ARAHE SOLUTIONS SDN BHD

Façado™ Technical Whitepaper

Suite B-06-06, Plaza Mont Kiara
2, Jalan Kiara, Mont Kiara
50480 Kuala Lumpur
Malaysia
Phone +603-6201-3836 • Fax +603-6201-3837

Table of Contents

List of Figures

Introduction

This document, technical by nature, describes the architecture of Façado™ and its main components.

It is targeted at software architects and developers who want to get a technical overview of Façado™ components and architecture

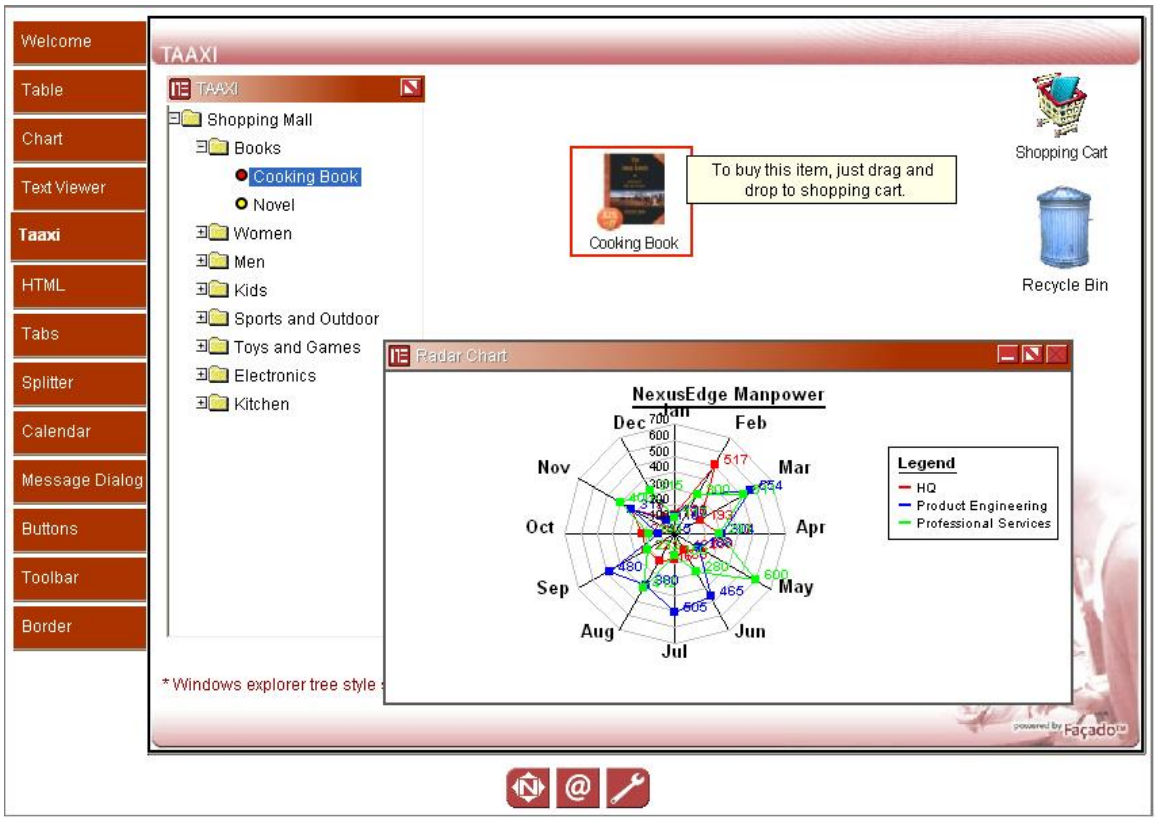

Figure 1 : Sample Façado application

Façado™ Runtime Architecture diagram

The foundation of Façado™ is Java programming language and Extensible Markup Language (XML). Façado™ Runtime architecture leverages these two technologies to provide developers with an object-oriented development platform for the browser, where data transits in XML format.

