

# EDM EMM Overview: An Extensive Software Solution Partner Showcase for Modern Composite Application Development

## Applies to:

SAP NetWeaver 2004s SP10 – IT Scenario “Developing, Configuring, and Adapting Applications”, IT Scenario Variant “Creating Composite Applications” [see [SAP NetWeaver 2004s Installation Master Guide](#) (requires SAP Service Marketplace login)].

## Summary

The document at hand is intended to demonstrate a typical SSP/partner showcase in the area of composite application development: Integrating your own non-SAP solution with SAP through means of composite applications. It looks into all steps from business case to development process to enable any reader in a similar situation to apply the procedure to his/her own scenario.

**Author(s):** Hans-Joachim Odlozinski

**Company:** axentiv AG

**Created on:** November 13th, 2006

## Author Bio



Hans-Joachim Odlozinski is an expert in SAP application development architectures at axentiv AG, a German SAP service partner. He looks back on many years of development and consulting both internally at SAP as well as for customers & partners in the field. He is currently working as Solution Architect at the Market Development Engineering team of SAP Platform Ecosystem. At axentiv AG he is team manager of Architecture & Development.

## Table of Contents

Introduction.....	3
Defining the Business Case.....	4
Designing Your Process.....	7
Engineering Your Solution Architecture.....	9
Preparing for Development.....	10
Setting Up the Development Process.....	12
Delving Into the Details.....	16
Related Content.....	17
Copyright.....	18

## Introduction

The EDM<sup>1</sup> Event Materials Management (EDM EMM) demonstrates a typical scenario within the SAP ecosystem:

Let us assume that an SSP (Software Solution Partner = typically Independent Software Vendor or System Integrator) has already developed a solution based on some non SAP-specific platform (J2EE, .Net, etc.). The SSP gains feedback from the market that SAP customers would go for his solution if a tighter integration with SAP, both in technology and business processes, would be possible.

The technology integration would require at least an integration of the solution's user interface (UI) in the customer's SAP Enterprise Portal. A J2EE-based solution (which is often the case) should preferably be operable on the SAP Web Application Java to save licensing costs and additional operational costs (infrastructure and knowledge). The business process integration often requires the possibility of integrating the SSP solution into the major process domains of ERP (SD, FI, HR, etc.)<sup>2</sup>. Combining both UI integration and process integration, this consistently leads to the concept of Composite Applications as solution architecture.

It was our goal to give SSPs and partners and of course everybody else in the community an entry point for solution design and implementation on the SAP NetWeaver platform, we decided to design and implement this demo showcase at hand. We hope that this demo showcase will give you both, a horizontal overview and a selective deep vertical insight into the SAP NetWeaver platform.

The document at hand intends to give you an overview about the business aspects, the architecture, and the development process that guided us in implementing the demo showcase. We will discuss the reasons that lead us to the decisions made on all levels of design and implementation. The implementation details are given in separate documents (How-to Guides) which are referenced appropriately.

---

<sup>1</sup> EDM stands for Educational Data Model. EDM is a standard Demo Application provided by SAP Product Management which is used here as a fictional SSP application. The application archive is shipped with this document.

<sup>2</sup> ERP – Enterprise Resource Planning; Application set to manage enterprise-wide resources (here: mySAP ERP).

## Defining the Business Case

The business case is based on the fictional assumption that the standard example application EDM (Education Data Model – see previous footnote) based on J2EE has been developed and already shipped to customers by some fictional SSP *EDM Inc.*

The SSP has identified the business problem according to the standard scenario mentioned in the introduction and derived a high-level composite idea that is assumed to deliver value to the customer. The executive summary of the scenario reads as follows:

### Executive Summary



**EDM Event Materials Management**

**Business problem**

The Educational Data Model EDM is an SSP solution offering extensive Training Management including course definition, course event and participant management. In addition to the standard processes it proved to be a customer requirement to be able to manage required materials such as moderation material, training books, beamers, flipcharts, etc. based on SAP ERP MM

initiate the procurement through an internal or B2B purchasing process via SAP ERP standard interfaces (Enterprise Services)

**Composite Idea**

EDM EMM is complementing the current solution for Training Management (EDM), by adding the management of event materials along with the triggering of the downstream procurement process of these materials.

