

# User Interfaces Made Easy: How to Design Forms-Based Processes in Manager Self-Services (MSS)

Simplification is the key to many a successful strategy, whether you're turning a loss-making production unit into a profitable one or preparing a sales force for a new range of products. By streamlining its business processes, a company can eliminate unnecessary work, make employees more effective, and elevate the company's business performance. But *over-simplification* can lead to sloppy processes that end in ruin for your business — when control mechanisms are circumnavigated or exceptions and downward trends go unnoticed. How do you transpose these requirements into an automated process so that users are shielded from its sophistication and the cost of software development is kept to a minimum? This article has two parallel threads: one showing how the user interaction is simplified, the other showing how the development is simplified.

In its own way, SAP Manager Self-Services (MSS), SAP's role-based desktop for managers, helps achieve this ease-of-use and process agility. In MSS, managers are always in control — they can see at a glance where they are in a process and how to trigger new processes. But while MSS keeps the user interfaces as simple as possible and managers interact with workflows in an intuitive way, underlying processes can still be very sophisticated and dynamic. New processes can be developed and old processes improved with minimal development effort and no disruption.

## Interactive Forms for Interactive Business Processes

So *how* does Manager Self-Services maintain both frontend simplicity and backend sophistication? One way is through electronic forms. In a previous column, we looked at using simple electronic forms to bring users into any workflow.<sup>1</sup> Now, release 60.1 of the business package for MSS (delivered with mySAP ERP 2004) has taken forms a step further by blending electronic forms with paper-like qualities — Interactive Forms based on Adobe software — into its manager-specific workflows, and combined this with the Internet Service Request (ISR) methodology (more on this later) to create an even simpler and more familiar mode of interaction for users (see **Figure 1** on the next page).

Interactive Forms are the result of the partnership between SAP and Adobe to provide a state-of-the-art solution designed for interactive business processes using electronic forms. But what's so special about Interactive Forms? On the one hand, they are the software equivalent of paper forms, in that they:

- Are easy to navigate and provide a familiar paper-like user interface
- Offer simple guidance to users on next steps, together with descriptions of the effects

<sup>1</sup> See my article, "Bring Even Casual Users into Your Business Processes with Easy-to-Generate Electronic Forms" in the July-September 2003 issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).

## Regular Feature

# Process Technology Tattler



Alan Rickayzen, SAP AG

- Include simple graphics, such as frames, to separate the different contexts within the form from each other or highlight areas of special attention
- Can easily be printed (WYSIWYG) to replicate paper forms as needed

On the other hand, the electronic nature of these forms allows for automated processes, such as:

- Automatically prefilling forms with context-sensitive data
- Allowing information to be picked from lists, avoiding typographical errors and incorrect data
- Immediate collection and verification of data entered into a form

What's more, forms-based processes are automated by a workflow engine, which distributes the forms to the right users at the right time. And to customize

your own processes, you'll find there is very little needed to add these forms to a current business process.

## The Internet Service Request (ISR) Framework

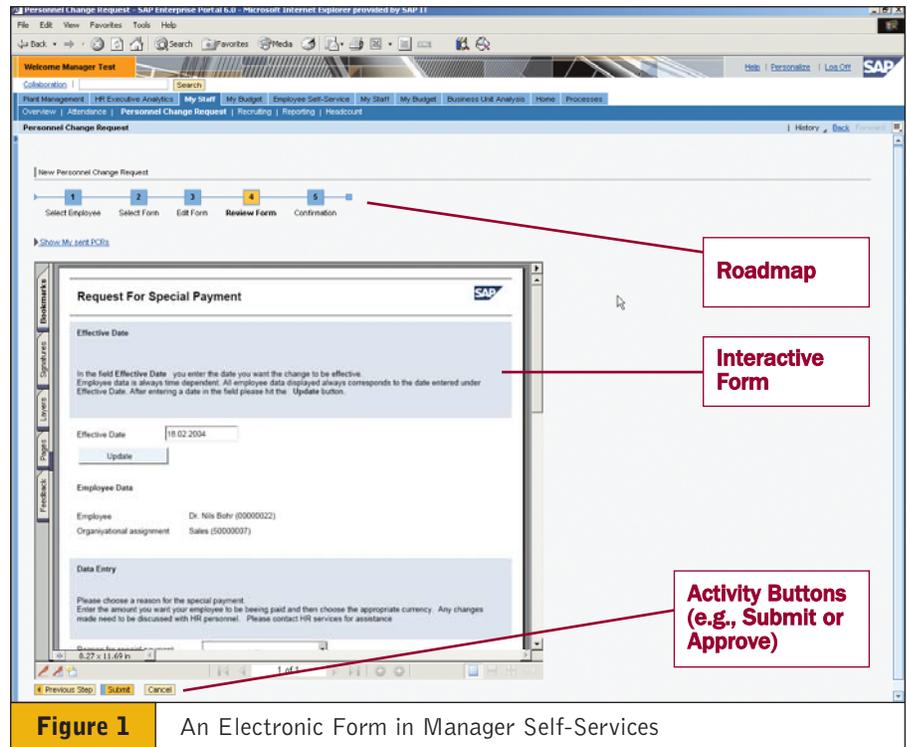
In the past, customers have been left to achieve the blend of forms management, business logic, and workflow management as best they can. However, this leads to a myriad of one-off implementations with no underlying common characteristics in the handling of the user interface or the development architecture. This means that process participants are confused by the different navigation/handling algorithms, and the maintenance (adaptability) of the processes is difficult because different development algorithms are used for different processes. For example, in any process, the business logic may be found in the form, in the workflow, or in the application.

