SOFTWARE APPLICATION STRUCTURE GENERATOR FOR MANAGEMENT APPLICATION PORTALS

Florin Răzvan Lupşa-Tătaru, PhD Student, Vladimir Mărăscu-Klein, Prof., PhD, "Transilvania" University of Braşov

Abstract: In management application portals, each software application has its own role and fulfils specific functions for the purpose for which it was created. Meanwhile, it complies with the general structure of the portal and must obey the operation rules of the portal. Each application consists of items related to file structure and operation logic structure. Based on this, one can create a standardized way of implementing any applications and more, a software application with the role of applications structure generator. The paper aims to present the logic behind creating such an application generator.

Keywords: software, generator, application, management, portal

1. Introduction

Management applications portals, by definition, include in addition to information, applications designed to perform various functions. These applications must comply with the general logic and file structure of the portal in order to contribute to the concepts of unified and integrated portal. Create a portal without taking into account the evolution in time can lead to the loss of these concepts. There aren't few situations in which portals were provided with some applications required initially and subsequently, newer applications were hardly integrated into the portal, compromising its integrity. On the other hand it is difficult to predict in long term what information must the portal incorporate. However, the structure of a management applications portal designed to be used over a long period of time should provide a generic, standardized way to integrate new information and applications. And here comes the role of an application structure generator. But this must not be misunderstood as a logic generator for the newly created applications, but as a helper in developing applications and a keeper of portals integrity.

2. Application structure generator

In creating a portal application structure generator one can start from identification of the elements used by an application, generally databases and file structures. In terms of application logic, they contain elements that are related to security of data and files and items related to functionalities that the application fulfils.

If databases needed by the application are difficult to standardize because of their variety and heterogeneity of a future application functions, file structure can be easily divided into its components. Thus, any application uses configuration files, files related to internal functions and third party files. An application structure can be outlined as in Figure 1.

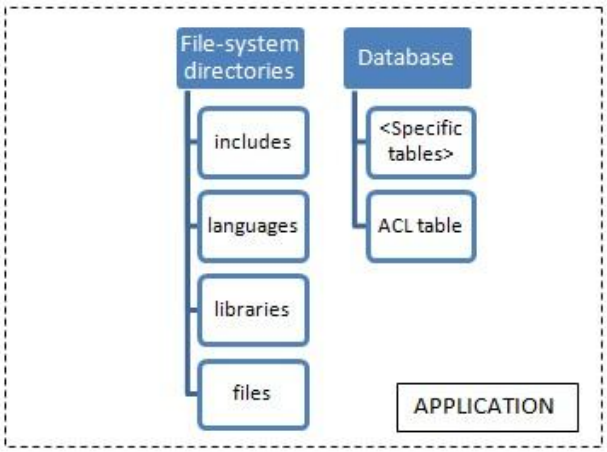

Figure 1. Generic application structure.

Includes directory consists of files that lead to the fulfilment of the back-end application functionalities. Besides those that are specific to each application, there are some that can be standardized, such as configuration files, files that contain parameters, and files that contain custom user settings.

Languages directory consists of files that allow translation elements, localization of the application. They must comply and do not conflict or be redundant with existing localization files in the portal.

Libraries directory consists of files containing application-specific library functions which can be standardized as name and content, according to the programming language used to create the portal. For example, the file that refers to functions that allow the use of database can be called *database_access*, those containing functions related to data display can be called *views* and those who manage access control can be called *access*. The directory may contain third party functions files.

Files directory consists of general files related to application without role in the functioning of the application logic, such as pictures, documents, etc.

The application database contains tables related to application and in addition, if a control system that uses database engine is implemented it may contain a table linked to an access control system based on ACL lists.

Of course, such a structure generator requires rights to create databases and write the necessary files. Also, the generator must give to any new application the possibility of access to the configuration and parameterization files of the portal, to general available function files.

3. Application structure generator implementation simulation

In order to simulate an implementation of the generator logic above, assume the existence of a portal of management applications created on web platform using programming languages PHP, JavaScript, HTML, XML files and MySQL database engine. The portal has the directory structure and files as presented in Figure 2.