In the first release, EDM EMM will enable the event planners to select materials from a SAP ERP MM based material catalogue and to assign them to a given course event.

The assigned materials list (modeled as a purchase requisition) is then routed through a two-step approval process until it is finally converted into an ERP purchase order.

**Value to Customer**

EDM Event Material Management (EDM EMM) delivers tangible benefits as:

- Efficient procurement of additional course event materials
- Controlled process via role-based approval workflow
- Direct B2B procurement for customers without internal materials and procurement management

The business scenario itself as well as the value to the customer sounds pretty plausible and we will not further discuss it here. What is more of interest is the reasoning that brought the SSP to the idea to implement this solution as a composite application. As you will see, this is not simply a business decision as it affects the general solution architecture, its release handling and the overall requirements to the solution development (without talking about any implementation details). Thus, the decision was taken jointly by business and IT executives.

The SSP was facing two basic alternatives during the decision process:

Simply extend the existing solution in form of a new release.

Implement the extension as a composite application stretching across the EDM solution and ERP leaving the existing solution unchanged.

Let us look at possible Pros and Cons of these two alternatives in the following table:

Alternative	Pro	Contra
New EDM Release	<ul style="list-style-type: none"> <li>Use of existing development infrastructure</li> <li>Established delivery channel to customers</li> </ul>	<ul style="list-style-type: none"> <li>Existing Customers are forced to upgrade</li> <li>No UI-based integration into SAP NetWeaver (EP<sup>3</sup>)</li> <li>Requires workflow implementation in existing solution (currently not available)</li> </ul>
Composite application	<ul style="list-style-type: none"> <li>Simple Add-On, no upgrade for existing customers</li> <li>UI-based integration into SAP NetWeaver (EP<sup>3</sup>)</li> <li>Available “workflow engine” through Guided Procedures</li> <li>Technology Upgrade including marketing effect [“Powered by SAP NetWeaver”]</li> <li>Consequential decoupling of service composition from service implementation and backend versions</li> </ul>	<ul style="list-style-type: none"> <li>New SAP-specific development infrastructure</li> <li>New model driven programming model</li> <li>Requires SAP NetWeaver infrastructure on customer site</li> </ul>

The decision is now driven by the weighing of these Pros and Cons in the lights of the targeted customer segment. The following fictional statements reflect the possible reasoning:

“Our target customer segment is that of SAP customers as SAP ERP integration is required anyhow. The expected small B2B segment is considered negligible. “

“If they are already our customers for the existing EDM solution, we won’t bother them with an upgrade of the productive version but give them a modern solution which is based on their familiar SAP NetWeaver platform. The easy SAP Enterprise Portal integration can be sold as a nice give-away for those who have a portal installation.”

“Considering the technology upgrade for our development team and the potential marketing effect through offering a *Powered by SAP NetWeaver* solution the investment in education and development infrastructure are considered manageable.

Anyhow, as discussions with SAP’s ecosystem experts from Market Development Engineering showed, the required development efforts will be pretty small (~ 30-40 md) and the required infrastructure will be cheap due to the small required development team.”

<sup>3</sup> EP – SAP Enterprise Portal

It is by following this line of argumentation that the SSP EDM Inc. decided to implement the solution as a Composite Application based on SAP NetWeaver. Given different boundary conditions, this could have of course looked fairly different.

 **The Pros and Cons table as well as the line of reasoning stated above are fictional!** Nevertheless do they reflect our experience from the field and represent rather a majority of scenarios than a minority. If you are in the situation to apply our experience to your own business case, please make sure that you derive your own Pros and Cons table and line of reasoning that is necessary and sufficient in your company's context.

With the decision being taken we will now go deeper into the implementation and will closely look into the process design.

## Designing Your Process

The assigned Business Process Expert (BPX) of EDM inc. had to base the process design on a definition of and relations between the underlying business objects that are sufficient from the Composite's view:

A Course is the abstract definition of educational content (description, objectives) along with additional data such as the course duration.

An Event is to be interpreted as a course event, i.e. an actual course being held in a given timeframe which can be attended by participants.