Enter the Internet Service Request (ISR).<sup>2</sup> This is the framework used to achieve this blend consistently. ISR helps non-specialist users design Interactive Forms for Manager Self-Services (and Employee Self-Services, or ESS) in a consistent way, and it gives participants a familiar structure to how they interact with the processes.

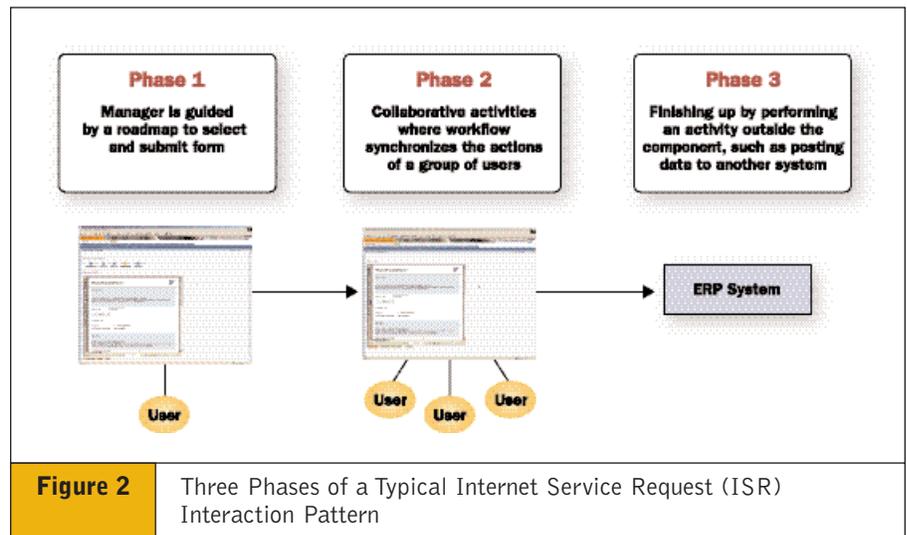
In a nutshell, the interaction pattern consists of three phases, the last two being optional (see **Figure 2**):

- *Roadmap* to guide the manager through the process of selecting and submitting the form
- *Workflow collaboration* with different participants to clean and verify the data
- *Final processing* of the form

<sup>2</sup> You may have met the ISR in previous releases when it was known as the Internal Service Request. The underlying principles have remained the same (separating the business logic from the display logic), but the user interface technologies supported have evolved significantly, with support for BSPs, iViews, and, most recently, Interactive Forms using Adobe software.



**Figure 1** An Electronic Form in Manager Self-Services



**Figure 2** Three Phases of a Typical Internet Service Request (ISR) Interaction Pattern

Similarly the development of these forms-based workflows, as we will see later in detail, consists of three separate phases:

- Form design
- Adding the business logic to BAdIs to be deployed in the scenario
- Workflow process design

With ISR, generic forms/workflow integration — what you'd find in a typical forms process in mySAP ERP —

has been implemented in a single software layer built on SAP NetWeaver. This is designed specifically to streamline forms-based development, making it very easy for customers to enhance or create their own forms-based scenario. ISR is responsible for the roadmaps that guide users through the form selection and submission, the workflow integration, and various follow-up actions such as transferring the results to another software component. Once a new scenario has been developed, it is

simply checked into the ISR as needed, without the need to modify the SAP component (MSS or ESS).

