Do you know Composite Designer? (SAP NetWeaver Composition Environment)

Applies to:

SAP NetWeaver Composition Environment 7.10 EhP1 and higher.

Summary

This paper gives a short introduction into the Composite Designer which is a new perspective in the NetWeaver Developer Studio. Its intention is to simplify development of composite applications by providing a better overview about the application as well as a tight integration between different tools already available in the Developer Studio. Therefore, the most important features of Composite Designer are highlighted and the usability improvements that come along with it are outlined.

Author: Stefan Henke

Company: SAP AG

Created on: 19 October 2009

Author Bio

Stefan Henke is a Software Developer at SAP AG. He was already working on several projects in the NetWeaver area. In his current project, he is working on the NetWeaver Developer Studio.

Table of Contents

What is it about?

Are you developing applications using NetWeaver Composition Environment? Yes? Have you ever heard about the Composite Designer? No? Then you should really take a look at it to simplify your development. If you are even not using CE so far, maybe the Composite Designer will be your first contact with it. It will be definitely worth trying it.

The Composite Designer is a new perspective in the NetWeaver Developer Studio that targets developers whose applications involve more than just one technology. In previous releases, it was hard to keep the overview about the application as there were only development perspectives available that targeted a certain technology (e.g. J2EE, Web Dynpro or Process development). To put it into other words: The model of your application was only present in your mind. To overcome this, Composite Designer has been introduced with CE 7.10 EhP1. For example, it allows you to visualize all components of your composite application in a cool graphical editor. But there are a lot more new things especially when it comes to relations and dependencies between components from different technologies. You will see all this in the following chapters.

Functionality of Composite Designer

Let´s dive into the Composite Designer and take a detailed look at the features it offers. First of all, the Composite Designer introduces the so called Composite Designer overview which is a graphical editor showing all major elements and relations in your application. This gives you a very good overview about the structure of your application and especially the dependencies between certain technologies. Technically spoken, Composite Designer uses a simple model to visualize your application. Thereby, the model consists only of elements and relations between two elements. Each technology that is integrated into the Composite Designer is able to visualize its entities using this model. The big benefit of this approach is to make it much easier to understand your own application.

Note: Composite Designer does not persist any of the information displayed in the editor. The model is completely transient and always filled with the data stored in the domain specific files (e.g. java files or metadata files).

An example of such an application is shown in Figure 1. The screenshot visualizes all elements of it. The type of each element is indicated by an icon in the left upper. But this is not all you can see or do with this element. If there is an error or warning present in the problems view for this element, this is indicated by the corresponding icon in the left lower corner. There are some examples shown in the screenshot. Take a closer look at the Business Logic layer and you will find them easily. By selecting one of the elements, all ingoing and outgoing relations of it will be visualized by drawing lines between the corresponding elements. This allows you to easily check the dependencies of an element. One could think that it is just a small nice feature, but it is much more: Relations in Composite Designer are evaluated even across the border of a single technology. In previous releases, this was a pain point as there was hardly any support for visualizing cross technology relations. If you were using the Developer Studio in the past, you will definitely like this. Especially if it comes to verification of relations, Composite Designer has a lot of strength. The tool is evaluating relations in the background and notifying the user if there are any problems present. One example can be seen in Figure 2. The relation from the Web Dynpro component “CComplete” to the CAF Application Service “TimeOffService” contains some kind of error. It is possible to get more information about this error by simply calling the corresponding action from the context menu available on the link.

Note: It is possible to switch on/off displaying the relations between elements. There is a toolbar button for this. So, if you don´t see any relation though you expected one, this should be the first place to check.

By using the palette on the right side of the editor, it is possible to create new elements in the application as well as dragging new relations between two existing elements. This makes it possible to create the building blocks of your application top-down. This opens the door for new ways of developing the application. Simply think about a development team which splits up the responsibilities along the technology stack (this means that dedicated persons are responsible for each technology). In such a scenario, this allows you to create the building blocks first in a controlled way so that the individual pieces fit together and later on work in local groups to implement the elements. If you are interested in more details about the creation capabilities, please continue reading, there will be much more details about this in the next chapter.

The Composite Designer does not provide editing capabilities for elements. You still have to do this in the respective technology perspective. This was done by intention as Composite Designer does not want to replace existing perspectives. But there is very well an integration available between these two. By simply double clicking an element, you will directly jump into the corresponding editor that is provided by the technology for this specific type of element.