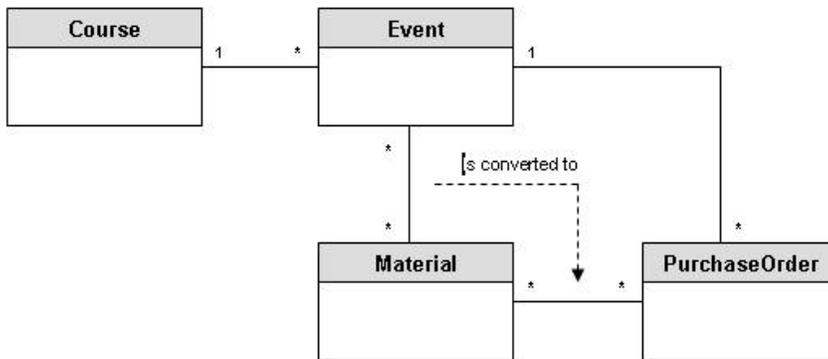
A Material is a defined product in a purchasing or sales process.

A Purchase Order is a business document that is sufficient for invoking a purchasing process for a set of defined materials (along with all required data such as the ordered quantity per material).

For a given Event at a given time it is possible to assign Materials in arbitrary amounts as additional course materials

The assigned Materials at a given time can be finally converted into a Purchase Order. For any given Event an arbitrary amount of Purchase Orders may be created.

The following Business Object Model results:

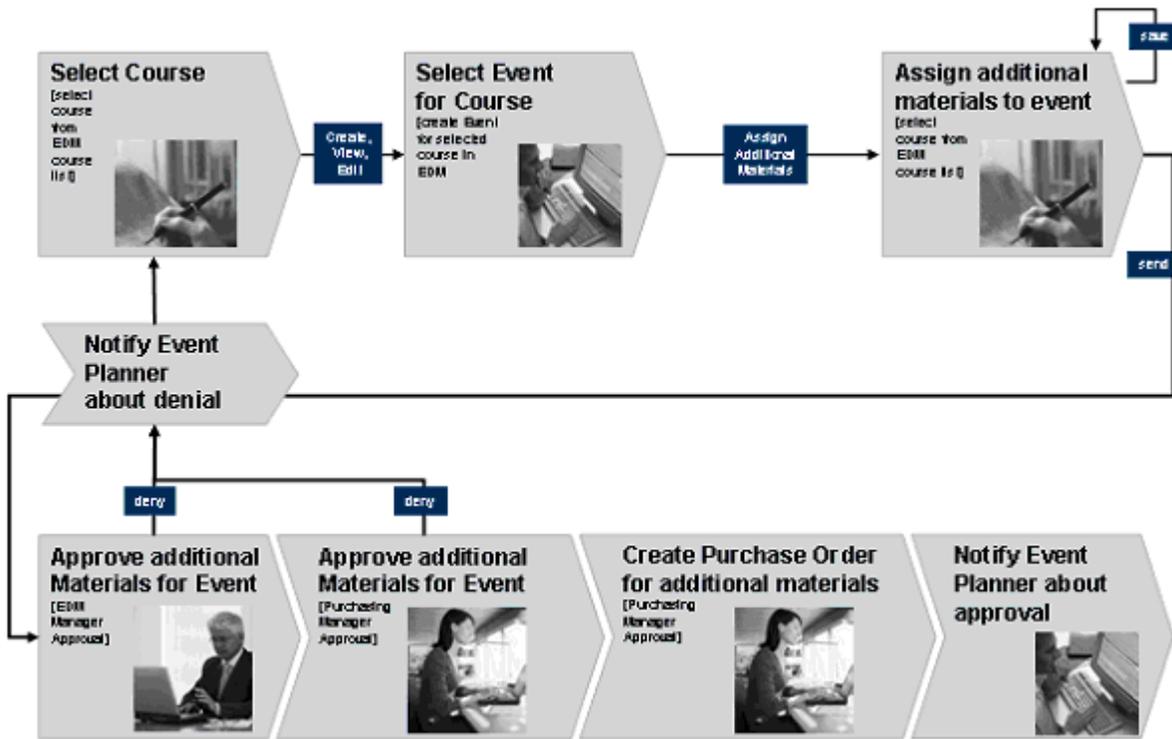


To control the process and to avoid the actual “arbitrariness” of ordered amounts and numbers of Purchase Orders per Event, the BPX decided to run the process of converting assigned Materials into Purchase Orders through a two step approval process:

A user with the Role *EDM Manager* checks the additional materials list for its meaningfulness from the educational point of view and for its validity in conjunction with the participants count. Result: Approval or rejection.

