
e-Learning for Physically Challenged

Zia Saquib

C-DAC, Pune

Abstract

The e-Learning research Programme at ACTS, C-DAC Pune is conceived with a vision to transform the internet into a powerful environment for teaching online, especially to persons with visual impairment. The e-Learning system will be used for launching almost all the postgraduate courses offered by C-DAC, ACTS Online. This system will make it easy to publish documents, lectures and exercises for faculty members and lab faculty members. They will be easily accessible to persons with physical disabilities through a text-to-speech engine. It is intended to be the main portal for all the students to get the most up-to-date information for their lectures and exercises.

Evidyapeeth is C-DAC's framework based on a component-oriented architecture, which divides the whole system into two parts viz. foundation components and extension components. The Foundation components represent the e-Learning domain oriented components namely administrator, faculty, assistant faculty student, and enrollment. The extension components represent some of the domain neutral functionalities namely discussion, chat, lecture, and whiteboard. This paper discusses the design and development of an e-Learning System based on eVidyapeeth.

1. Introduction

Evidyapeeth is C-DAC's e-Learning Framework evolving into an indigenous "Learning Management System" and e-Learning infrastructure product comprising of the following features:-

Features of this system :

1. Student Registration & Enrollment
2. SCORM based Course Delivery
3. On-Demand Examination
4. Threaded Discussion
5. Virtual Classroom and Voice-over IP
6. Shared White Board
7. Institute Administration

- 8. Faculty Operations
- 9. Text-to-Speech engine

2. Architecture of eVidyapeeth

eVidyapeeth is built on multi tier architecture. It is divided in Presentation Layer, Business Layer and Data Layer. **(Figure 1)**

Multi Tier Architecture

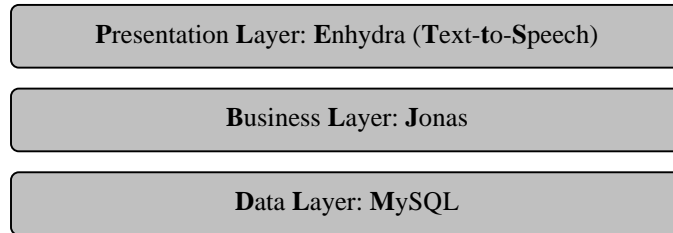


Figure 1. eVidyapeeth- A Multi Tier Architecture

The data layer of eVidyapeeth tries to keep the database simple. This layer is only intended to deal with the storage and retrieval of information. It doesn't care about how you plan on manipulating or delivering this data. Data layer is implemented using MySQL database.

The presentation layer provides an interface to the end user in the application. That is, it works with the results/output of the Business layer to handle the transformation into something usable and readable by the end user. This layer is implemented using Enhydra web server. The business layer is basically where the brains of

the application reside; it contains things like the business rules, data manipulation, etc. This layer does not know anything about HTML, nor does it output it. It does not care about SQL, and it shouldn't have any code to access the database or the like. Those tasks are assigned to each corresponding layer above or below it. This layer is implemented using JoNAS EJB server.

Figure 2 illustrates the components of eVidyapeeth. **Figure 3** gives the detailed architecture of e-Vidhyapeeth

