

Get Ready to Build Your Own Composite Application

An Introduction for Developers

Regular Feature

Under Development

Consider what a long-standing HR installation — or any powerful CRM, ERP, or other installation with a large and growing database — means to an organization. This system must be up and running 24x7, and will always require some added custom features. This may involve modifying the backend system; but what if you could create some of these same custom processes without tinkering with the underlying solution?

The push to reduce the costs and effort associated with the modification and maintenance of backend systems has made *services* very appealing. Web services technology and service-oriented architectures allow you to add certain types of innovations — especially for business processes that have a routine workflow or involve forms-based

procedures — without modifying the back end. By using what SAP calls *composite applications* or *xApps*, you can reuse existing services and other assets to create applications that are integrated with, but don't run directly on, these backend applications.

SAP has made more than 200 xApps available to customers, but SAP is not the only source of composite applications. With SAP NetWeaver 2004s, SAP provides the tools and framework to streamline and standardize composite application design. Now, business process experts can play a larger role in designing the flow of these processes.¹ Most

¹ For an introduction to the role of the business process expert, see "Are You a Business Process Expert? A Day in the Life of a BPX" by Helen Sunderland in this issue of *SAP Insider* (www.SAPinsider.com).



Karl Kessler, SAP AG

importantly, SAP customers, partners, and software vendors can develop their own composite applications to run on top of existing SAP solutions.

So what is required to build composite applications? SAP provides the tools, framework, methodology, and runtime

What Distinguishes Composite Applications from Other Solutions?

Composite applications differ from typical application development projects in a number of ways:

- The innovations of composite applications do not require upgrading or modifying the backend system.
- The process of building these applications relies on "composition" from various preexisting sources — that is, the reuse of services, assets, and other existing building blocks to create the new application.
- These solutions are services-based applications. Because services rely on other applications to get the work done, they're perfect for those processes that work off of a heterogeneous set of existing applications.
- Intuitive UI design and a clear workflow for end users are key.
- Composite applications benefit from fast prototyping and leverage this speed for shorter development and deployment cycles.

From a developer's perspective, composite application design follows a specific and consistent methodology — one that differentiates between the various levels of an application: the process level, the user interaction layer, the composite application's business logic level, and the underlying interface to backend systems. This methodology means that each layer could be implemented by different people, using the tools appropriate to their respective skill level.

environment for building composite applications — guiding everything from designing the interfaces, accessing the building blocks during process design, and managing the application at run time. (Remember, a composite application works with — but does not run on — your backend system.)

Two key tools for building composite applications are *Guided Procedures* (GP) and the *SAP Composite Application Framework* (CAF). SAP partners and customers are already using these tools to create innovative composite applications for existing packaged solutions.

The Tools for Creating Composite Applications

SAP NetWeaver 2004s, the basic platform for mySAP ERP 2005, provides a process-level tool for creating workflows and interfaces, as well as a tool for modeling business logic and persistence:

- **Guided Procedures** is the process-level design framework for creating applications that guide the end user step by step through a task. Its powerful workflow features simplify the design process for the business process expert (BPX). GP is a browser-based runtime and design-time tool, accessible directly from SAP NetWeaver’s portal (see **Figure 1**). Here, the process designer and developer can view a gallery of actions, objects, and other assets available to build a procedure — all without drilling down into code or other details.

- The programmatic framework of the CAF is the **CAF core**. Developers use this core to design the user interface, business logic, and persistence of the composite application, including setting access to remote back ends via RFC or Web service technology. The

tooling environment of the CAF core is available as part of SAP NetWeaver Developer Studio (see **Figure 2**). This environment helps the developer make services available for reuse, as well as manage attributes, data sources, and permissions.



