Faceless Componentization Versus Componentization with Assistance Class in Web Dynpro ABAP

M. Osaci*, A.D. Berdie*, G. Prostean** and A.A. Pop***

* “Politehnica” University of Timisoara/Faculty of Engineering Hunedoara/Department of Electrical Engineering and Computer Industry, Hunedoara, Romania
** “Politehnica” University of Timisoara/Faculty of Management in Production and Transportation/Department of Management, Timisoara, Romania
*** “Politehnica” University of Timisoara/Faculty of Automation and Computer Science, Timisoara, Romania

mihaela.osaci@fih.upt.ro, a_berdie@yahoo.com, gabriela.prostean@mpt.upt.ro, popika_88@yahoo.com

Abstract — This paper analyzes, by case study, the similarities and differences between two existing componentisation techniques of a project in Web Dynpro ABAP: the faceless componentisation and assistance class componentisation, made possible due to the MVC architecture (Model-View-Controller), which are the basis of the Web Dynpro ABAP framework. We highlighted the similarities and differences, advantages and disadvantages.

I. INTRODUCTION

To meet the challenges of contemporary competitive business environment, the best companies ensure transparency in all the aspects of their business, enabling them to react quickly, owning best quality, efficient and flexible information.

The study case project presented in browser a catalogue of hardware with product information, capability of online ordering of the products, offers and news about computers. Fig. 2 shows a homepage capture.

The Web Dynpro for ABAP [2], [3], [4], [5], (shortly called WD4A or WDA) is the SAP current standard for the Web development in ABAP environments. It contains a runtime environment and a graphical development environment with Web Dynpro tools that make up a special framework, integrated already within the ABAP Workbench. The WD4A applications use a variety of technologies, such as: HTTP, HTML, CSS, XML, and client-side scripts behind any Web application.

The Web Dynpro component - Fig.3 – [2], [3] is the fundamental programmable unit, and the Web Dynpro application is the fundamental executable unit.

With the integrated software platform SAP NetWeaver – Fig.1 [1], the SAP applications allow the companies worldwide to improve the relations with their customers and partners, to streamline their operations, and to obtain significant benefits within the entire organization [2].
MVC stands for Model-View-Controller [6] and it is a software design pattern (pattern, structure, and model) that enables decoupling of the data model from the visual part (user interface).

The controller is an intermediate component that facilitates the communication between them. Any framework works on the MVC principle. However, in case of a Web Dynpro framework [2], [3], [4], the design data separation is reflected by the objects producing data and the data consuming objects. Meanwhile, the entire structure is designed with components.

The project componentisation is the project split in several components, according to the Model-View-Controller paradigm.

II. FACELESS COMPONENTIZATION

According to the faceless componentization paradigm [7], the Web Dynpro project consists of some Web Dynpro components linked in a structure that contains three basic components (controller component, model component and view component). The model component is the faceless component that becomes the logical part of the project, the view component contains the visual part of the project and the controller component, which contains also the executable application, links the model component and the view component. The component controller uses the model component and view component.

From the technical point of view, to realise a project using the faceless componentization technique involves using in the implementation the concepts of usage, internal and external mapping, component instantiation and decoupling of the data model from the visual part [7], [8], [9], [10]. In the literature not exists a comparative study relative to techniques componentization that can be used in Web Dynpro.

III. COMPONENTIZATION WITH ASSISTANCE CLASS

When using the componentization with assistance class [4], the model component and controller component disappear, and the logical part of the project is taken over by a global assistance class - Fig. 4. This class inherits the SAP class cl_wd_component_assistance [4] and is automatically instantiated in the project components which are used. The implemented methods of this class will form the logic part of the project.