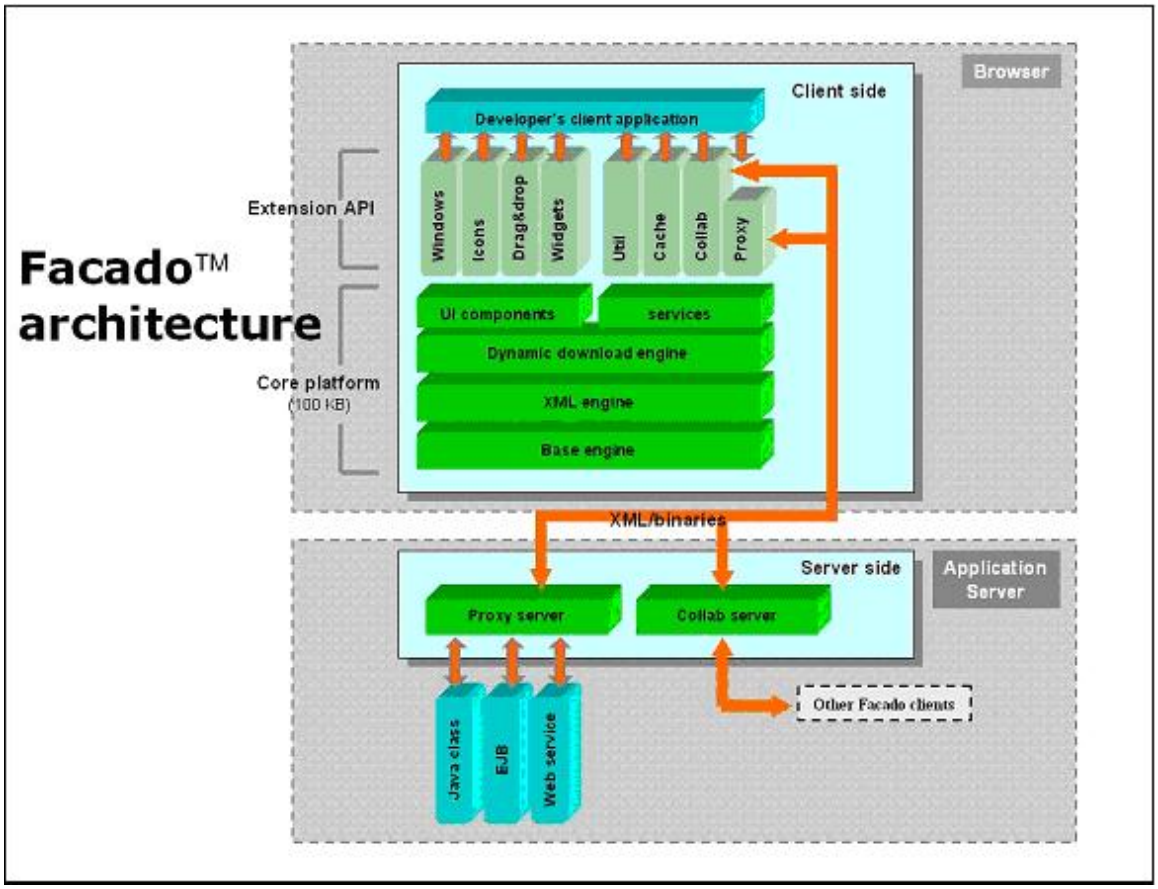

Figure 2 : Façado Runtime Architecture

Façado™ Architecture contains components on both the J2EE application server (these components need to be installed) and on the browser (these components are downloaded when the user accesses the application).

Overview of the server side components

The server side components of Façado™ are to be installed in a Java application server. These components are based on the Servlet technology, so the minimum requirement for the server is to support Java Servlet technology. A full-fledged J2EE application server is not required, although it is preferred for the robustness it offers.