Figure 1 The Guided Procedures UI, designed for easy access to predefined content for creating new processes

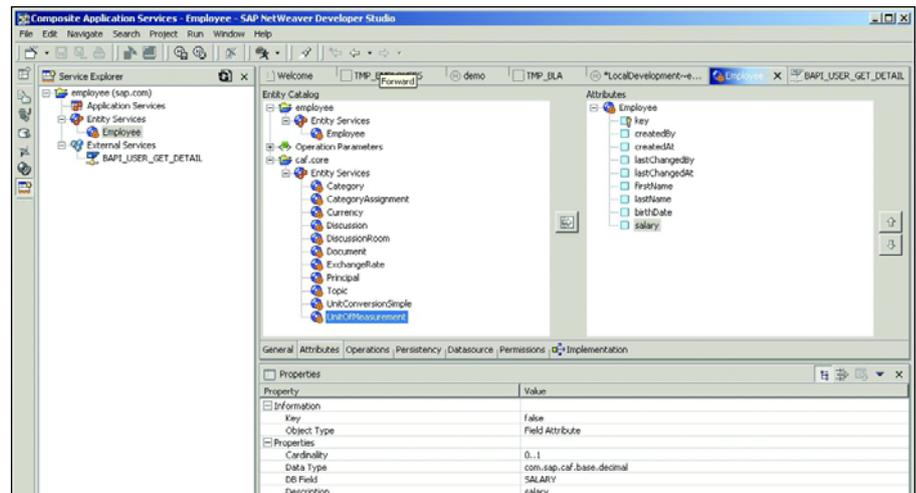


Figure 2 From CAF, developers can create business objects, attach services (such as the employee services shown here), and manage persistence

Before You Start Using Guided Procedures...

Although using Guided Procedures to build composite applications requires no coding, and CAF generates a lot of basic services automatically, both tools rely on some behind-the-scenes development expertise.

There should be flexible building blocks, such as Web Dynpro components, available for developing a composite application. The flexibility of these blocks is key to easily

developing guided procedures on top. While GP also supports low-level components such as business server pages (BSPs) or ITS templates, those approaches are less flexible.

If Web Dynpro components that fulfill the GP specification are already available, you can start right away. If not, you might consider developing these components as an important first step.

In this way, the tooling environment is optimized for both the BPX and the developer, and the designtime environment is fully browser-based and requires no coding skills. The GP framework and the CAF core are fully interoperable.

Figure 3 shows the architecture of a composite application. The end user views the entire business process through a *work center* (more on this later). *Guided Procedures* allows a BPX to create a process flow, define the actions, and assign roles that are visible to any end users involved in the process. The *CAF core* allows developers to create business objects, attach services, and link the composite application's process layer to backend systems. Because it follows this layered approach to design and development, the CAF helps you develop innovative processes that work with loosely coupled backend applications.

Design User-Centric Processes with GP

Guided Procedures works particularly well as the interface for any kind of recurring task; consider any request/approve/update process, such as a time-off request for leave or travel. Approval by an employee's manager triggers an update by a backend system user, such as a member of your HR or accounting team.

✓ Note!

The Leave Request scenario described here is also a sample scenario in the GP designtime environment and includes all required Web Dynpro components.

GP offers a framework for modeling and executing user-centric processes for tasks like these. Its intuitive UI for designing and running processes makes it easy for process experts to adapt an

existing process quickly, and for end users to understand and analyze the current process.

GP's Building Blocks

With GP, the designer or developer has access to application building blocks — from simple parameterized Web pages (think of a Google query), BSP applications, BI reports, Knowledge Management (KM) resources, or Web Dynpro applications.

It takes just a few steps to make a Web Dynpro component ready for GP.² Let's start with a quick example of a guided procedure:

1. From a work center interface, the employee fills in the form and submits it. In **Figure 4** on the next page, the Create Time-off Request form is a straightforward Web Dynpro form.
2. GP automatically notifies the employee's manager. Typically the manager will find this notification in his Universal Worklist inbox, but he can also access it through the GP work center directly.
3. The manager then either approves or rejects the request, and adds comments. Once it's approved, the request goes to HR, where an HR person reviews it and triggers the update of the underlying HCM system (e.g., mySAP ERP 2005). The status field returned by the HCM system is filled accordingly.
4. Finally, the employee sees his original leave request in the portal with the proper status attached.