IV. ANALYSIS OF FACELESS COMPONENTIZATION VERSUS COMPONENTIZATION WITH ASSISTANCE CLASS

The differences between the two techniques are presented in Table 1.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Faceless componentization technique</th>
<th>Componentization with assistance class</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of project components</td>
<td>Components project + three components in componentization structure</td>
<td>Components project + a component for global view</td>
<td>Advantage componentization with assistance class</td>
</tr>
<tr>
<td>Logic of the project</td>
<td>Component Controller of Faceless Component used in Controller Component</td>
<td>Assistance class instantiated in all components of project when using</td>
<td>Equality</td>
</tr>
<tr>
<td>Methods in logic part of the project</td>
<td>Can be implemented with the logic of the visual part</td>
<td>No supports logical of the visual part but working with the data model</td>
<td>Advantage faceless componentization</td>
</tr>
<tr>
<td>Decoupling data model of the visual part</td>
<td>Complex procedure</td>
<td>Easy procedure</td>
<td>Advantage componentization with assistance class</td>
</tr>
<tr>
<td>MVC concept</td>
<td>Very well reflected</td>
<td>Component view must keep the interface logic</td>
<td>Advantage faceless componentization</td>
</tr>
<tr>
<td>Context nodes</td>
<td>It is possible the visibility of the context nodes in both directions (the components that are used to components used and vice versa)</td>
<td>It is possible visibility of context nodes only in one direction (the components that are used to components used)</td>
<td>Advantage faceless componentization through visibility nodes</td>
</tr>
<tr>
<td>Mapping nodes</td>
<td>Internal and external - complex procedure</td>
<td>Internal mapping - easy procedure</td>
<td>Advantage componentization with assistance class by easy procedure but disadvantage in visibility nodes</td>
</tr>
<tr>
<td>Membership executable web dynpro application</td>
<td>Component Controller</td>
<td>Component for global view</td>
<td>Equality</td>
</tr>
<tr>
<td>Errors Debugging</td>
<td>Difficult for projects with many data nodes</td>
<td>Less difficult</td>
<td>Advantage componentization with assistance class</td>
</tr>
<tr>
<td>Maintenance project</td>
<td>Difficult for projects with many data nodes</td>
<td>Less difficult</td>
<td>Advantage componentization with assistance class</td>
</tr>
</tbody>
</table>

![Figure 4 Global assistance class](image-url)
Some of the similarities between the two techniques are: the teamwork possibility, in both techniques the model is decoupled from the visual part (the componentization faceless reflects better the Model-View-Controller paradigm), in both componentization techniques we can use and reuse components and last but not least, both techniques can be used to develop complex projects.

V. ADVANTAGES AND DISADVANTAGES

As common advantages of these componentization techniques, we mention: the teamwork possibility of the opportunity to realise large and complex projects. In the faceless componentization technique, the biggest disadvantage is the extensive external and internal mapping procedure of the data nodes that can cause errors difficult to troubleshoot in the projects rich in data nodes. But this disadvantage is compensated by the full visibility of the data nodes in the project. Another disadvantage of the faceless componentization technique is the laborious procedure to decouple the data model from the visual part. This disadvantage is however compensated by a clearer reflection of the Model-View-Controller concept, i.e. the methods of the model component accept the logic of the visual part.

Some disadvantages of the assistance class componentization are: the one way visibility of the nodes in the project, the weaker reflection of Model-View-Controller concept, because the assistance class methods do not support the logic of the visual part, but the advantages, i.e. less components, easy procedure to decouple the data model from the visual part, error debugging, and easy maintenance of the project, compensate the disadvantages.

The execution time is important information about the project performances. The execution time can be tested with the ABAP Runtime Analysis tool – Figs. 6, 7. The test results can be saved in graphics files on the Application Server. These files can be used to realise the necessary optimizations.

Fig. 5 Transaction

To test the Web Dynpro applications made with these two techniques, we can realise transactions for each of them – Fig. 5, to be used to run the web dynpro applications from the ABAP Runtime Analysis tool window - Figs. 6, 7. The test results for the project implemented through the two techniques are presented in fig. 6.

From the graphic - Fig. 8, we can read, in microseconds, the execution time of the ABAP instructions, the time required to work with the database and the operating system working time. You can see that if the componentization is made with assistance class, the total execution time is lower than in case the componentization is made with the faceless technique.

![ABAP Runtime Analysis: Initial Screen](image)

Fig. 6 ABAP Runtime Analysis tool – Testing the application using assistance class componentization

![ABAP Runtime Analysis: Initial Screen](image)

Fig. 7 ABAP Runtime Analysis tool – Testing the application using faceless componentization

![Results of project testing with ABAP Runtime Analysis](image)

Fig. 8 Results of project testing with ABAP Runtime Analysis
VI. CONCLUSIONS

In this paper, we carried out a comparative analysis of two current techniques for Web Dynpro ABAP: the faceless componentization and the componentization with assistance class. Thus, we were able to highlight, on a case study, the similarities and differences between these techniques. As a summary, both techniques showed advantages and disadvantages, but if we choose a technique to realize a large project, complex and very rich in data nodes, the balance tilts towards the assistance class componentization technique.

REFERENCES