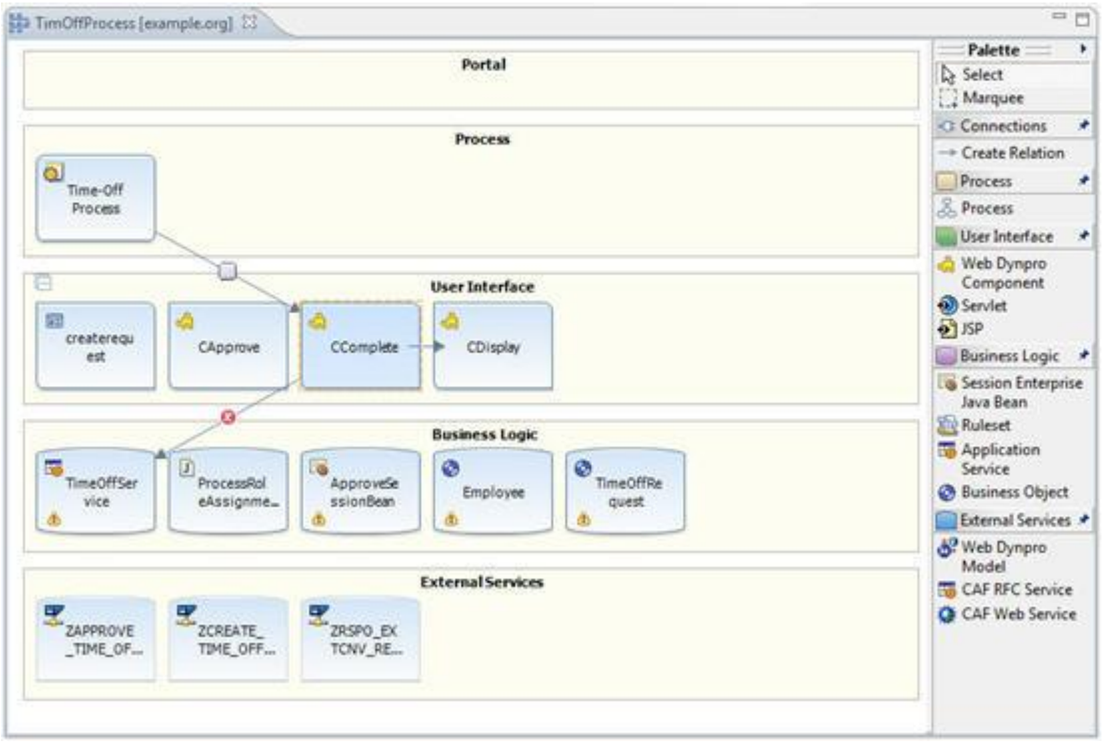

Figure 1: Composite Overview showing a sample application

You might already have recognized that each element shown in Figure 1 is displayed in a particular labeled box. This is because Composite Designer arranges all elements in layers. This has been done in order to emphasize and underline the paradigm of composite applications which includes a well defined separation of layers. It is worthwhile to mention that this separation also introduces the fact that you should only have relations from top to down (for example, a Business Logic element should never depend on a User Interface element). A typical composite application consists of the following layers (from top to bottom):

- Portal
- Process
- User Interface
- Business Logic
- External Services

If you already reviewed the screenshot in Figure 2 in detail, you maybe wondered about the relation from the Web Dynpro Component “CApprove” to the Session Bean “ApproveSessionBean”. In Web Dynpro you normally have to explicitly import any business logic component like an EJB as a Web Dynpro Model in order to call the logic from your UI. But this relation is drawn directly between the Web Dynpro Component and the EJB. To put it short: it´s not a bug, it´s a feature! The Composite Designer Overview wants to give a brief overview only and therefore hides complex models if possible. For the user it is more important to first know that there is a relation between these two entities. The Web Dynpro Model is just an intermediate element which is required from a technical point of view, but not from a semantically one. However, Composite Designer allows you to analyze the relations in detail as well. You can visualize all incoming and outgoing relations of one element in the “Relations View” which can be called from one element´s context menu.

