

ONE Client: An RCP application to browse NewsML news

Alessandro Assab¹, Angelo Marrara², Gianvito Rossi¹ and Rodolfo Totaro¹

¹ SP-Process s.p.a. c/o Tecnopolis CSATA
Strada Prov.le per Casamassima, Km. 3
70010 Valenzano - Bari, Italy
{aassab, grossi, rtotaro}@sp-process.it

² ANSA Via della Dataria, 94 00187 Roma
angelo.marrara@ansa.it

Abstract. ONE Client is an application developed with Eclipse RCP technology in the publishing domain. Ansa (Agenzia Nazionale Stampa Associata), the main Italian press agency, developed a new communication protocol for real time news: ONE (Open News Engine). The protocol uses the NewsML G2 as news format. ONE client is a news client for ONE protocol and NewsML format. We developed the application using Eclipse RCP 3.3.2 and Nebula Project widgets.

Keywords: RCP, NewsML, Eclipse, client/server, Nebula project, ONE

1 Introduction

ONE Client is an application realized with Eclipse RCP technology in the publishing domain. Ansa (Agenzia Nazionale Stampa Associata) needed to create a new client for its new communication protocol for real time news: ONE (Open News Engine). The new protocol allows supporting multi agency connection and uses the NewsML G2 as news format.

After the feasibility studies realized by ANSA for comparing different developing platforms, described in this paper in the section 2, SP-Process and ANSA worked together to realize ONE Client. ONE Client is a porting/evolution of an existing application, MNB (Multimedia News Browser) 2.0, implemented using Swing libraries and a different news protocol and format.

ONE Client allows receiving news in real time from the selected press agency ONE compatible servers and supports different news kinds: text, photo, movies and RSS. Moreover ONE Client embeds web browser functionalities, so the user can use an all-in-one application to access to information that s/he needs. In the section 3 ONE client features will be shortly described.

We chose to use the Nebula project widgets and SWT browser to support the different kinds of news. The sections 4 and 5 describe some technical solutions implemented in the client and report some occurred problems. In particular:

- the implementation of Gallery Viewer based on the Nebula Gallery widget
- the use of SWT Browser
- an example of extension point usage

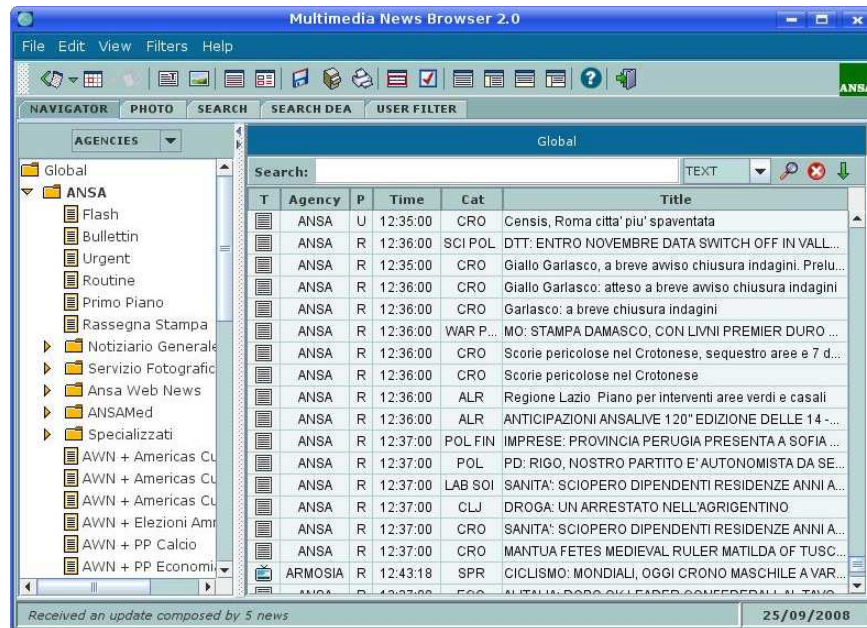


Fig. 1. The main screenshot of MNB 2.0 on Linux Platform

In the following paragraph we give you more details about the usage scenario and the NewsML standard.