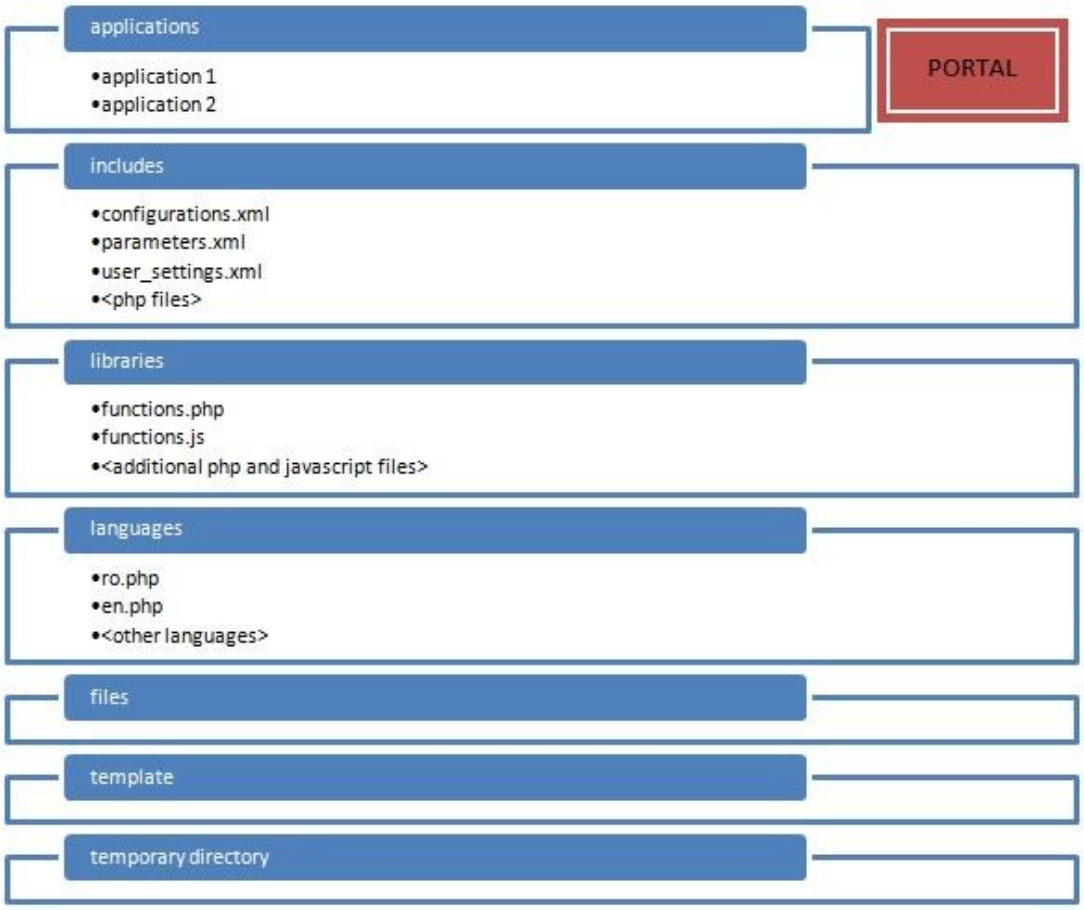

Figure 2. Portal structure of directories and files.

All directories and files shown in Figure 2 contribute only to the operation of the portal as an integrated system.

When the need arises for a new application, *application 3*, using generator structure, this new application will be integrated in the *applications* directory of the portal, with the structure shown in Figure 3.

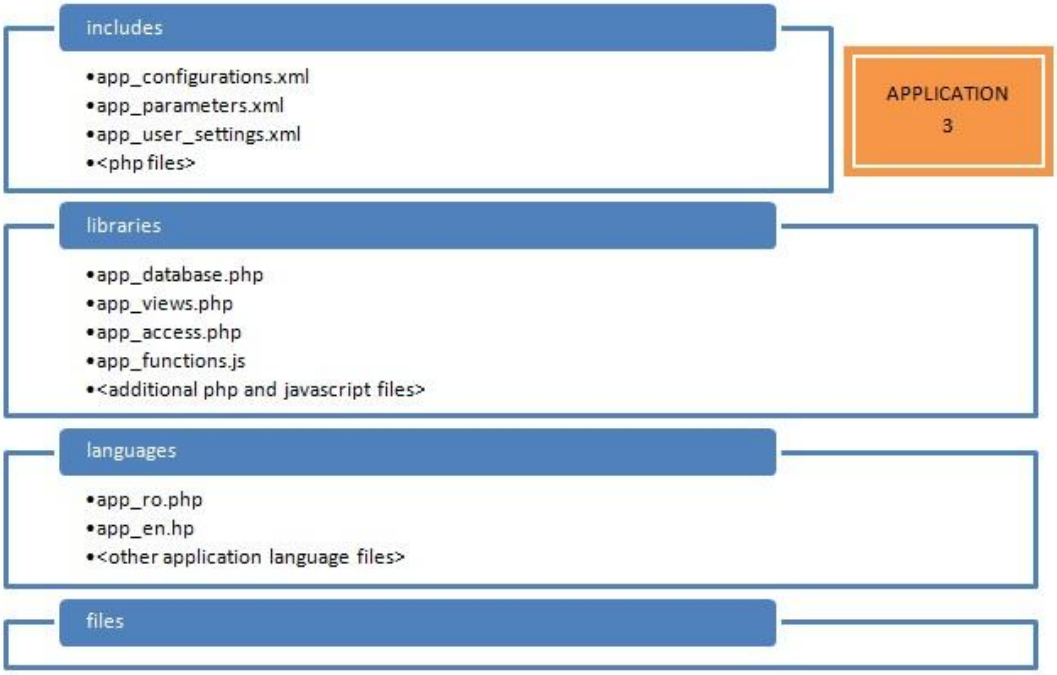

Figure 3. Application structure of directories and files.

4. Conclusions

Using this way of structuring management applications, the portal can be an integrated one, reliable for a long time. Going forward with this logic, the portal can also be considered as an application, with the same structure that it also fulfils its specific functions, those of ensuring support for incorporating new information and integrating new applications.

Bibliography

[1] http://www.onextrapixel.com/2011/12/12/building-web-apps-file-and-directory-structures/

[2] http://framework.zend.com/manual/1.12/en/project-structure.project.html

ACKNOWLEDGEMENT: This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), ID137516 financed from the European Social Fund and by the Romanian Government.