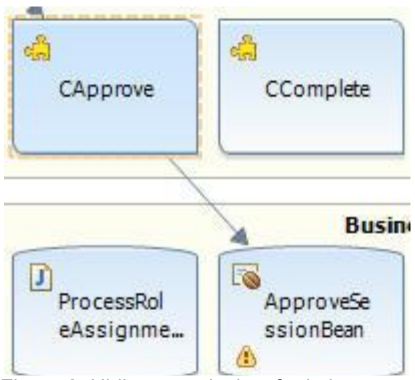

Figure 2: Hiding complexity of relations

Figure 3 is exactly showing such an example. There you can see that there are several intermediate elements present in this relation. Actually, there is not only the Web Dynpro Model in between, but also the java interface of the EJB Session Bean. You can also check the different visualization of usage and implementation relations which are indicated by a solid respectively a dashed arrow.

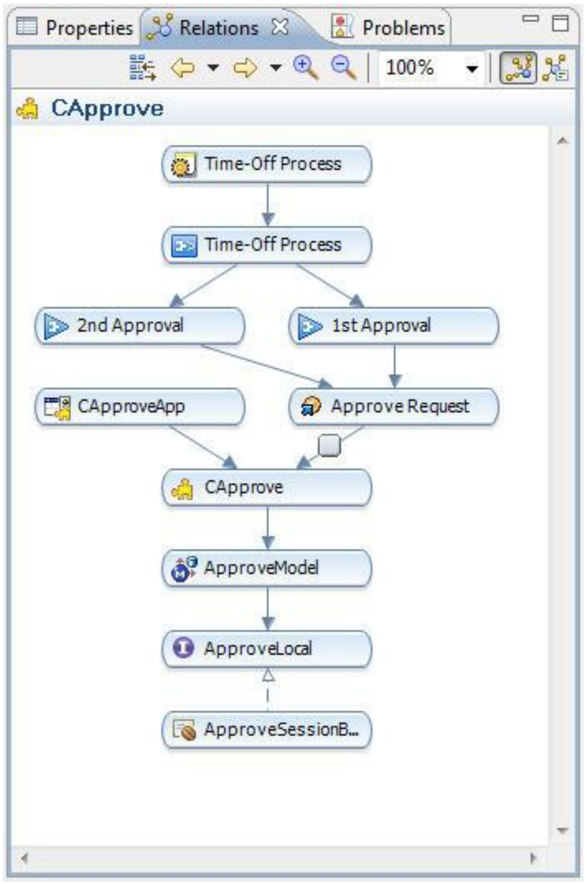

Figure 3: Relations view of Composite Designer

Supported Technologies

In its first release, Composite Designer supports a wide range of technologies that can be used in the NWDS. This includes both technologies invented by SAP (e.g. Web Dynpro, Composite Application Framework) as well as “community” technologies (e.g. EJBs, Servlets).

There are several levels of support: some technologies only allow DC creation and visualization of the content in the Composite Overview. This means that the elements still have to be created in the technology specific perspective, but they will be displayed in the Composite Designer next to elements from other technologies. But most technologies provide a much deeper integration into the Composite Designer by allowing to directly create elements. Therefore, these technologies integrate their wizards into Composite Designer. Of course it is still possible to create such entities using the regular way in the perspective provided by the technology. It´s just another way of creation that is provided, but a homogeneous one across the technology stack of the NWDS.

Note: Even if there are two ways of creating an element or a relation, there won´t be duplicated wizards or editors. Things get re-used to also ensure a common usability. Regardless the way how you created an element, you won´t see a difference in the element visualization in Composite Designer.

Viewing this from an end-user point of view, this integration allows creating a skeleton of your application containing the main building blocks including dependencies between these building blocks. This is a very good way to start creation a component-based application. Of course it is possible to iteratively improve the application later on by adding more elements, deleting certain elements or adding new relations. As the Composite Designer Overview only uses a transient model, you don´t have to fear any inconsistencies.

The following table gives you an overview about the elements that you can create using Composite Designer. All these elements will be listed in the palette which is displayed in the CoDe overview editor.

Technology	Supported element	Layer
Process Composer	Process	Process
Web Dynpro	Web Dynpro Component	User Interface
	Web Dynpro Model	External Service
J2EE	Servlet	User Interface
	JSP	User Interface
	EJB Session Bean	Business Logic
CAF	Application Service	Business Logic
	Business Object	Business Logic
	CAF RFC Service	External Service
	CAF Web Service	External Service
Business Rules	Ruleset	Business Logic

Besides these elements, there are some elements which can be only visualized in the editor like a Guided Procedure or a Visual Composer Model. However, this brings a benefit for users as these elements are as well part of the application. The reason why these technologies don´t provide a higher degree of integration is basically that these tools do not provide an NWDS-based design time, but only a server based one which makes a better integration much more complex or even impossible. Naturally, the ideal world would be to improve this situation. Let´s see what the future brings in this area.