1.1 Scenario and NewsML

Currently, news written by press agencies can be read by User in two different ways: Local Reception and Remote Access.

In Local Reception news are sent by Agencies and received by User, through a broadcast system via satellite. This solution needs a very specific communication infrastructure on Client side (satellite dish, decoder ...).

The Remote Access allows User to access to the news directly via Internet. However, in this case the User needs a specific client application for each agency installed on her workstation.

The ONE solution allows meeting the needs of Users and Agencies. In particular, User can access to news via internet using only a client and without any specific

communication infrastructure. Agencies can control access to every news from each workstation or from each single user connection. Moreover, Agencies can monitor and trace every news access, and directly manage user activation and deactivation.

ONE is a client/server solution based on the following components:

- *ONE Client* allows user to connect to more Agencies at same time and to receive news.
- *ONE GatewayServer* manages a lot of news sources and communicates with the connected ONE Clients.
- *ONE Protocol* is based on HTTP specification and allows the communication between ONE Client and ONE GatewayServer.

To exchange news between server and client side, ONE adopts the NewsML G2 format developed by IPTC (International Press Telecommunication Council). The NewsML allows to support many kinds of news (text, photo, movies) and to contain meta-information about contents.

2 Why Eclipse RCP?

There exists a lot of useful technologies as an alternative to Eclipse RCP, to implement ONE client; in particular, Netbeans RCP and the Microsoft .NET Framework. In the following we will shortly introduce these technologies and describe reasons why we chose Eclipse RCP.

Netbeans RCP: is a generic desktop application. It provides the services common to almost all large desktop applications: window management, menus, settings and storage, and file access. The reuse of these standard components allows you to concentrate fully on your application's business logic. It is based on Swing library, so it is a pure Java application.

Microsoft .NET Framework: is a software technology that is available with several Microsoft Windows operating systems. It includes a large library of pre-coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

In the table 1 the main advantages and disadvantages of the technologies above mentioned are synthesized. A ONE Client project constraint was to develop a multiplatform application, so we discarded Microsoft .NET framework.

It is not simple to take a final decision between Netbeans and Eclipse RCP: the Swing libraries used by NetBeans allow developing a richer GUI if compared with SWT. On the other hand, SWT libraries provide a native operating system Look&Feel but are only an OS widgets subset. Both Eclipse RCP and Netbeans have a pluggable architecture, but the Eclipse one is based on OSGi standard and, moreover, it has a more complete frameworks set, in particular help and update support. Another important issue for ONE Client project was the table performance because a user can receive about 10-20 thousands news for day, so SWT performance and JFace viewer virtualization support were very helpful for our goal.

Table 1. Summary of advantages and disadvantages of candidate technologies to be adopted for ONE client

Technology	Advantage	Disadvantage
Eclipse RCP	<ul style="list-style-type: none"> - Native look&feel via SWT - Update and help framework - Plug in support with OSGi standard - High performance with SWT - Multiplatform 	<ul style="list-style-type: none"> - Very complex architecture - Low customization
Netbeans RCP	<ul style="list-style-type: none"> - Pure 100% Java - Plug in support 	<ul style="list-style-type: none"> - Swing performance - No native look&feel - No Browser support - No OSGi support
.NET Framework	<ul style="list-style-type: none"> - Good IDE - Very simple to deploy on windows system - High performance 	<ul style="list-style-type: none"> - Complex to deploy on not windows system - IDE expensive - IDE needs a very performed workstation

We chose Eclipse RCP for its completeness (if compared with Netbeans), for SWT performance and for the help and the update support. On the other hand ONE Client did not require a so rich GUI and the Nebula Project widgets cover all the more complex needs.

