International Journal of Web Portals
Volume 13 • Issue 1 • January-June 2021

An Information Retrieval System Based on Multiple Portlets:
Communication Between Its Components

Glauber José Vaz, Embrapa Informática Agropecuária, Brazil
Jayme Garcia Arnal Barbedo, Embrapa Informática Agropecuária, Brazil

ABSTRACT

Information retrieval systems built with a service-oriented architecture have numerous advantages, and portlets are a key technology to implement services which interact with each other in the presentation layer. This work presents an efficient approach for the communication between the components of an information retrieval system based on multiple portlets in a single user interface. It also presents the architecture and the main methods of the system implemented as a case of use for this approach. It is shown that the proposed solution yields the best inter-portlet communication mechanism in each situation, while possessing the ability to deliver aggregated search and superior user experience.

KEYWORDS

Aggregated Search, Elasticsearch, Inter-Portlet Communication, Java Portlet Specification, Liferay Portal, Portal Technology, Search Engine, Service-Oriented Architecture, User Interface

INTRODUCTION

Aggregated search represents a new information retrieval paradigm, in which information is not only retrieved but also assembled (Kopliku, Pinel-Sauvagnat, & Boughanem, 2014). Its results include diverse content (images, videos, news, maps) and relational content (similar entities, features) in addition to ranked items. The final result in an aggregated search is presented in a single interface and can be whatever content combination useful to the user (Kopliku et al., 2014).

Aggregated search can be provided by an information retrieval system based on a service-oriented architecture (SOA) approach, whose benefits are well known (The Open Group SOA Working Group, 2019). It frees users to concentrate on their business problem, leaving the details of the solution to the service (Laskey & Laskey, 2009). According to the Java Portlet Specification 2.0, JSR-286 (Hepper, 2008), in a SOA one does not write monolithic applications, but separate services that can be orchestrated into applications. Portlets provide an user interface based on components for the services, in a way that they are consistently integrated. Portlets play a crucial role in developing SOA applications, as they represent services in components that can interact with each other at the user interface layer (Sarin, 2012). In order to provide component consistent aggregation, there are mechanisms for inter-portlet communication which result in a better user experience.

DOI: 10.4018/IJWP.2021010105

According to Gartner, Inc. (Murphy, Phifer, Tay, & Lowndes, 2018), organizations use digital experience platforms (DXP) to provide improved digital experiences. The Gartner's Magic Quadrant for DXPs has replaced the Magic Quadrant for Horizontal Portals to emphasize this support to individualized digital experiences. It listed four leaders in 2018. Three of them implement the portlet specification, which reveals the relevance of this technology in this context. Wilson, Daniel, Jugel, and Soi (2011) claim that inter-portlet communication mechanisms are limited because the interactions occur on the server-side and it is limited to the Java world, but they are appropriate when the system requires backend data processing, it is embedded in a portlet-based platform and its components must be integrated and coordinated. These are exactly the conditions under which this work was developed, which motivated the proposition of a solution for information retrieval in a platform that implements portlets. Furthermore, inter-portlet communication is consolidated by Java specifications (Hepper, 2008; Nicklous, 2016).

Many studies show this trend toward SOA and portlets. Thomas (2007) cited some portal architectures that use portlets in a component-based architecture, as they can provide a user experience that hides the complexity of services that support the portal. The benefits of portlet technology and SOA are also highlighted by Akram, Chohan, Meredith, and Allan (2007). Daniel et al. (2007) investigated the problem of components integration in the presentation layer. Even before the JSR-286 specification, which introduced inter-portlet communication mechanisms, they had defended that web portal development was the most advanced approach to user interface composition, due to the portlets associated. The first version of Java Portlet Specification, JSR-168 (Abdelnur & Hepper, 2003), does not provide portlet events or public render parameters for coordinating portlets. Public render parameters are intended for sharing view states across portlets and portlet events enable portlets to react to actions or state changes not directly related to an interaction of the user with the portlet. They both were first published in the second version of Java Portlet Specification, JSR-286 (Hepper, 2008). The third and current version, JSR-362 (Nicklous, 2016), does not change these communication mechanisms.

Tvarožek, Barla, and Bieliková (2007) used portlets to integrate different functionalities into a single user interface of web-based information systems. They cited some advantages of portlets: various types of presentation of the same content can be provided to different users, access to portlets can be controlled by user roles and access privileges, the usability of the interface can be improved and the functionalities can be adapted according to the needs and goals of the user.

Bauer, Berleant, Cornell, and Belford (2015) also used portlets in WikiHyperGlossary(WHG), an information literacy technology for chemistry documents. It enhances reading comprehension by hyperlinking terms in documents to additional information and resources. When a user clicks on one of these terms, the information associated with the term is retrieved and displayed in the WHG Portlet, which is a JavaScript generated overlay that is superimposed on the document. More than one portlet can be activated and moved around the screen by the user.

Haubmann-Haidvogel, Scharl, and Weichselbraun (2009) investigated methods to build Web content repositories and visual means for searching and exploring them. They used portlets for user interface so that an action in any of the provided portlets triggers an immediate update of all other portlets. For synchronizing the multiple coordinated views, they proposed an architecture using an observer pattern and a JavaScript object so that interaction in any coordinated view notifies the main one, which in turn notifies and triggers updates in all related views.

Wong, Crowder, Wills, and Shadbolt (2006) presented a solution for a document repository in a specific area based on multiple portlets that communicate with each other, but it was before JSR-286 and its new inter-portlet communication mechanisms. Although they used portlet sessions in application scope to propagate changes of viewing contexts, sometimes it is better to use portlet events or public render parameters.

Information retrieval systems are very common in the web. CiteSeer, for example, is a digital library and search engine in the field of computer and information science. Due to its monolithic