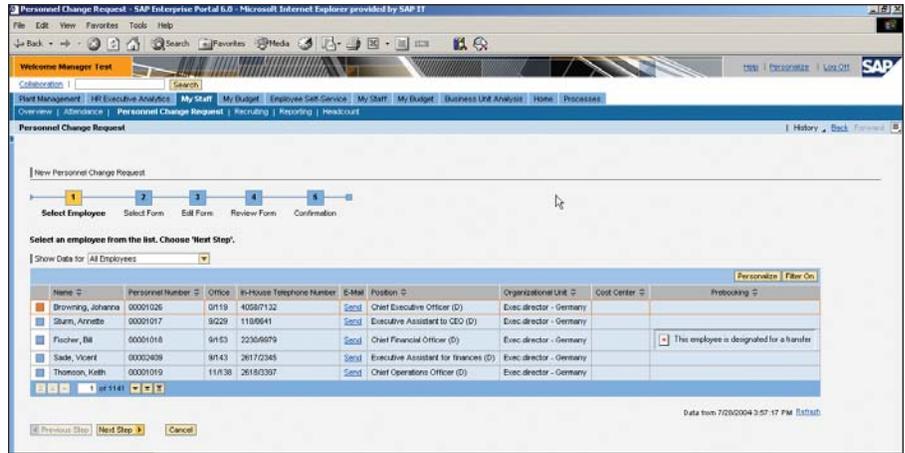
This means that business process owners — anyone with a penchant for graphic design — can paint a form with Adobe Designer (which comes with SAP NetWeaver), laying the foundation for the complete process definition. This removes ambiguity and helps ensure that the detailed development that takes place downstream will be in line with your business requirements.

### How Will Managers Use Interactive Forms with Self-Services?

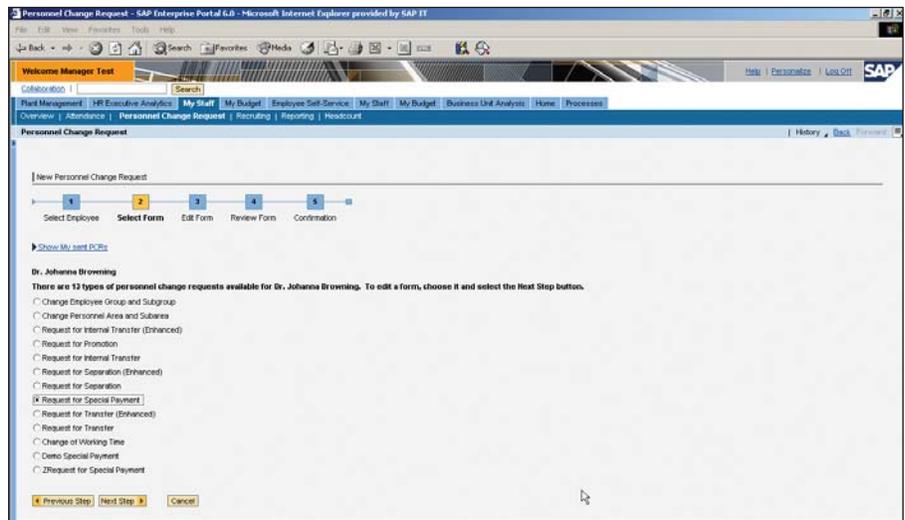
Before describing how to create a custom forms-based process, let's take a closer look at what managers will see when they use Interactive Forms. Consider a *Personnel Change Request* process, which is delivered with SAP MSS and included with mySAP ERP. This process is triggered when a manager decides to change an employee's data (because of a relocation or a promotion, for example, or when instigating a one-off bonus payment to a team member).

Let's say an employee recently changed job functions and is moving to a different group within the office. The manager will need to modify the employee's status, so she selects the MSS area of the portal, "My Staff," and follows the roadmap as shown in **Figure 3**.