A user with the Role *Purchase Manager* checks the additional materials list for its expected costs. Result: Approval or rejection. The approval is final and results in the creation of a Purchase Order.

The complete process is depicted in the following informal flow diagram:



The first three steps are processed by a *Event Planner* who is provided with an application to assign the required materials to a selected Event and to start the approval process.

Based on this (admittedly coarse) process picture, the BPX hands over to the Solution Architect to derive the (equally coarse) Solution Architecture based on SAP NetWeaver.

## Engineering Your Solution Architecture

The Solution Architect has in mind that the Enterprise SOA approach of SAP suggests a service-based integration of all components. On a technological level this seems to imply the utilization of Web Service interfaces. But there are two problems immediately identified:

SAP Enterprise Service Definitions are not yet available for the customers of EDM Inc.

The customer base resides and will continue to reside on current and older SAP ERP releases. ABAP Web Services are only available from base release 6.40 onwards so for older releases the old BAPI interfaces would need to be utilized.

The Solution Architect triggers a call back to the management to request a decision on which priority is given to the set of available backend base releases. As the term "Composite" suggests a modern approach, the management decides to focus on the releases that allow for ABAP Web Service integration (base release 6.40+). This focus is supported by a customer base analysis done by marketing showing that pre-6.40 base releases play only a minor role. If there are - after market announcement of the composite – critical customer requests for support of older releases, the management will revisit the current decision.

As it is the management goal to engage with the "Powered by SAP NetWeaver™" program, there are two basic certification paths that seem to apply:

J2EE Deployment and SAP NetWeaver Portal Business Package (Integration scenarios [J2EE-DEP](#) + [EP-BP](#)).

SAP NetWeaver Composite - Composition Tools Based (Integration scenario [NW-CA-CTB](#))

Bringing the desired technology upgrade into play, the Solution Architect decides to follow the following action path

The composite application will be a SAP NetWeaver Composite - Composition Tools Based using Visual Composer as UI Technology, Guided Procedures (GP) for the approval process and Composition Layer for the service integration layer.

The Event Planner application will be realized as a separate Visual Composer application that triggers the following Guided Procedure.

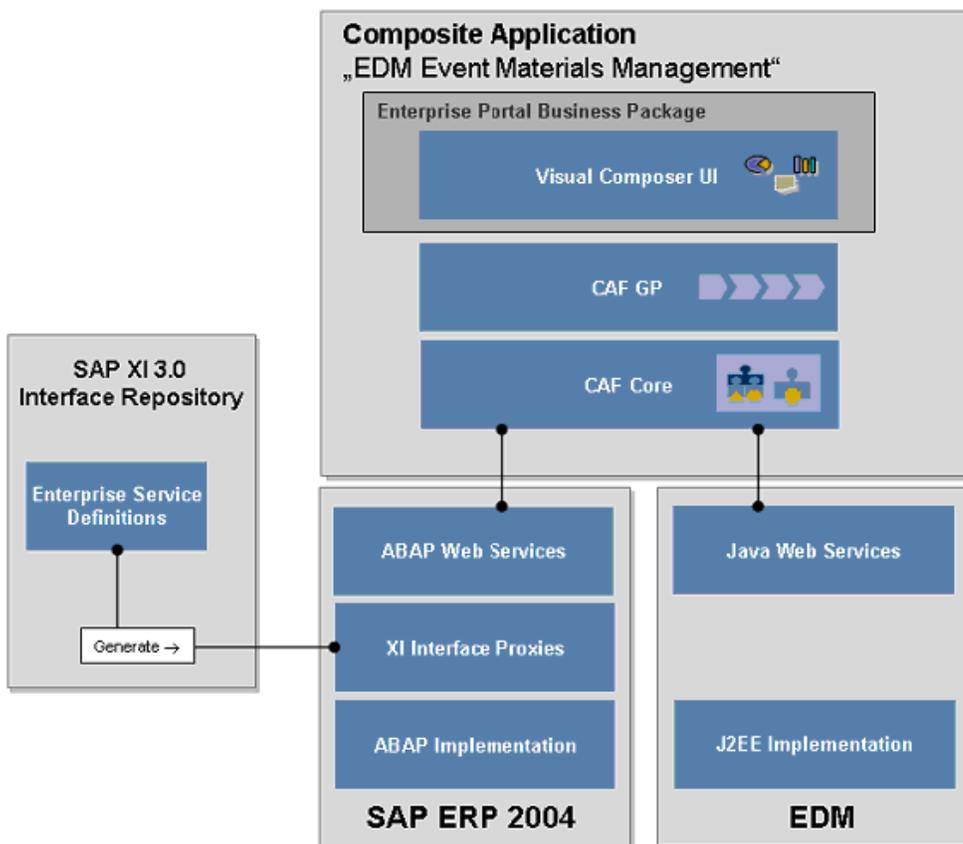
The composite application will be delivered together with a Portal Business Package.

