

Experiences from portlet deployment

Mikael Berglund¹

¹Ladok Division, Umeå University, S-901 87, mikael.berglund@ladok.umu.se

Keywords: portals, web, portlets, uPortal, student services

1. EXECUTIVE SUMMARY

With the increasing consolidation of student information systems in Europe, new solutions are needed to enable cost efficient student services and at the same time giving institutions means to customize look and feel according to their own standards. These services have to be deployed in accordance with the local authentication and authorization schemes and technologies.

In the Swedish Ladok system, a new way of deploying student services as Java Portlets has been developed. This will ease initial deployment and the installation of new versions. This paper will describe the technical solution in detail and give insight to the difficulties and solutions associated with security, user preferences and graphic design. In addition, this paper will describe when this solution is applicable.

1.1. Background

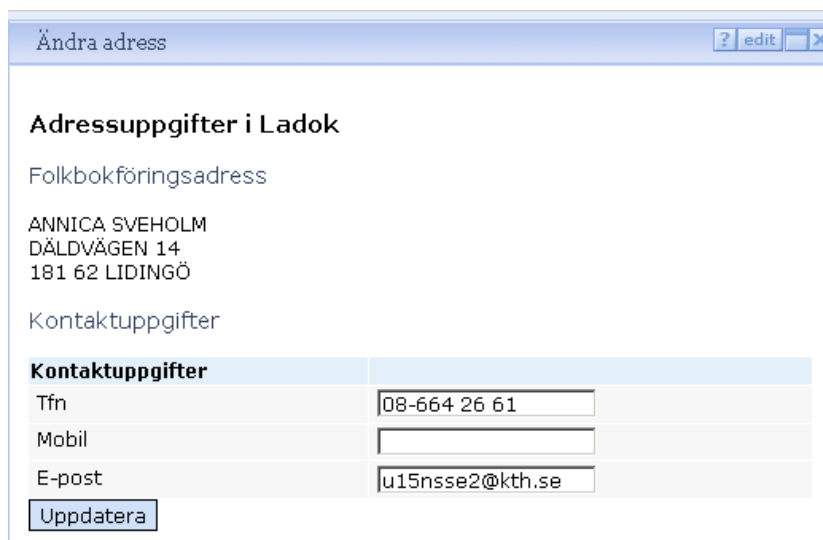
Ladok is the national system used for documentation of academic information at higher education institutions in Sweden. It is jointly owned by these through a consortium. Each institution has its own separate installation of the Ladok system.

Student portals require access to student information. This need resulted in the Ladok on Web application, which is a set of SOAP services. Currently, 24 different web services has been developed, among these are; transcript of records, course registration and examination registration. This has enabled institutions to integrate these services in their portals. However, the resources required for this type of development can be expensive and limited, especially for smaller institutions. The development costs combined with the cost of maintenance has limited the adoption of these services. The increased complexity of AJAX web interfaces and users' demands for more intuitive and accessible applications widens the gap to implement high quality services.

2. PORTLETS

Portlets are pluggable user interface software components that are managed and displayed in a web portal. Portlets produce fragments of markup code that are aggregated into a portal page. Typically, following the desktop metaphor, a portal page is displayed as a collection of non-overlapping portlet windows, where each portlet window displays a portlet (Portlet, 2009).

Portlets are run in a portal server. This server can provide services such as personalization, authentication and theme support.



Ändra adress

Adressuppgifter i Ladok

Folkbokföringsadress

ANNICA SVEHOLM
DÄLDVÄGEN 14
181 62 LIDINGÖ

Kontaktuppgifter

Kontaktuppgifter	
Tfn	<input type="text" value="08-664 26 61"/>
Mobil	<input type="text"/>
E-post	<input type="text" value="u15nsse2@kth.se"/>

Figure 1. Change address portlet.

The image above shows a simple portlet for updating a student's contact information. All graphic elements are theme-able by CSS. This portlet is run in the uPortal framework.

2.1. Authentication

One of the biggest issues is to integrate the institutions authentication and identity management systems. The portlet specification does not specify this. However, there is support for single sign-on solutions such as CAS built in to uPortal and Liferay Enterprise Portal. If there is a need to use proxy tickets, a custom solution has to be developed.

2.2. Authorization

The portlet specification has a standard method to find out if a user is in a certain role. The portal specific implementations on the back-end side vary, and custom solutions have to be implemented to map roles to a local LDAP or other sources of information.

2.3. Development efficiency

Earlier, portlet development has been difficult and without tool support. This has changed with the emerging of the Spring framework. With Spring, portlet development is very similar to traditional Java web development. There are some caveats with deployment to a local test environment, which is not as streamlined as the modern techniques used today. This is a minor point, but nevertheless worth to mention.

2.4. Preferences

When deploying generic services, there is a large need for customization such as authentication and graphic profiles. One often overlooked aspect of generic services is slightly different business procedures at various institutions. In the portlet standard, there is a preferences API used to manage this. We have identified two different types of preferences that can be managed.

- System-wide preferences such as business logic configuration or authentication settings
- User-defined preferences such as sort orders and view settings

The system-wide preferences can be set during installation of a new portlet. These settings are kept during upgrades. The user-defined preferences is stored by the portlet server and kept between log-ins. This is a great way of defining the contract between the portlet and the executing environment.

3. CONCLUSIONS

There are some clear advantages to the portlet development and deployment model, provided that the portlets will be used by several different institutions. In this scenario, the slightly higher development effort will pay for itself during the lifetime of the services deployed. There are some limitations that have to be addressed. These are listed below in no particular order.

- Authentication. There is no support in the portlet standard for standard single-sign on solutions. This has to be customized for each deployment
- Authorization. For services with multiple roles, configuration and development specific to the portal server is needed.
- Theme support. Adoption of the institutions' current themes and style sheets is also needed.

To conclude, delivering student services as portlets can be both efficient and powerful if the issues mentioned above can be addressed.

4. REFERENCES

Portlet - *Wikipedia: The free encyclopedia*. (2009, April 3). FL: Wikimedia Foundation, Inc. Retrieved May 24, 2009, from <http://en.wikipedia.org/wiki/Portlet>