The main goal of the Façado server side components is to facilitate seamless integration between Façado™ and backend business logic. As highlighted in the architecture diagram, building backend logic for Façado is the same as building backend logic for JSP applications. Developers embed their business logic using Enterprise Java Beans, Java classes or Web Services. What Façado™ provides is a layer that allows these business logic components to be exposed to the client application.

Overview of the client side components

The execution environment on the client machine contains the following layers:

- The browser provides the runtime environment for Java (Façado™ can run on Java version 1.1 and above);
- Façado™ client components provide the runtime environment for the application, by managing the application execution, the download of the different services, the communication with the server and the entire graphical environment;
- The developer's application components provide the application specific User Interface and client logic

The client side components of Façado™ and the developer's application components do not need to be installed on the client machine. They are installed on the web server, and get downloaded on the client machine's browser when the user accesses the application.

In order to start a Façado™ application, a user simply needs to browse an HTML document that contains the information related to the application. Browsing this HTML document causes the browser to start the Façado application. When a Façado™ application starts, it first downloads from the web server a core platform of about 100KB in size, which contains the patent pending fundamental components of Façado™.
Once downloaded, this core platform starts executing on the client. It reads a set of XML configuration files that describe the application and its various components, and starts executing the application based on this description.
Depending on the application needs, additional Façado™ extension components might be required on the client machine and therefore downloaded by the Façado™ core platform. These additional components are widgets and services that the developer of the application can make use of. The downloading of these components is seamless for the user and for the developer, as it is entirely managed by Façado™ runtime environment.

The server-side and client side components of Façado™ are described in the more details in the following sections.

Façado™ core platform

Façado™ core platform is the foundation of the Façado™ runtime environment. It provides developers with a graphical environment around a “Tab” paradigm that allows them to replicate desktop-type functionalities, together with an advanced service-on-demand engine to selectively download the components that are needed to run the application.

Base engine

Façado™ base engine contains the runtime kernel for applications to execute. It articulates the application in multiple tabs and contains the desktop features that give applications the look and feel of a client server application.

One important feature of the base engine is memory management. Because memory footprint is a concern in Java and especially on the browser where resources are limited, Façado™ manages the quantity of memory that the application utilizes. Two patent pending mechanisms have been developed for this purpose:

- Caching of object - Some objects are frequently used by an application, for example, fonts and images. Façado™ is able to cache these objects and share them between components so that one and only one copy of this object is used. This reduces the memory footprint of the application. Performance also increases because new services requiring this object will not have to create it but simply access the cached version.

- Freeing of unused objects - A memory module monitors the use of the cached objects. Objects held in cache that have not been used for pre-defined periods are cleared from the cache and the memory that they occupied is freed.

XML engine

All Façado™ applications are described in a set of XML documents. From the color schemes, orientation of tabs and other User Interface settings, to the list of components to display and their detailed properties, these XML documents are the main source of customization of Façado™ applications. As XML is fundamental in Façado™, a dedicated engine was developed to handle XML conversion to Java and vice versa. More details on this component can be obtained from the document “Bean to XML converter”.

Dynamic download engine

This download feature forms the heart of the Façado™ technology and is the key to deploying complex services and applications efficiently and fast over the Internet. This patent pending engine seamlessly downloads only those services and applications components needed by the user as and when the user requires them. Application components here denote Façado™ components and classes developed by the application developer (therefore not part of Façado™). User interaction and tab panel

activities are prime determinants of which services and applications components are to be downloaded. Downloading of services is done automatically by the platform and does not require the user or the developer to do any particular action or special handling.

With the dynamic downloading, a large sophisticated application is downloaded in several steps, thus reducing the startup time.

This makes use of the 80/20 rule. This rule states that 80% of the time, users interacts with only 20% of an application. Because not all application components are required for a given application module to work, Façado™ ensures that only the necessary components are downloaded, making the application more responsive and faster to download.