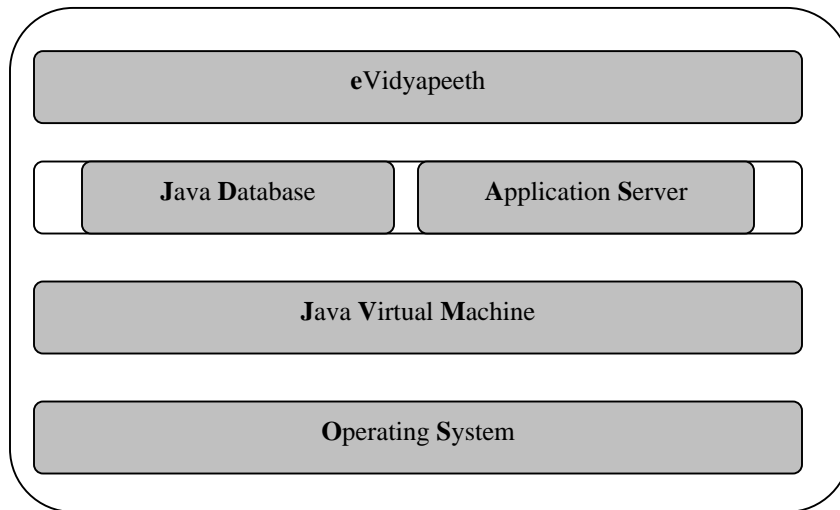


Figure 2. Components of eVidyapeeth

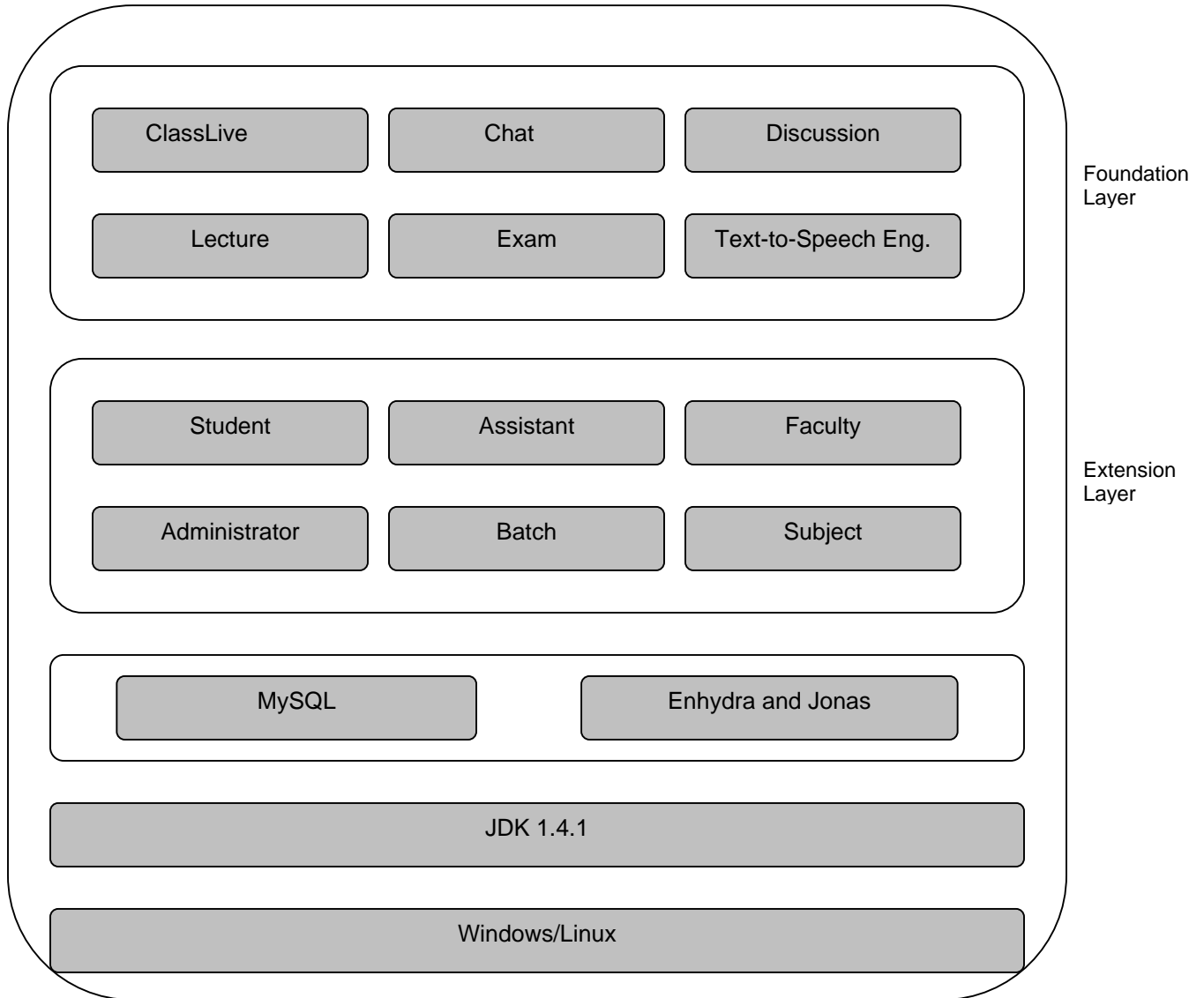


Figure 3. Detailed architecture for eVidyapeeth

2.1 Foundation Component Layer

Student component covers all properties and activities of a student at the portal. This component holds personal details of the student like name, e-mail address, registration no etc. It supports the activity of "apply for a login at the portal" for a student.

Assistant component actually covers instructors, professors and assistants. The role of these actors will be summarized as assistant component. It holds details like name and title. Activity like "manage

students' login" should be implemented in this component.

Administrator component administrates the whole eVidyapeeth. The administrator can build a new university, faculty, course or institute. Besides it can give an access to the certain instructors, professors and assistants for the first time.

Faculty component takes care all the things about universities, faculties, courses or institutes.

Batch component gives every faculty a time zone. This could be half a year or one

year and so on. This can be defined at the development time.