3 ONE Client

The ONE Client is an Eclipse RCP multi user application that allows each user to define a list of connections to press agencies.

The main window contains a menu bar with all functions available in the application, a toolbar in which the user can activate the main functions and can change the reference date and a custom perspective switcher for the following five perspectives:

Agencies Perspective contains a view to visualize the news titles list. The user can choose the news to read from this list. Moreover, there are two views to manage local news archives and custom filters for news.

Photo Gallery Perspective contains a view where the user can visualize a preview of photo news.

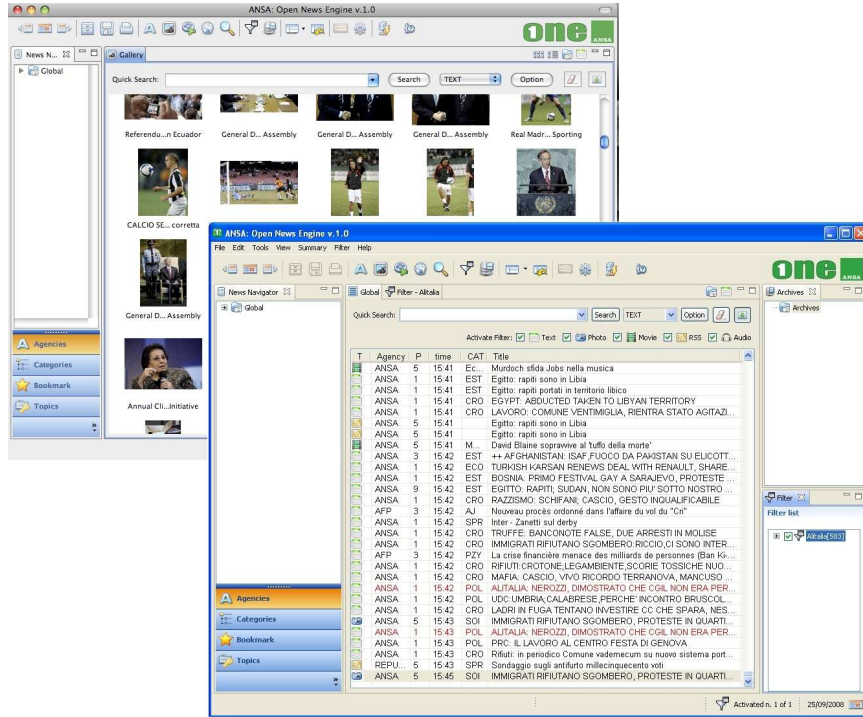


Fig. 2. Gallery Perspective on Mac platform and Agency Perspective on Windows XP platform

Video Gallery Perspective contains a view where the user can visualize a preview of video news and can play videos.

Browser Perspective contains the Eclipse browser view to navigate the Internet.

Search Perspective allows the user to define a query based on CQL language, to find news on real time servers or historical archives.

Moreover, the workbench contains a view named Navigator View that allows user to filter news in each perspective.

4 Technical solutions

In this session some technical solutions adopted in ONE client that use specific Eclipse RCP or Nebula Project features will be described. In particular, we will explain why we chose to use SWT Browser, how we exploited the Eclipse extension registry to create a modular and extensible filtering feature and how we equipped the Gallery widget with an ad hoc JFace-like viewer.