By dragging a new relation between two elements in your application, it is possible to create new usage dependencies between these elements. This works very smooth for the user as Composite Designer will display a wizard which was specifically designed for the selected element types. This wizard will hide a lot of complexity from the user as the relation creation action also covers steps which you had to be done manually if the relation was created using the tool specific wizard. Such steps for example include public part creation and configuration, DC dependency creation or simply execution of validation steps if the elements are in a state to be used in such a relation. Though Composite Designer is doing such steps behind the scenes without user interaction, it still wants to document what it is doing. Whenever you create a relation, the first page of the wizard will list a set of steps that will be executed by the wizard in order to create the relation. This allows understanding how Composite Designer is working in a quite nice way.

In the EhP1 release, not all relation types (combinations between element types) are supported. But again, let´s see what the future brings.

Turn Your Application into a Product

The NetWeaver Developer Studio leverages the component model which consists of Software Components (SCs) and Development Components (DCs) to organize an application from a lifecycle management point of view. As part of the Composite Designer, this model is slightly enhanced by introducing the notion of a Product. If you are familiar with the System Landscape Directory (SLD), you might know this term already. A product is basically meant to group a number of SCs which built up a running application. In previous releases, there was no way to build relations between SCs with the semantics that they belong to the same composite application. As a consequence, there was no appropriate visualization of this fact in the Developer Studio.

Introducing the Product as a new entity, this has changed. As you can see in Figure 4, the Composite Explorer is showing the Product “TimeOffProcess” as root element in the tree. On the next level, there is one container element for each group (layer) of the application. Inside these containers you will see all projects (respectively DCs) which belong to the application. In a real-world scenario, it is common and very likely to spread your DCs across multiple SCs for software lifecycle reasons. SCs are the level of granularity that can be shipped in form of a Software Component Archive (SCA). From this, you can derive that you have to ship the content of a whole SC if you want to deliver a bug fix. If you decided to do so, all contained DCs (even across the border of SCs) are displayed under the umbrella of the Product.

If you are not yet familiar with the component model, you will find some useful links in the section “Related Content” which links to corresponding documentation and tutorials for the NetWeaver Development Infrastructure (NWDI).

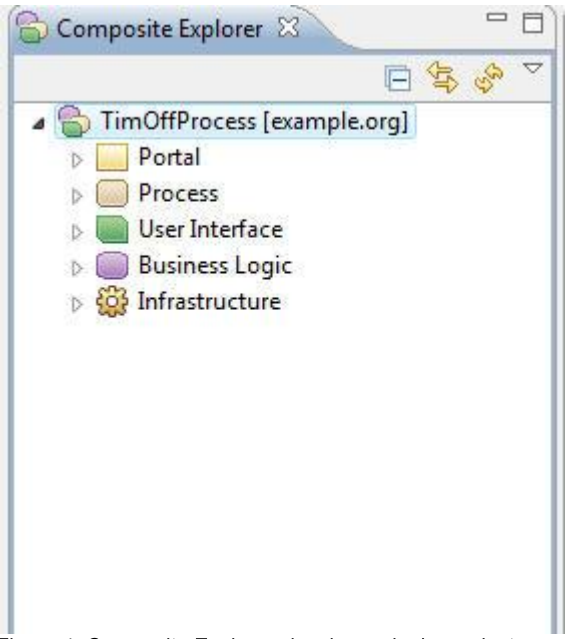

Figure 4: Composite Explorer showing a single product

Outlook

If we made made you craving for trying out what we described in this article, we have the right things for you here (Link to evaluation copy). For the next release, it is planned to increase the number of technologies that support Composite Designer. This is especially true for element and relation creation. So, you are good advised to keep an eye on Composite Designer and start working with it. You will find everything that is described in this article if you launch the NWDS and open the perspective called “Composite Designer”.
Have fun and happy composite designing ;-)

Related Content

Find more information about NetWeaver Composition Environment 7.1 EhP1 on SDN:

SAP NetWeaver Composition Environment 7.10 EhP1 Evaluation Version

Sample Center for SAP NetWeaver CE 7.1 EhP1

SDN area for NWDI-related parts

Copyright