The ERP service definitions will be tailored to decouple the composite application from the specific backend version (current and coming) and customization. They are developed and delivered as XI Content.

The ERP service implementation is done using XI proxy generation and a default ABAP implementation on the ERP ABAP stack followed by exposure of the proxies as Web Services.

As the product EDM is already Web Service enabled, the integration into the composite should be straightforward.

The following picture illustrates the general architecture including the relevant components and tools:



## Preparing for Development

After generally designing the solution it is time to head towards implementation. The Solution Architect hands over to an assigned Development Manager whose first task is to prepare the necessary infrastructure for development. He finds that there are actually quite a few things to consider:

For new customers that want to purchase EDM together with the new composite solution, the EDM product should be able to run on SAP NetWeaver Application Server Java as well. Therefore a migration including certification according to the SAP Integration Scenario J2EE-DEP should be triggered in parallel to the composite development. The SAP documentation offers all required documentation resources (J2EE Migration Kit, SDN Knowledge Center).

A NetWeaver Development Infrastructure (NWDI) should be set up to implement the most appropriate Change Management for SAP composite development. This involves installation of the NetWeaver Application Server Usage Type "NetWeaver Development Infrastructure" [DI] and some remaining post-installation configuration [see NetWeaver 2004s Installation Master Guide]. An estimated 3 day session, maybe with some support from SAP, is considered sufficient as there will be clearly only one NWDI development track and only one Software Component for now.

At least a two-stage NetWeaver development system landscape should be set up (development system, test system). The required usage type EP can be derived from the IT scenario variant "Creating Composite Applications". The also required usage type AS Java is automatic prerequisite EP.

At least one ERP system must be configured and the material master data entered for tests. One system might be sufficient for both development and testing.

The up-to-date Developer Workplace needs to be rolled out to the team. As the team will be pretty small (2-4 developers), this is not a big issue. Installation and Configuration including setup of NWDI connectivity will be finished within one day. Actually, the small number development does not require any large scale NWDI installation at all.