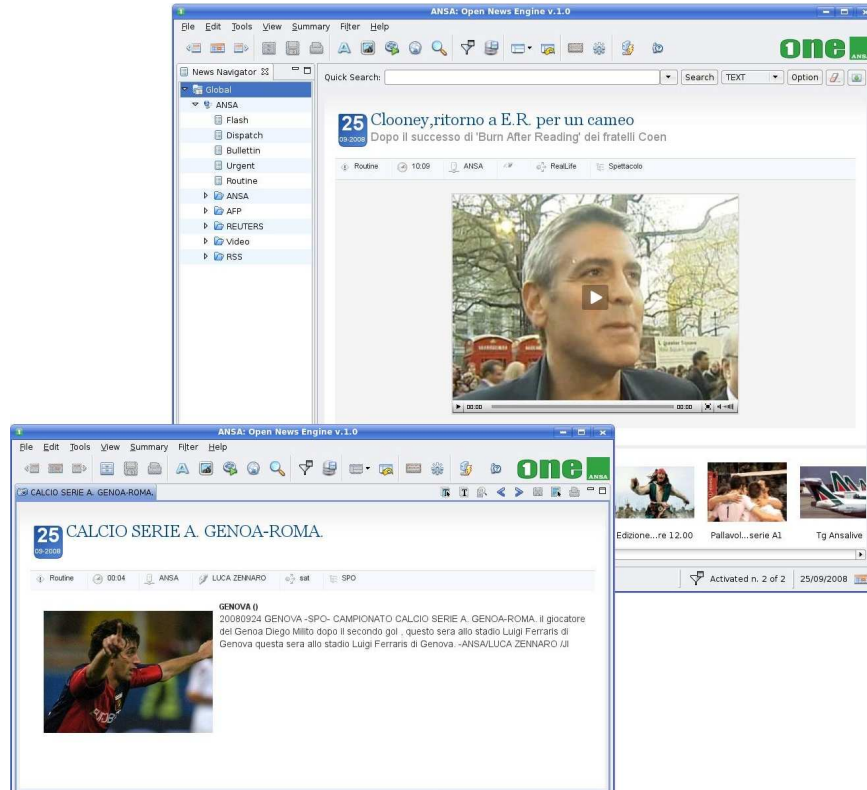


Fig. 3. A movies and a photo news on Linux Platform

4.1 SWT Browser

The main technical problem to solve for ONE client was to support many kinds of news: text, photo, movies and RSS. Supporting the visualization of text news is very simple, for example using the `StyledText` widget of SWT libraries, but supporting photo and movie news is more complex. Photos and movies could be coded in a lot of formats, and new formats could be created in the future. On the other hand, Java does not provide useful libraries to support movies decoding and the existing ones, for example JMF, are so far from the state of the art of movies technologies. A possible solution could be to use native libraries through JNI technologies, but the multiplatform project constraint makes complex the software maintenance. For all this reasons we chose to use the SWT browser widget to visualize all kinds of news. SWT browser delegates the news visualization to the installed browser, moreover any new movie or photo format is usually quickly supported by any browser, so these potential changes do not impact on the ONE client code.

Unfortunately SWT browser has a very simple API and we were forced to use JavaScript code to interact with it. JavaScript code is often browser dependent so we needed to test all this code on all supported platforms.

4.2 Navigator Extension Point

In the ONE client the Navigator is a view used to filter news in the Title or Gallery view. This view is composed by a **CollapsibleButtons** (Nebula Widget) and a **Composite** on which is set a **StackLayout** named “Navigation area”. Each button in the **CollapsibleButtons** is associated to a widget put in the navigation area and it represents a specific filter type; currently news can be filtered by agency, topic, category and user bookmarks. In the future other filter types could be needed, so we defined an extension point to add new buttons, therefore filter features, to navigator view.

