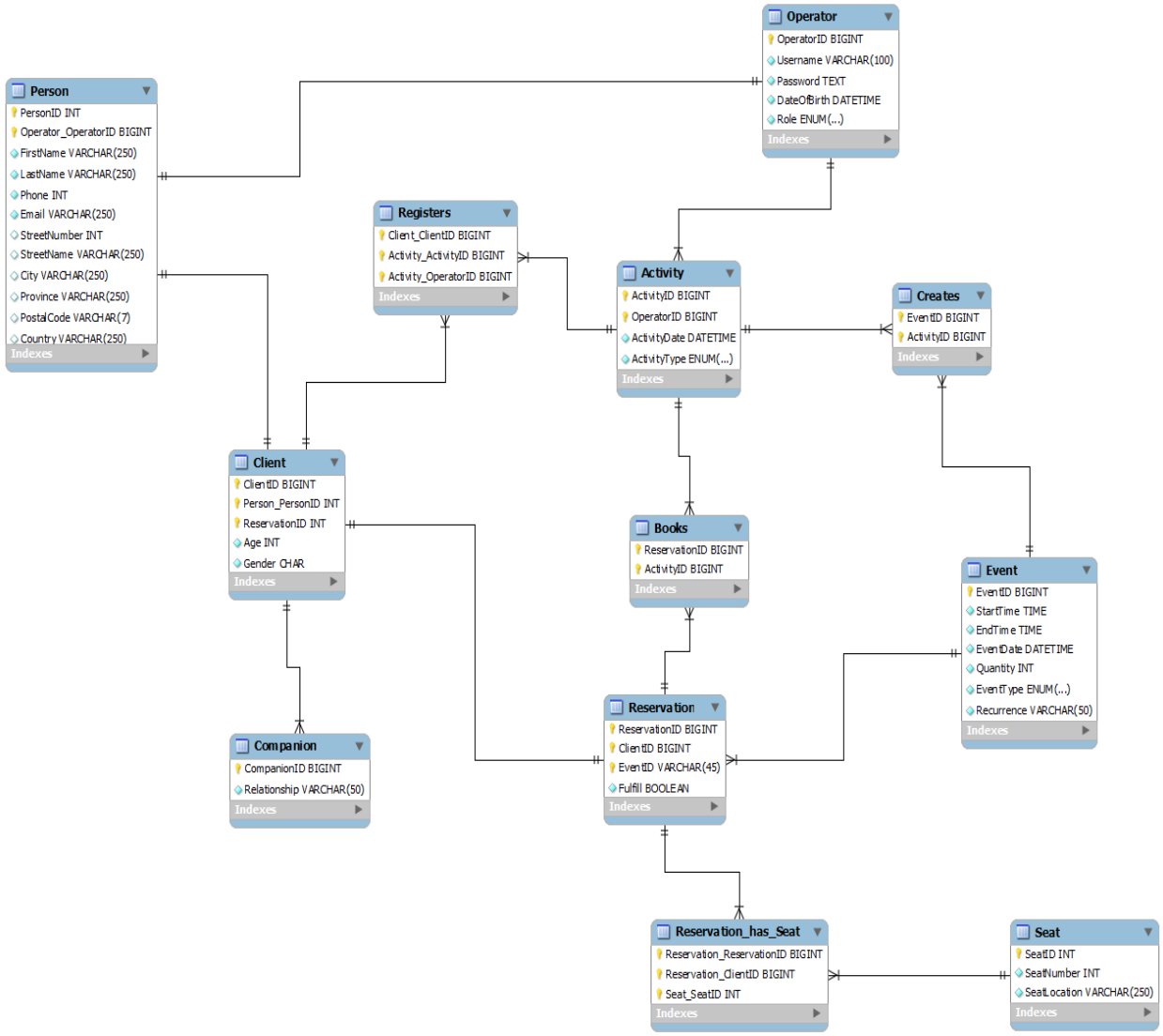


Figure 10: First draft ER-Diagram

C. Testing

This section describes the test plan adopted to ensure that EMS performs as expected in its execution environment.

1. Unit Testing

In progress

2. Integration Testing

In progress

3. System Testing

In progress

D. Implementation

In progress

III. Quality assurance

In progress

Conclusion

In progress