Another way to describe this feature is to draw a parallel between an application and a Jigsaw Puzzle. An application can be compared to a large poster image. It contains various components, just like an image can be divided into pieces of jigsaw puzzle. A user interacting with a module of an application is like a person looking at the details of a poster. In order to see the detailed portion of an image, one does not need to assemble the whole image; the pieces that form this particular portion are sufficient. Every time a person wants to see a new portion of the image, the corresponding pieces are assembled, so that if the person goes from one portion to another one, the image forms slowly on the screen. In a similar way, Façado™ breaks an application in multiple pieces, so that when the user accesses a module, the corresponding components get downloaded. When the user goes from one module to another one, the application slowly forms on the client machine, just like the full image of the jigsaw puzzle.

Without this downloading engine, the download of the full application would be required in order for the user to start using any module. For complex business applications, this is not practical, because the full application might be a few Megabytes in size. This is why the dynamic download engine is a critical part of Façado™ platform.

Extension Platform

Façado™ extension platform contains all the optional Façado™ components. These components are downloaded and used by a Façado™ application only if the application code makes use of them. They are divided into two categories:

- User Interface components: these are widget components that help building complex User Interface. Base windows, icons and various other widgets (table, tree, calendars, charts etc…) are among the components in this category
- Services: these are non graphical components that provide application developers with already-built service that typical application would require, such as communication with backend business logic or caching

Windows

These represent the basic windows widgets that make Façado™ applications look and behave like a desktop in a browser. Also know as Frames, they are independent, can be moved, resized, collapsed and minimized. Developers typically extend their functionalities to include their application specific behavior. What the developers need to concentrate on is the User Interface inside the frame. Façado™ handles the external behavior of the frame, giving the users of the application a rich graphical environment with no coding required.

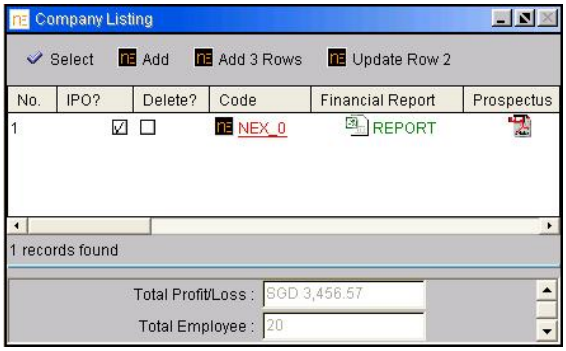

Figure 3 : example of Window component in Façado

Figure 3 shows an example of a frame customized by the developer. The table and buttons are customized by the developer, but the frame, title bar and their associated behavior are part of Façado™ Window component.

Icons

Façado™ gives the user a paradigm where icons and graphics are displayed onto a desktop, and can be move and dragged and dropped to trigger specific behaviors. Each of these icons represents an object that carries data and behavior, bringing the concept of Object-Oriented technology to the browser.

For instance, users can see icons representing database records, and when manipulating these icons, the underlying data gets affected.

This highly graphical workspace gives the user a visual and intuitive representation of the operation that he/she is trying to do, thus enhancing the user experience and productivity of the application.

Drag and drop

Façado™ provides built-in drag and drop function between graphical components. As long as their components execute in the Façado™ runtime environment, developers can provide drag and drop capabilities to the user, by simply implementing Java interfaces that dictate the action to be triggered when a drag and drop operation occurs. No User Interface coding is required as Façado™ handles it at the platform level.

Widgets

Modern applications and business process require complex user interfaces. Façado™ provides an extensive list of User Interface widgets that cater to complex application development.

Among this list, we find components list tree structure, tables, charting components, calendars, various date and time fields, buttons, multilingual text viewer, tabs and toolbars.

These components are Java components that developers can use to quickly assemble a complex application.