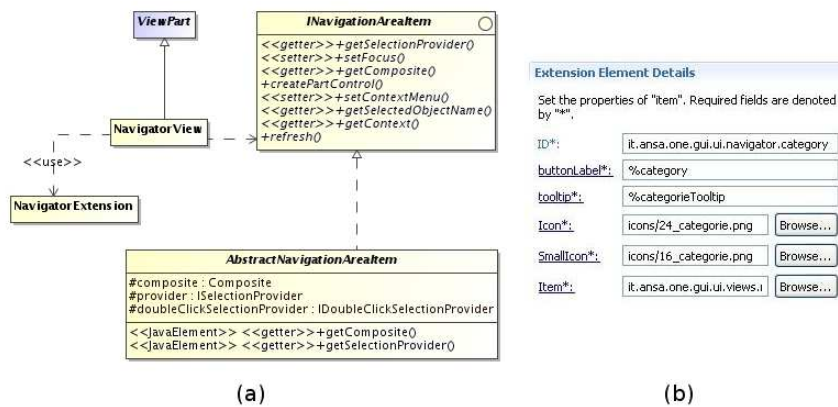


Fig. 4. Class diagram: Abstract structure of Navigator extension point (a) and a screenshot of the related extension form (b)

A new filter feature is characterized by a name, tooltip, icons used in the **CollapsibleButtons**, and an instance of **INavigationAreaItem** that is put in the navigation area. The figure 4 illustrates the abstract structure that allows the Navigator view to use an extension point for instancing **INavigationAreaItem** classes.

4.3 Gallery Viewer

We use the Gallery Widget by Nebula Project to show to the user a preview of photo or movie news.

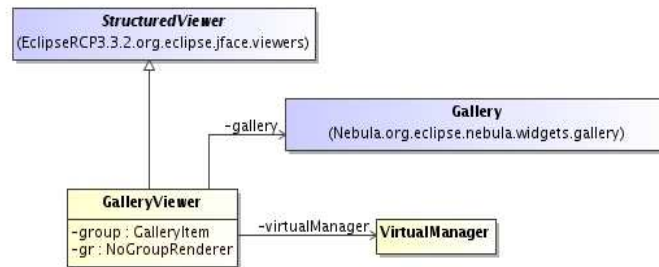


Fig. 5. Gallery Viewer Class Diagram

We needed to have a common interface to interact with the gallery and the title view from the Navigator. In the title view, we use the JFace `TableViewer` with the virtualization to implement the news title table; this choice makes simple to apply filter and sorter to the news through the JFace API. We needed the same API interface for the Gallery to simplify the code. Nebula Project provides also a tree viewer based on the gallery widget and the JFace viewer hierarchy, but it is not helpful for our goals. This viewer does not support the virtualization and manages a tree structure where photos can be only leaf of the tree.

We needed to manage a simple photo list without any hierarchy and, most importantly, we must have a virtualized gallery because we load a lot of photos in the widget.

For this reasons we chose to implement a new viewer with the previous features. The figure 5 shows the class diagram that describes how our viewer interacts with JFace and Nebula Project classes. Our `GalleryViewer` extends the abstract class `StructuredViewer` of JFace and uses the `Gallery` widget of Nebula Project. Our viewer inherits all filter and sort features from JFace hierarchy, but we implemented an ad hoc virtualization policy.

5 Conclusions

We developed ONE Client using Eclipse RCP technology with good performance and, in the first test phase, with user's satisfaction.

On the technical point of view, some problems occurred on Mac-OX platform on which we must use Java 5.0 because Eclipse still doesn't work with Java 6.0 on Mac-OX. We found a lot of debug and maintenance problems with SWT browser caused by JavaScript interaction, so we hope in a new and more complete API for the browser support in future Eclipse release. Finally we found some expected bugs in Nebula Project widget that we fixed or worked around, but it is an incubator project yet.

References

1. Eclipse Rich Client Platform, <http://www.eclipse.org/rcp>
2. Robert Harris, Rob Warner: The Definitive Guide to SWT and JFACE Apress 3.
3. Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional
4. Jeff McAffer, Jean-Michel Lemieux: Eclipse Rich Client Platform, Addison-Wesley Professional
5. NewML <http://www.newsml.org/>
6. IPTC (International Press Telecommunications Council) <http://www.iptc.org>
7. Netbeans Platform: <http://www.netbeans.org/products/platform/>
8. Nebula Project: <http://www.eclipse.org/nebula/>
9. Microsoft .NET: <http://www.microsoft.com/net/>