After selecting the employee, the manager goes to the next step of the roadmap to select the form (**Figure 4**), which automatically triggers the exact process that she needs. Here, the manager may not even be aware that she is selecting a process — she is more likely just thinking in terms of choosing the right form. So you can see that implementing a forms-based process is not just a matter of simplifying user interfaces (UIs) — it also simplifies the



**Figure 3** Selecting the Employee for a Personnel Change Request in MSS



**Figure 4** Choosing the Right Form to Fill In

user experience and relies on a level of abstraction to shield the user from the complexities of the underlying process.

After the manager selects the form, she fills it out, reviews it, and submits it (the roadmap phase). At this point, workflow takes over and the form is routed to all users involved in the process, with the BAdIs (where the business logic is embedded) validating and retrieving data along the way. A form might finish off by going to accounting for payment, to human resources to update personnel records, or to facilities management to make sure the new employee has a telephone when he arrives at his new post.

The appropriate users during the workflow phase will receive the form either as a work item in the Universal Worklist or as an email notification. Here again, simplicity is key: users will simply regard this as receiving a form that they need to approve, check, or perhaps supplement with additional data (see **Figure 5** on the next page). No matter how complex the underlying process, on the surface the form is simply whisked away to the next user in the chain.

So now that we know how Interactive Forms simplify UIs for managers and users, how does ISR simplify development?

## How to Develop a Forms-Enabled MSS Process

There are two steps involved in the integration of forms into MSS: first, designing the form and the dataflow using ISR, which is easy and can be done by even the business process owners, and then designing the workflow that uses the form, which is performed by workflow designers and MSS developers.

### Designing the Form

The form can be designed from the ISR framework (transaction QISRSCENARIO in the SAP Web AS). The ISR framework (see **Figure 6**) accesses Interactive Forms for development using the ABAP Workbench integration.<sup>3</sup> To make the complexity of backend connections and workflow design transparent to the forms designer, ISR has whittled this down to just a few basic steps. The first two can be done in any order:

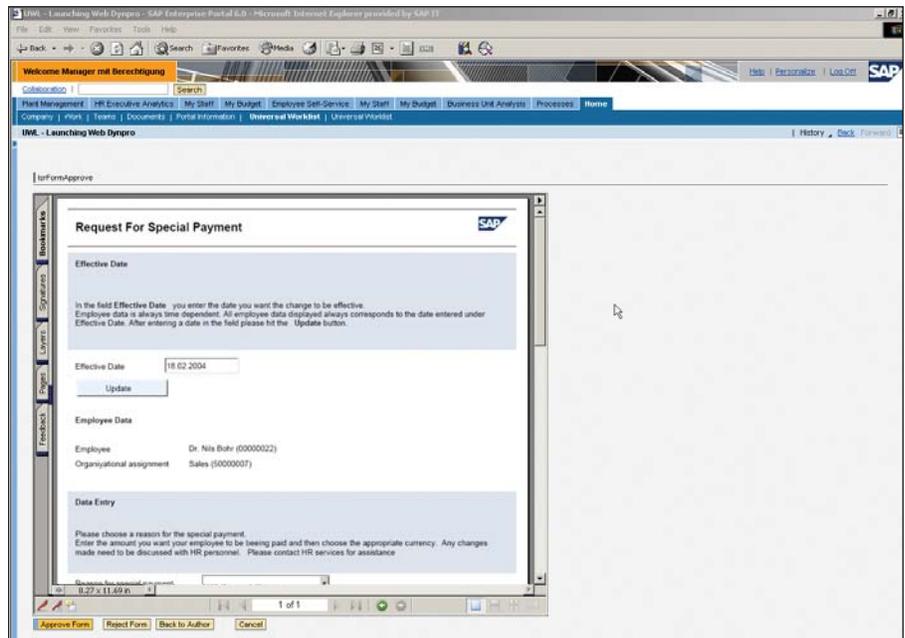
- Specify that the user interface is based on Interactive Forms (as opposed to Java Server Pages, for example) and paint the form.
- Declare the attributes of the form, or predefined *scenario characteristics*, which are listed in the ISR for them.

Finally, the characteristics are mapped to the display elements used in the form, such as drop-down lists, list boxes, or text fields.