For instance, in order to re-create a component that looks like a file explorer, a developer can make use of a tree, a table, a panel splitter, and add all these components in a Façado™ frame, as illustrated in below.

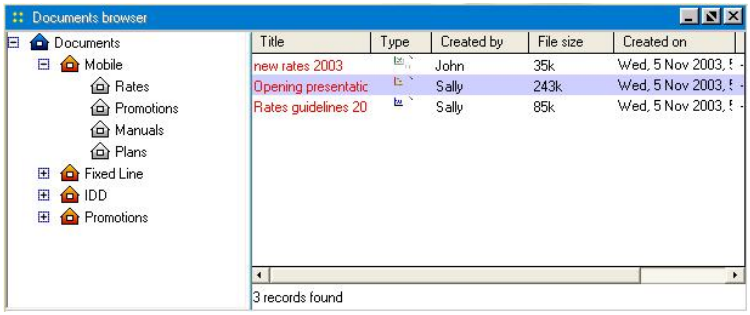

Figure 4 : document browser using multiple widgets

The document “Façado™ Universal UI Server: User Interface Widgets” gives an overview of all the widgets that Façado™ offers.

All Façado widgets are Javabean compliant and can be imported into Java development tools. Façado comes with a packaging of those Javabeans for Netbeans 3.6 (see “Façado widgets for Netbeans 3.6 User Guide”) and for Eclipse Visual Editor 0.5 and 1.0 (see “FaçadoStudio™ Eclipse plugin User Guide”).

Util/cache

This group of components handles services that are necessary on typical applications, and that developers usually need to re-create. In this group we find components like image loading and caching repository, environment customization, fonts factory, various conversion tools.
Caching features also play an important role in applications. This module contains data caching functionalities that ensure that the data gets efficiently managed on the client browser

Proxy/proxy server

This component handles the integration between the client application and the server business logic components. As highlighted in the architecture diagram, this integration is done through a client side component (server proxy) and a server side component (Façado server).

A developer who needs to invoke a business logic resource on the backend server can do so from the client by calling a service available in the proxy. The Façado™ platform handles the communication between the client and the server, and the invocation of the business logic on the server.

The following figure shows the architecture of the proxy.

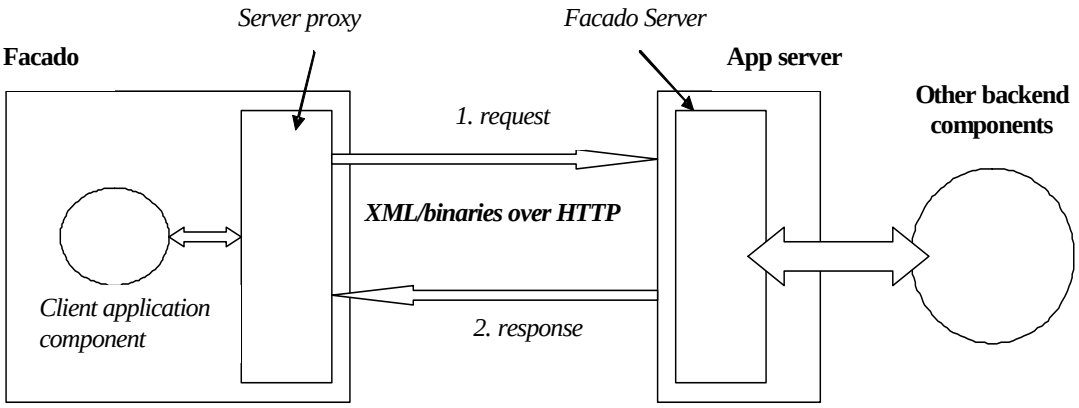

Figure 5 : architecture of the proxy

The client application component makes a local call to the Proxy on the client. The proxy sends the request to the server proxy through HTTP using either XML message (in a similar way as SOAP) or binaries (Java serialization). The server proxy reads the request, locates the server resource, invokes it and returns the result to the client calling client application component.