Once a guided procedure is running, an end user can execute applications under the control of the GP run time, according to the roles set in the GP process definition.

² For more detail on the exact methods to use, see the GP online documentation at www.sdn.sap.com/irj/sdn/developerareas/xapps → *Quick Link* → *CAF* (in the panel at the right).

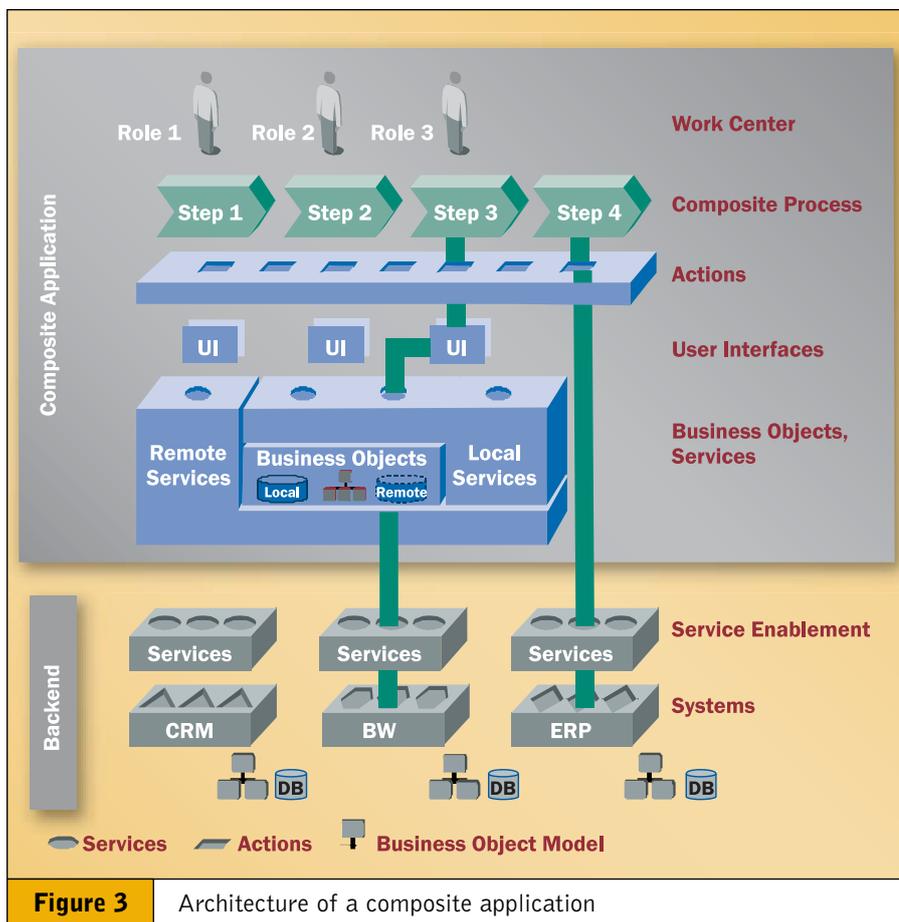


Figure 3 Architecture of a composite application

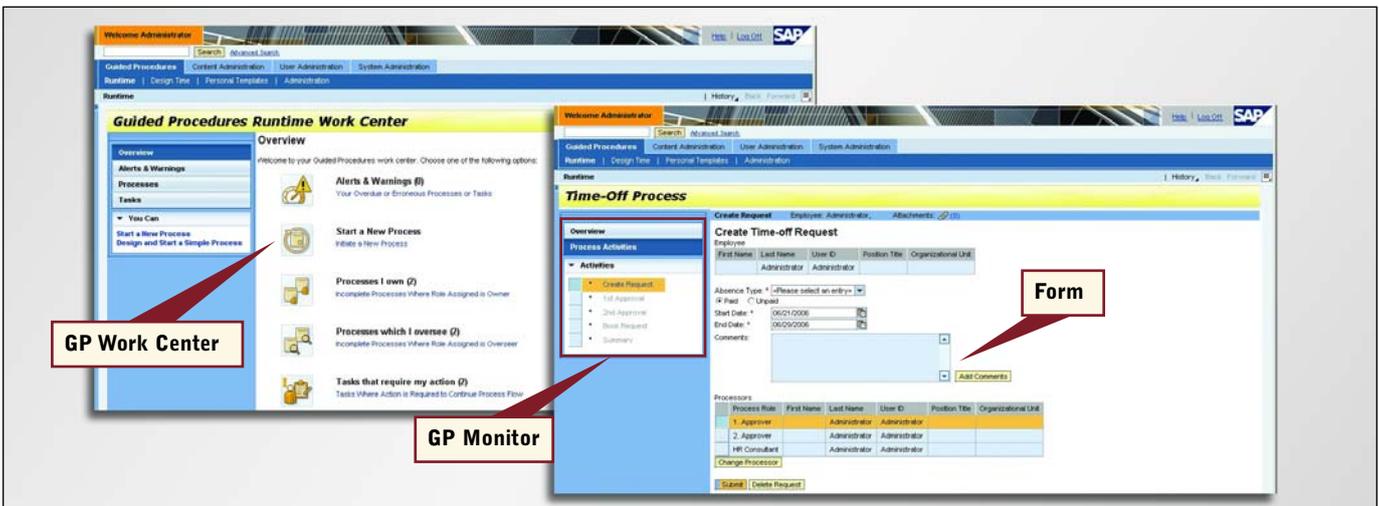


Figure 4 From the portal-based Guided Procedures work center, end users can initiate a process — such as submitting a request for time off — or act on a request

Now that we've reviewed what users would see, let's take a step back and look at how it was designed and built.

Modeling the Process with the GP Design Time

The basic GP modeling entities are *callable objects*, *actions*, and *blocks*. A business process expert or developer can perform top-down modeling, starting with the overall process and breaking it down into manageable pieces. But GP also lends itself nicely to bottom-up design, starting with the existing elements and reusing the building blocks that were previously designed.

In our example, we started with existing Web Dynpro components. In GP's terminology we have *callable objects* that are invoked by the GP run time. These callable objects are the atomic units that cannot be broken down further from a GP designtime perspective. The callable object describes *what* should be executed at a certain stage of the GP process.

Figure 5 shows the creation of the callable object, LeaveCO. GP supports a variety of callable object categories (shown in the Type list). LeaveCO uses Web Dynpro component technology. Other

Why Use Web Dynpro Components in Guided Procedures