Subject component covers the subject within the course at the given batch.

Text-to-Speech engine translates the text into speech for visually impaired persons.

2.2 Extension Component Layer

These components can be installed into the foundation layer. All the functionalities of eVidyapeeth are put in this plug-able layer. It is important to know that we can always extend the components in this layer. Following are the implemented components:

Lecture component gives an opportunity for assistants to publish their lecture materials. Also students that already registered can download them easily.

Chat component makes it possible to chat with instructors, professors, assistants and other students depending on the subject.

ClassLive component supports connections from an unlimited number of users, and can be used for both text conferencing and 'whiteboard' drawing.

On Demand Online Examination component allows the students to take the examinations online and record the scores.

Mailing list component enables the students to apply for a mailing list so he or she will get the current information in their e-mail box.

Discussion component functions like chat component but in asynchronous way.

3. Relationships and Dependencies between the Foundation Components

Foundation Components

As we can see in the architecture specification of eVidyapeeth, the system provides some foundation components such as Student, Assistant, Faculty, Administrator, Semester, Subject, etc. that should be used to handle common services at a faculty or a university. **Figure 4** shows

the relationships and dependencies between these foundation components. The Enrollment component will be used mostly to communicate with the Extension Components of the eVidyapeeth system. An enrollment is a combination between a faculty, a subject and a semester.