A more detailed description of this component can be obtained in the document “Façado™ integration with a server”

Collab/Collab server

FaçadoCollab is a framework that allows applications to be updated in a real-time fashion from events happening on the server, through a normal HTTP connection (meaning that even if the client is behind a firewall, the update can still be obtained).

FaçadoCollab consists of a client side component (collab) and a server side counterpart (collab server), and is based on exchange of messages between server and client.

The collab server component is responsible for routing the messages to the client layer, based on user id and message type.
The client side collab handles the reception of the messages and dispatches the message to the components that have subscribed to this type of message.

Developers can make use of existing message types to send notification to a Façado™ application, or they can extend the framework with their custom, business specific message types.

A typical example of use of FaçadoCollab is the asynchronous update of stock prices. Whenever a change of stock price occurs, the server can notify Façado™ by sending a message of type ‘stockPrice’. An application component intercepts the message on the client and updates the User Interface accordingly.

More information about FaçadoCollab can be found in the “FaçadoCollab technical white paper”.

FaçadoStudio™

FaçadoStudio™ is the development environment for Façado. It provides the developer with a WYSIWYG editor to visually design a Façado application, select and design tabs and desktops, and assemble the application by selecting components that will appear on each desktop panel. Based on Java 1.4, it comes as a standalone Swing application or a plugin to the Eclipse platform.

As illustrated below, FaçadoStudio™ is an instance of Façado and it provides immediate update of the application while editing its properties.

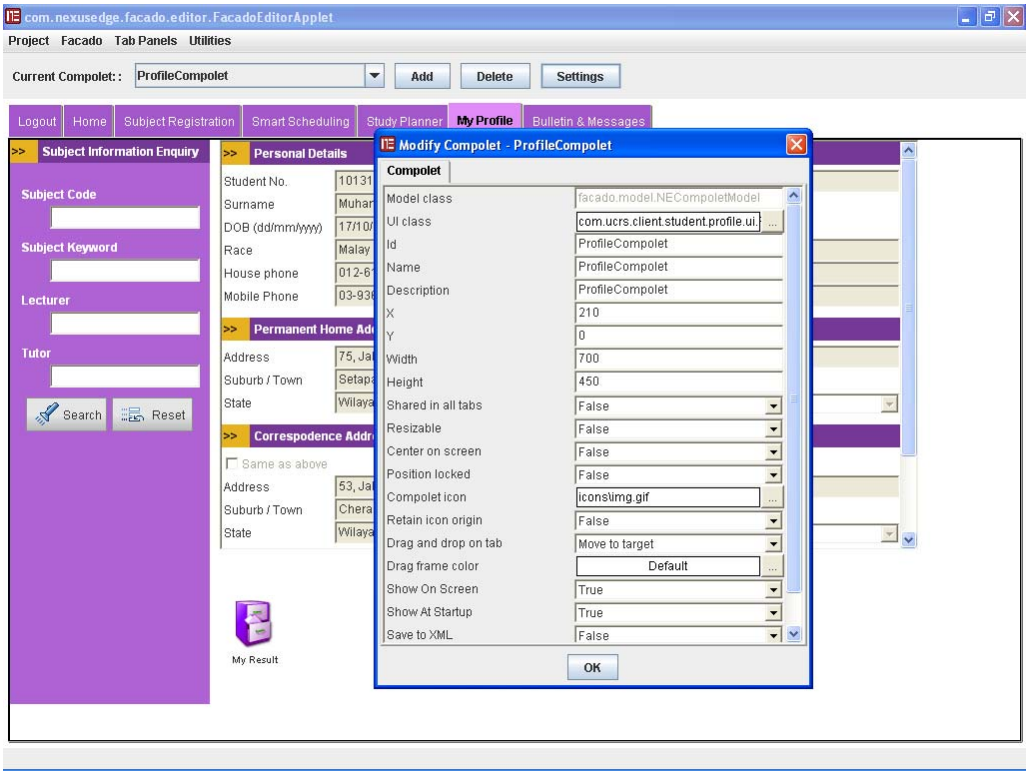

Figure 6 : screenshot of FaçadoStudio™

The document “FaçadoStudio™ User Guide” provides a detailed description of the installation and various functionalities of the tool.