Ideally, a guided procedure relies on Web Dynpro components for several reasons. For one, the layered approach of Web Dynpro applications gives you optimal possibilities for reuse and flexibility in your process design. The UI can use Web Dynpro controls and expose a component interface, which allows you to manipulate the component's inner context variables.³

For example, the Web Dynpro component used for approvals is similar to the one used for creation; most of the fields are identical. During GP design time, this similarity makes it easy for designers to reuse and automate the data flow between the different process steps.

³ The state of a Web Dynpro component is set in the component's context variables and can be changed via the Web Dynpro component interface by the user (from the user interface) or from the outside (in this case by the GP runtime environment, which controls the component interface). Other UIs such as ITS and BSP are less flexible and sometimes require rewriting the URL.

component types (BSP applications, ITS transactions, etc.) can also be integrated, but the flexibility of data binding depends on the technology chosen.

In the GP design time, each callable object is paired with an *action* that describes *who* should execute a step in the GP process. For each action, GP assigns a default role to process the action.

One of the important tasks of the process designer or BPX is to assign the roles during process definition. **Figure 6** shows a variation on our Leave Request scenario that requires four callable

objects and associated actions. The GP design time suggests four different roles (Employee, FirstApprover, SecondApprover, and HRConsultant). GP allows you to consolidate roles, for example, if one user is responsible for more than one step.

Modeling Advanced Processes in GP

In a straightforward leave request, all steps occur in a strict sequence. For more sophisticated processes, GP provides advanced modeling and structuring capabilities using *blocks* (shown in Figure 6).