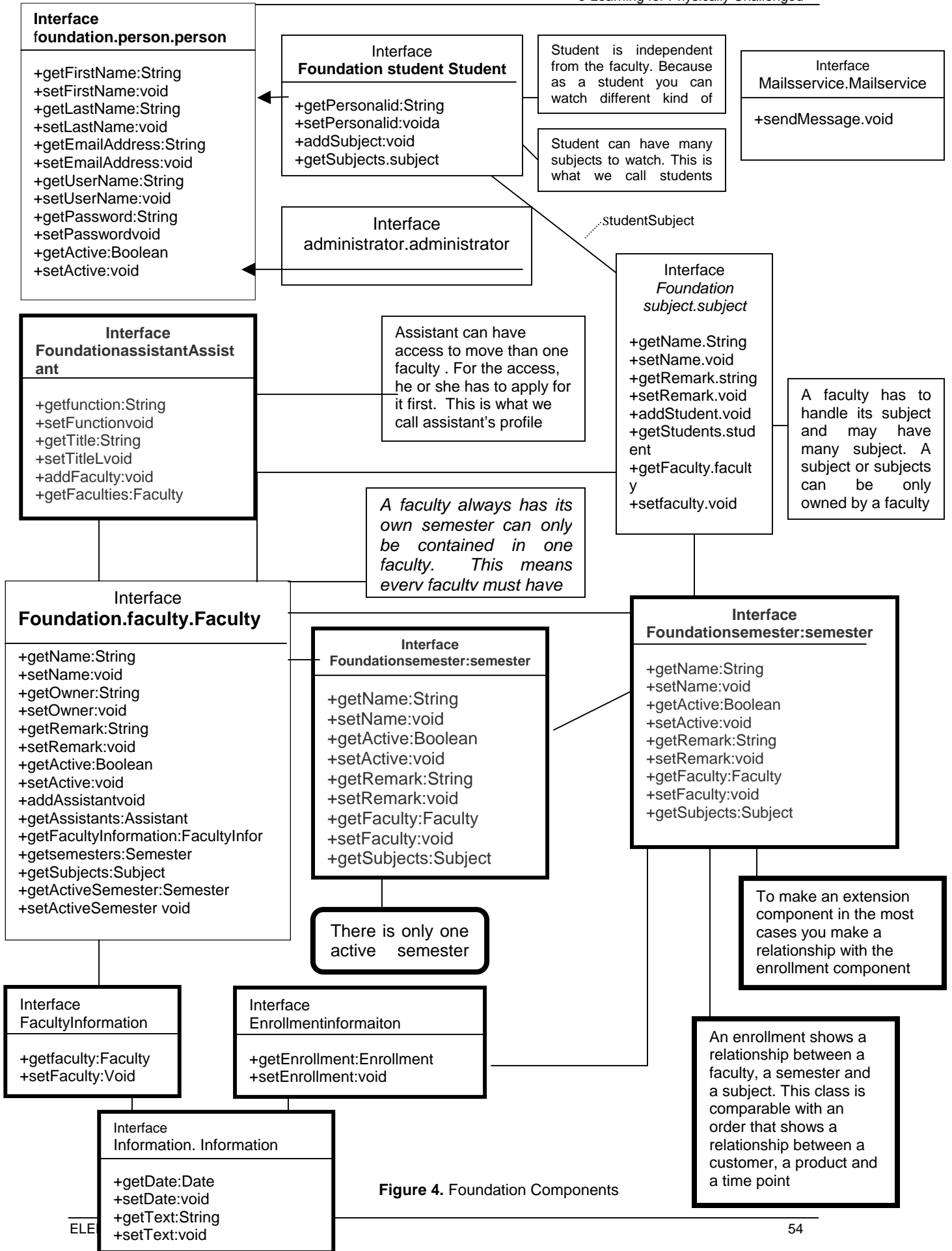


Figure 4. Foundation Components

4. Design Pattern for the Foundation Component

Every foundation component can be divided into three categories as we have seen it in the components specification before: *Business Object*, *Workflow Object* and *Relationship Object*

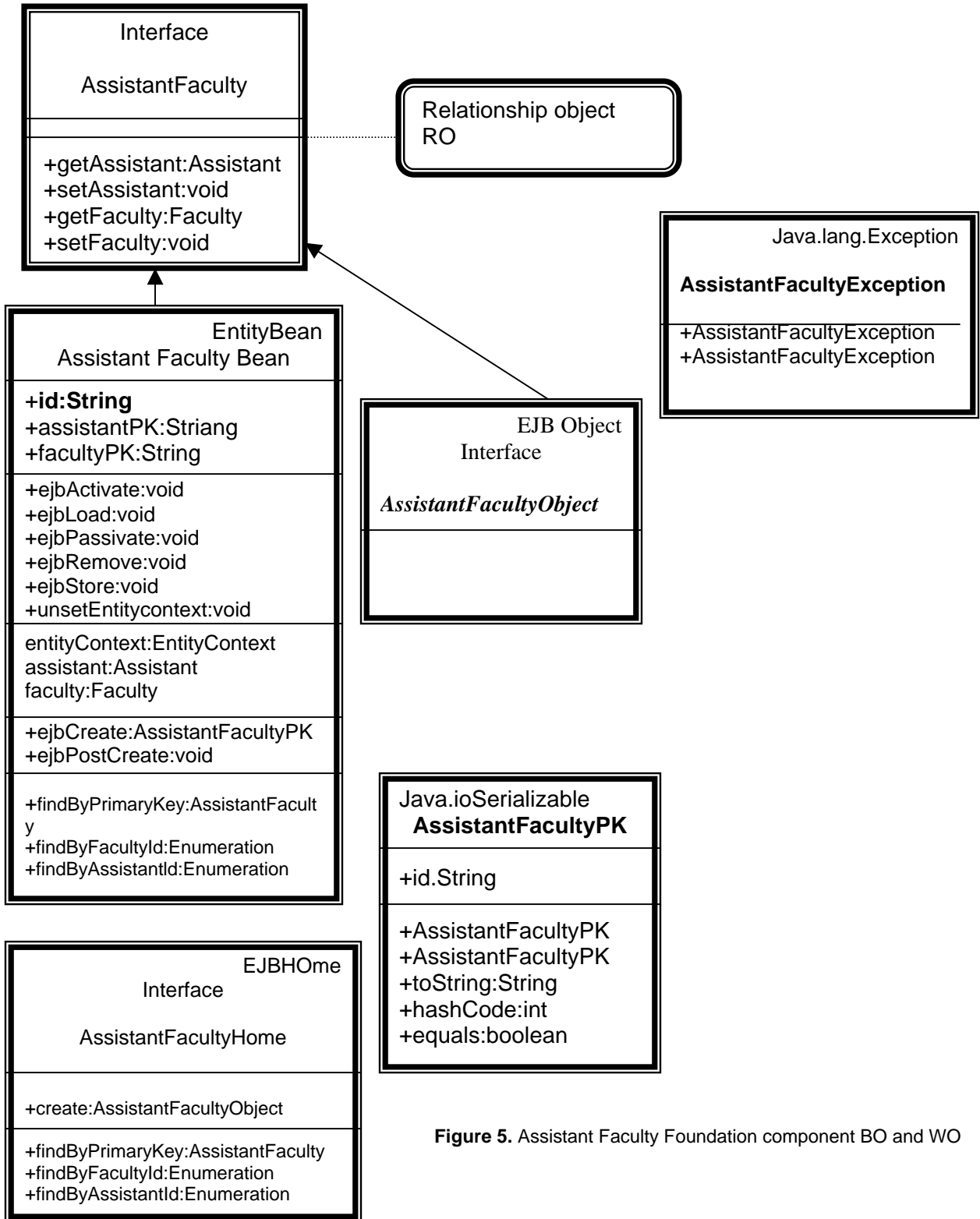


Figure 5. Assistant Faculty Foundation component BO and WO