FaçadoStudio™ Eclipse edition provides a fully integrated development environment (IDE), combining the following:

- Powerful Project management and Java editing from Eclipse platform
- Visual designer to build User Interface panels by dragging and dropping basic Java components and advanced Façado UI widgets controls
- FaçadoStudio™ integrated with Eclipse

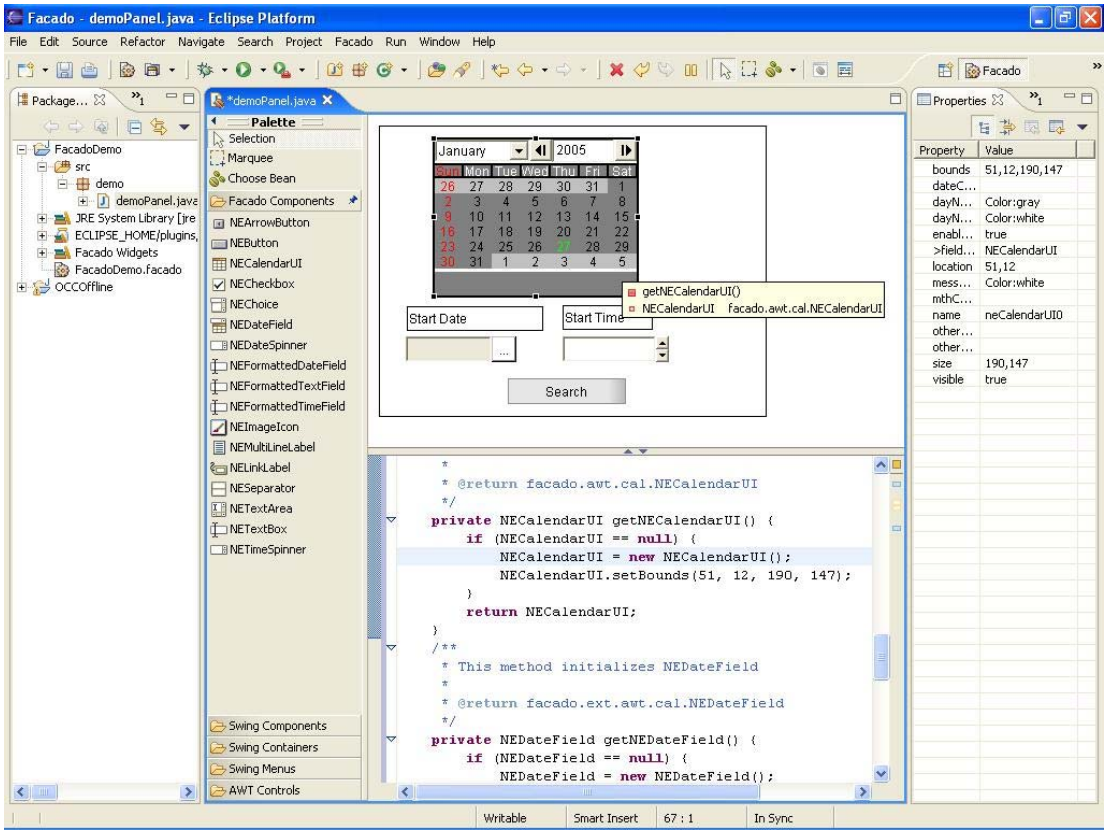

Figure 7 : FaçadoStudio integrated with Eclipse

For more information, refer to the document “FaçadoStudio Eclipse plugin User Guide”.