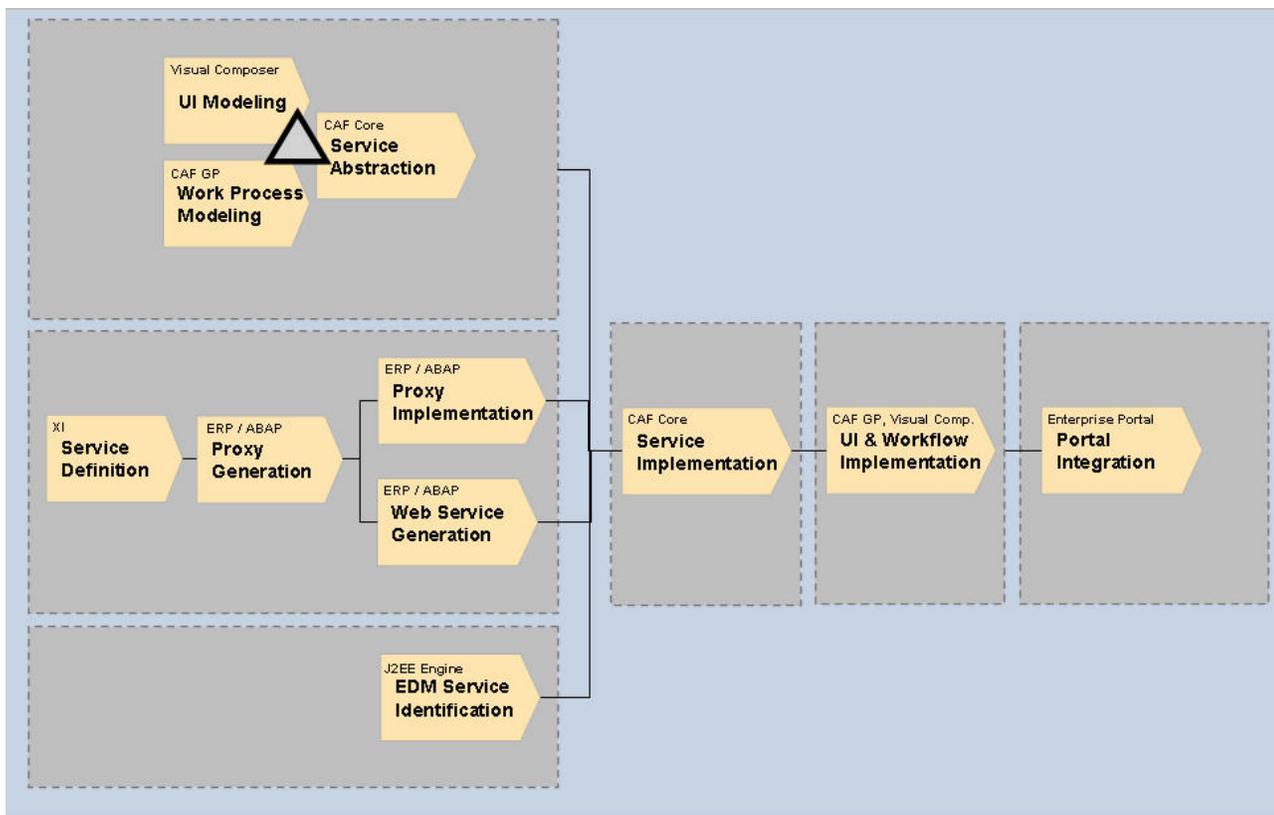
The development team either needs a pretty broad skill set or good support back-up through external experts to be able to deliver all required components.

Let us have a closer look at the expected types of development deliverables:

Deliverable	Details
EDM Application	Original solution migrated to SAP NetWeaver Application Server Java. An import into NWDI is not required!
VC Model	Delivered together with Enterprise Portal Content
BP Content	DC of type "Content → Guided Procedure 2.0"
Composition Services	One or more DCs of type "Composite Application Services"
Enterprise Portal Content	DC of type "Enterprise Portal Content Archive"
SI Content	Service definition (Namespace Export)
ERP Transport	Default implementation of ERP services

This table directly reflects the skill set that is needed, so the Development Manager has to plan a sufficient development ramp-up time to enable his team accordingly.