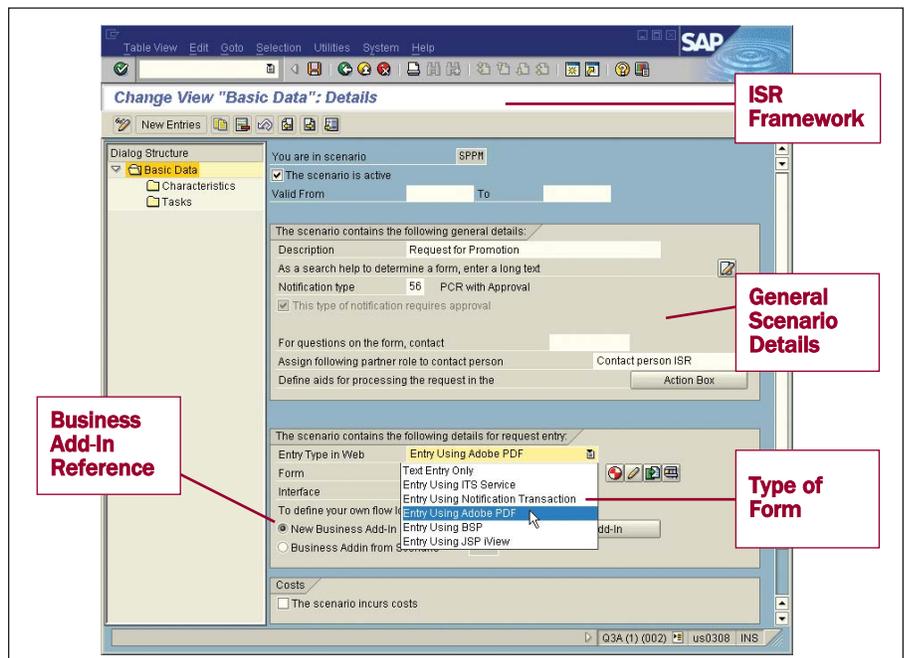
When the form is displayed to users, these characteristics are populated by the business component and displayed within the form. So during design, the forms designer must determine exactly where in the form each characteristic is displayed. In addition, the appearance of the field — including its font, whether a drop-down list or text display is used, any frames or graphic enhancements, etc. — are also specified.

Whether or not you specify the attributes before designing the form,

<sup>3</sup> You'll see it referred to as Form Builder in the ISR screen, which is now integrated with Adobe Designer.



**Figure 5** Executing the Form in the Universal Worklist via the Portal



**Figure 6** The ISR Framework

when you use Adobe Designer (see **Figure 7** on the next page) you will see that it really is a piece of cake — you can paint a form without any programming or development background. Designing a form with Adobe Designer is just as easy as doing so with the desktop software Adobe Acrobat

Professional.<sup>4</sup> In fact, the user interface is virtually the same; the only significant difference is the list of characteristics that you chose earlier, which allow the

<sup>4</sup> Adobe Acrobat Professional is a software solution that enables users who work with graphically complex documents to improve the reliability and efficiency of business-critical document exchange.

designer to map the scenario characteristics to the graphical form elements that you'll create in Adobe Designer. This "binding" is a simple drag and drop affair — designers can drop the characteristics one at a time onto the corresponding graphic elements.

Binding also needs to be performed if you are replacing an SAP-delivered form with the customer's own form created from scratch. But the good thing is that you can do this without affecting the business or application logic that the developer encapsulated in the BAdIs.

In the background, an XML-based file (form template) is generated, which is reused at runtime when the forms are rendered.