In the simplest case, a block contains a sequence of actions. The block, in turn, is the only constituent of the process. The

block allows you to define actions that are executed in parallel. Control flow constructs, such as logical decisions and

loops, can be defined at the block level, as well. Blocks can also be nested, allowing a large process to be decomposed.

Keep in mind, however, that GP's focus is user-centric processes. The charm of GP lies in the user's ability to oversee and quickly inspect the GP process's status. The GP monitor (shown back in Figure 4) shows the end user which actions are currently under execution.

Our simple GP example is a straightforward procedure. But what if, for example, the manager wants to see the history of an employee's time-off requests from the previous quarter? This would require access to the HCM backend systems, as well as to technology to model the entities and services required for the composite application.

In other words, you need a foundation for GP that manages both the persistence and process data. The CAF was designed to relieve the underlying applications from managing persistence and to avoid any modifications to the underlying applications.

The Role of the CAF Core

The basic idea behind CAF is to manage only those entities and services that a particular composite application requires — nothing else. Whether the underlying business objects are stored in a backend system or are persisted on behalf of the composite application, CAF core services help make the composite application independent of the backend implementation.

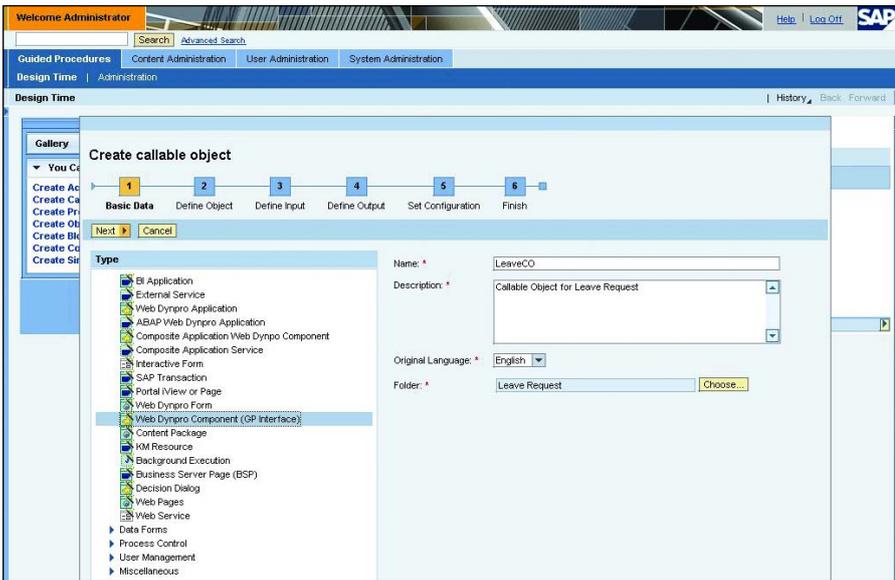


Figure 5 Create a callable object using Guided Procedures

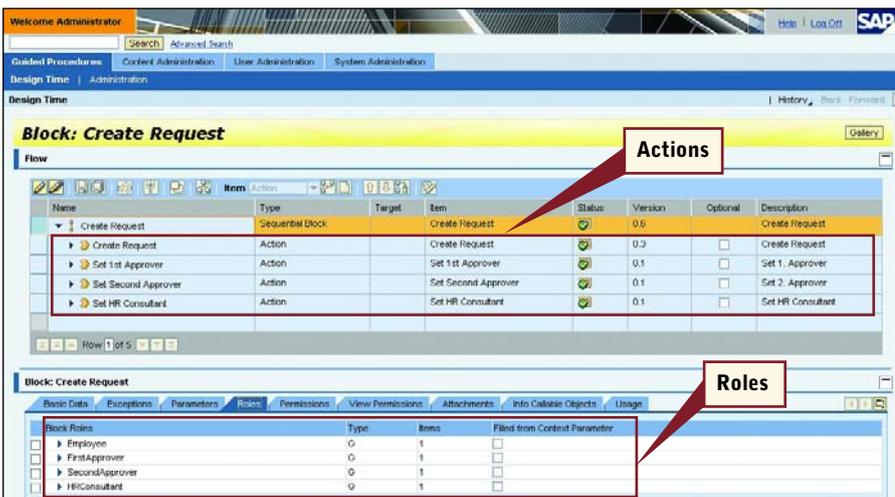


Figure 6 Assign roles for each step (action) using GP design time

How Does Guided Procedures Differ from SAP Business Workflow?