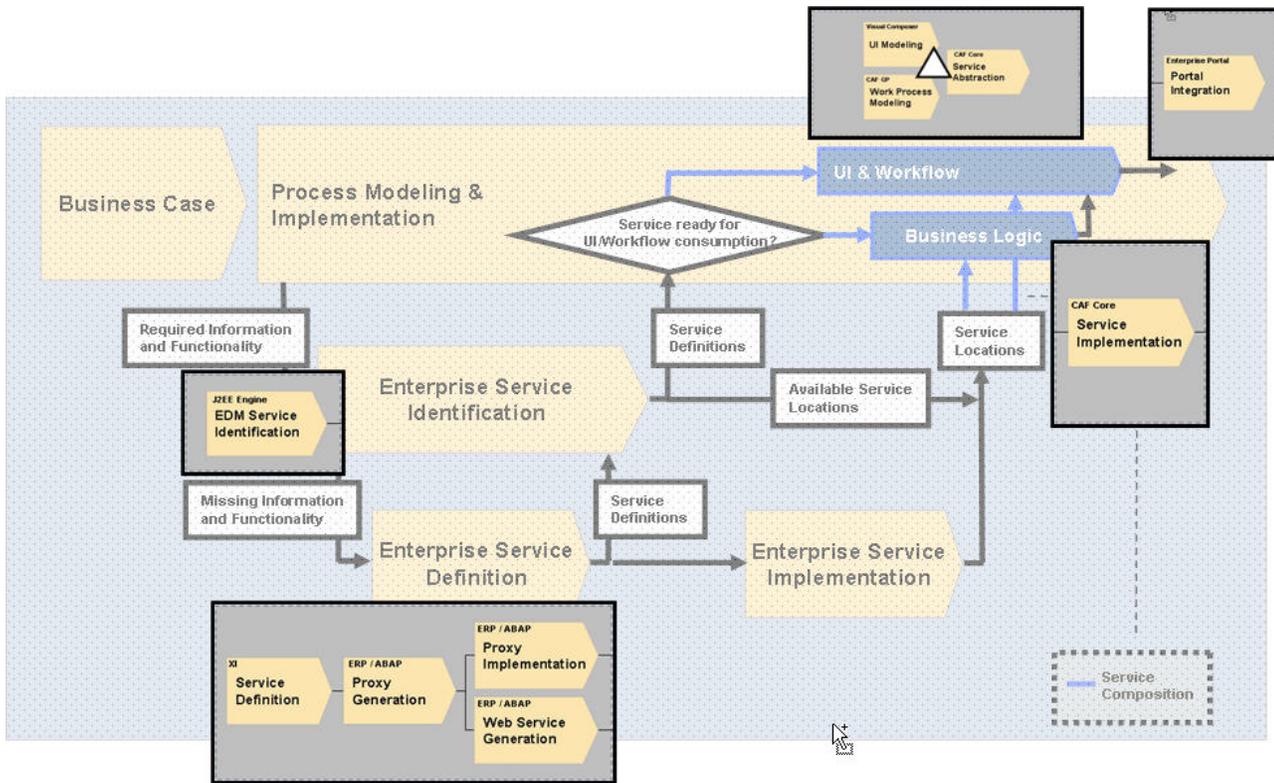


You can see three main parallel threads of work. Note, that they are not at all independent of each other. They operate in different toll domains and typically require different skill sets.

The top thread concentrates on the modeling aspects where **UI Modeling** with Visual Composer and **Work Process Modeling** using GP (down to the level of GP actions) stood at the beginning. During the early modeling phase, required services were identified. This triggered the two other threads which focus on identification and provision of “Enterprise Services” from ERP and the Educational Data Model (EDM). The **Service Abstraction** in the top thread derived and designed required Composite Application Services that aggregate the Enterprise Services to a level that can be consumed by the UI and workflow layer above (Quite often simply due to design tool restrictions). The triangle connecting the three tasks in the top thread is supposed to illustrate the dynamic interaction between them. A simple flow diagram would not be sufficient.

After having completed these three threads, the actual implementation of the Composition Layer was conducted. This practically completed the Visual Composer UI. For testing the UI, we had to simply configure the service endpoints in the EP system configuration. Having all building blocks at hand, the GP process can be finally implemented (Creating Callable Objects and assignment to Actions). The Enterprise Portal Integration (Bundling of EP Business Package) finalizes the development process.

The following graphic depicts where these tasks within the development process are located in the overall development process view discussed beforehand:



Now you might argue: “Nice, but somehow the tricky part seems to be hidden around this nice triangle in the top development thread. How do I really do that? The tools do not suggest a straightforward way.” Again, I have to admit that you are perfectly right and that is why I will delve into this in a bit more detail before we will fork off the discussion into more detailed documents that explain the exact how-to for each task involved: *Our Realistic Modeling and Design Cookbook for Composite Applications with NetWeaver 2004s!*

## The Realistic Modeling and Design Cookbook:

### □1 Getting started

I suppose that it will not surprise anyone that we actually started the whole process in front of a whiteboard with a lot of colored markers at hand. There was no design tool involved at all. We identified the rough components needed: An application that allows assigning materials to events and to initiate an approval process, and the approval process with the final creation of the purchase order. The EDM data model made it pretty clear, what such an application would look like.

Now we could actually use the first design tool: We could design the GP process down to the Action level.

### □2 Where can I get or put my data?

We are back at the whiteboard. The question simply is, where can I get my information and where can I put it during my process and at the end. We identified semantical services, such as `GetCoursesForSearchTerm`, `GetAllEventsForCourse`, `GetAvailableMaterialsList`, `StoreAssignedMaterials`, `CreatePurchaseOrder`.