Although Guided Procedures shares some conceptual similarities with SAP Business Workflow, it offers additional innovations for creating composite applications: browser-based design time tools that are easy to learn and use, and the integration of innovative SAP NetWeaver technologies such as Web Dynpro, Adobe Interactive Forms, and KM capabilities.

They also cater to different skill sets. Using SAP Business Workflow requires a deep understanding of

the underlying application, including programmatic enhancements at the ABAP level, for some complex workflow capabilities.

In contrast, Guided Procedures assumes that the underlying applications support some simple rules needed for proper data flow between the different process steps. For creating relatively simple forms or workflows, Guided Procedures is an excellent choice, well suited for a fast learning curve.

How does it work? The CAF core builds on the foundation of J2EE technology, but provides the means to introduce a higher-order abstraction compared to pure entity and session beans suggested by the J2EE standard. The CAF core is a metadata-driven approach that generates a lot of the underlying Enterprise JavaBean code. It also offers smart interoperability with Web services and RFC technology to ease backend access.

CAF core development takes place in the CAF perspective of SAP NetWeaver Developer Studio.⁴ Once you open SAP NetWeaver Developer Studio's CAF perspective, shown back in Figure 2, you can create new entity and application services. External services based on RFC or Web services are also supported:

- An *entity service* starts with the definition of an entity type and its attributes. Entity types might refer to other entity types going beyond a simple relational persistence approach. From a development perspective, you always should start with the entities to define the persistence layer.
- Once you have modeled an entity type, *basic services* are automatically generated for you — services such as create, read, update, and delete methods (CRUD). The CAF core design time generates the underlying session beans following the J2EE standard, including the required deployment descriptors.
- You may also define additional *application services*, which can be used later as an entry point for the callable object interface used by GP.
- *External services* allow you to wrap existing RFCs or Web services to make them seamlessly available within the CAF context.

The CAF's design also means that if you replace a legacy HR system with

mySAP ERP, the existing composite application and processes on top can remain almost unchanged — provided the core entity and application services are properly reflected in the new system context.

Outlook

SAP partners and customers are already using SAP NetWeaver 2004s tools to create innovative composite applications on top of existing packaged applications such as mySAP ERP 2005.

Future releases of SAP NetWeaver will extend this offering in a variety of ways. For example, applications that have been modeled using SAP NetWeaver Visual Composer can easily be made GP-ready. This will allow business process experts to take a fully code-free approach to developing composite applications — from both a process and business logic level.

Additional runtime options will also be available. SAP NetWeaver will have greater prominence as an environment to actually run composite applications. A preview of the underlying JEE 5 engine that would run these applications has already been presented at the

2006 JavaOne conference. Composite applications that provide strong backend access, but do not require a full SAP NetWeaver stack (e.g., ABAP) for execution, can then be executed directly on SAP NetWeaver.

For more information about GP and CAF, see <http://help.sap.com> → *SAP NetWeaver 2004s* and <https://www.sdn.sap.com/irj/sdn/developerareas/xapps> → *SAP Composite Application Framework*.

To learn more about xApps and composite applications, including offerings from SAP and partners, see www.sap.com/solutions/xapps/index.epx and www.sdn.sap.com/irj/sdn/developerareas/partners-and-isvs. ■

Karl Kessler joined SAP in 1992. He is the Product Manager of the SAP NetWeaver foundation, which includes SAP NetWeaver Application Server, Web Dynpro, ABAP Workbench, and SAP NetWeaver Developer Studio, and is responsible for all rollout activities. You can reach him at karl.kessler@sap.com.

⁴ CAF also interoperates with the SAP NetWeaver Development Infrastructure (NWDI) if you decide to use NWDI capabilities such as source code control and transport.