Now we had everything at hand to start the service identification. We identified the required EDM Web Services and the ERP BAPIs (there was no ERP 2005 with ES packages available at the time!). As Visual Composer and Composite Application Layer still have restrictions concerning the consumable complexity of Web Services, we decided to strictly put a Composite Application Service layer on top of all enterprise services. Besides allowing us to adapt the service layer to these restrictions it allowed for rapid prototyping because we could provide early dummy implementations for the complete service layer with no dependency to EDM or ERP. This is exactly what we did in the phase of service abstraction: We designed all required Composite application service interfaces, provided some dummy implementation and exposed the services as Web Services for design time consumption.

### □3a Model the User Interface components

After completing step 2, this step was actually pretty easy. Actually, it wasn't really because we had to fight with tool restrictions, missing documentation and simply bugs. Do not forget that we started before all the nice SDN documentation was available on the matter. But if you omit this part, it actually was and now is which we could prove by implementing another composite demo application for the banking sector for SAP Teched Amsterdam this year. Anyway, we already had a prototype with a dummy implementation to present to the outer world.

### □3b Model and implement the ERP services

Parallel to Step 3a, we wanted to provide the required ERP services. As we did not want to have any ERP specific logic in our Composition Service layer, we decided to define streamlined interfaces that suit our purposes. These were modeled in the XI Interface Repository and generated as ABAP proxies in the ERP system. The ABAP proxies were first exposed as web services to make them early available for integration in the Composition service layer as external services. Finally, we implemented the proxy glue code that linked the interface to the standard BAPI layer.

### □4 Replacing the Composition Layer dummy implementation

Now, we linked the web service interfaces of the Composition Services to the real enterprise services by replacing the dummy implementation with a real one. A pretty straight forward issue which in its most complex form consists of mapping code from array to `java.util.List` objects and vice versa.

### □5 Implementing the Guide Procedure

With a well (and appropriately) designed service layer at hand, this step is actually pretty straight forward and not that difficult, if you know how to apply the tools correctly. We faced quite a few difficulties in the areas of context mapping, role handling and the general editing and activation process which we fed back into the tool development process. We are confident that you will soon benefit from our efforts.

### □6 Portal Integration

I will not detail on this step either as this is well documented knowledge. SAP Enterprise Portal is a solid and well-documented product. In short, we defined a Portal Role with assigned worksheets, pages and iViews, and integrated GP with the UWL (Universal Worklist). For backend authentication, user mapping needed to be configured.

 The identified services drove in a way our UI design because we had to see which data is actually available and which data should be selected to be put on the screen.

Starting with the detailed UI design might raise expectations that cannot be fulfilled at all or that can only be fulfilled with high additional effort in the enterprise service layer. If the business specialist looks at and tests the first prototype and says: "Hey, there is an important field missing! Without this information I cannot decide with which process step to continue!" then this statement is probably valid and must be considered. Even a business object model change might be required. But if the business specialist says: "I definitely want this and that field on the screen!" before the definition of the business objects and thus the actually available data have been investigated, this statement might not be quite as valid.

## Delving Into the Details

Now that you have a pretty detailed overview across our development process, we do not want to hold you back from delving into the detailed How-to-Guides that describe the implementation details and the information needed to follow our steps. Here, the goal lies more on hands-on, compact information. You will find redundant information that can be found in the NetWeaver documentation or on SDN in various sources. This was done to make this set of documents more directly applicable without having to jump back and forth in various documentation sources. Nevertheless, you will find a big variety of links that you may follow to widen your view on the topic.

The set of documents will be released step by step on a weekly base starting with the document that you are currently reading. The coming document titles are

**ERP Service Enabling** – Covers the complete mechanism from service definition in XI via proxy generation, implementation and web service enabling in the ERP ABAP stack.

**Service Abstraction** – Covers the design and implementation of the Composition layer.

**UI Modeling** – Covers the design and implementation of the required UIs using Visual Composer

**Process Modeling and Portal Integration** – Covers the Guided Procedure steps and the integration of all visual components into the Enterprise Portal.

## Related Content

Migration of J2EE solutions to SAP NetWeaver:

[SAP Integration Scenario "Deployment on SAP Web Application Server J2EE Engine \(J2EE-DEP\)"](#)

[J2EE Migration and Third-Party Integration Knowledge Center](#)

[J2EE Migration Kit](#)

[SAP NetWeaver 2004s Installation Master Guide](#)

## Copyright

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.