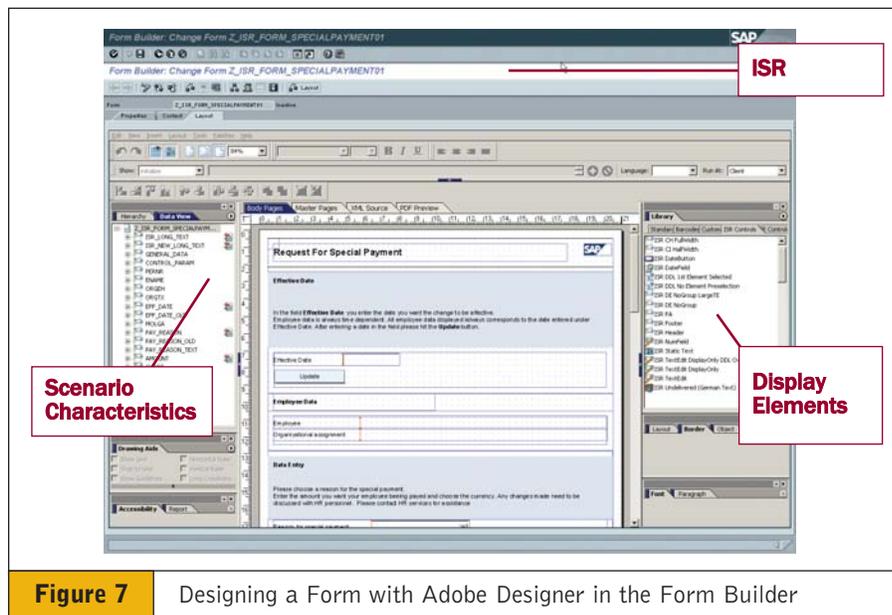
### ✓ Building the Workflow

Building the workflow for an Interactive Form is just as straightforward as

building any other workflow. From the workflow builder, define the trigger — activating the event linkage to the business object "Notification" (*BUS7051 CREATED*), which is the generic event

used whenever a form is submitted via ISR.

At runtime, the ISR controls the form rendering and data validation



**Figure 7** Designing a Form with Adobe Designer in the Form Builder

## Benefits of the New Interactive Forms Design for MSS

MSS was developed with both reusability and business users in mind. The new forms and visual elements for the MSS user interface support both — they offer new interfaces and a new look and feel for business processes, but without having to redo any customizing in your HR master data.

A big advantage of MSS is that its development was business-oriented from the beginning. The forms — including the design and processes — have been designed by business experts from start to finish. But the forms are also *customizable* — there's nothing to stop developers from making minor changes (by adding the company's logo, for example) or even replacing the form with one designed from scratch. Since the forms and scenarios delivered by SAP are templates for processes, customers and partners will probably want to adapt them to the local environment.

The MSS user interface is state-of-the-art, and thanks to Web Dynpro logic on the client side, there are no compromises when it comes to complex, cross-application processes, either.

The business logic is encapsulated in the BAdI and kept in the backend business component, whether it is mySAP ERP or even SAP R/3 4.6C, provided that there is an SAP Web Application Server in the software landscape

available to handle the Interactive Forms. Your backend system contains the data for the form (HR customizing and master data) together with the business logic within the application and the process logic for developed workflows. When you design or interact with a form, the SAP R/3 system is queried via a remote function call (RFC). Both runtime and design-time are taken care of by an SAP Web AS, which is linked to Adobe's supporting technology — Adobe document services — delivered as part of SAP NetWeaver.

To work in MSS, the underlying mechanism behind a form is a Java Web Dynpro application. The business logic that accompanies the form is embedded in BAdIs, which are used whenever the form data is validated or prefilled, such as when a drop-down list needs to be generated in the form. By sticking to this principle you will avoid embedding business logic in the form itself (through messy JavaScript routines, for example), which is difficult to maintain and keep in sync with the backend component. The ISR provides a powerful mechanism to determine which fields should be displayed when (different roles and actions require different information to be displayed), so it is a good idea to use it wherever possible to avoid dispersing the business logic in an inconsistent way.

✓ **Note!**

When you're using MSS, you'll see that items such as approval buttons are not rendered within the form; instead they are part of the outer frame surrounding the form where the workflow steps are presented. This helps the user distinguish, at a glance, between data in the form and actions they have to take to move the document to the next step in the workflow. The same pattern is used during the first phase of the process, where the roadmap guides the user through the submission procedure.

within the workflow step while the workflow management system routes the forms to the different users. So once again the technical responsibilities of the different components are clearly separated, and this development model will yield a consistent process definition every time. The steps and control elements in the process, together with the rules determining which agents perform the different form steps, are modeled in the normal way within the workflow builder.

The only specific relating to the Interactive Form is the task used to call it. This task is based on the generic WebFlow service method EXTSRV PROCESS,<sup>5</sup> and the launch mechanism for the Universal Worklist (the inbox) is configured with transaction SWFVISU to call the Web Dynpro transaction. The parameters here are *application* and *package*, which specify the instance ID of the form.

So after the manager fills in the initial form, reviews it, and submits it, the workflow takes over, eventually generating work items in the Universal Worklist so that the other participants in

<sup>5</sup> For information on the EXTSRV PROCESS, see my article "Accessing External Web Services from SAP WebFlow" in the January-March 2002 issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).

the workflow process can perform their duties (such as approving the form or adding or changing form data).<sup>6</sup>

One difference between Interactive Forms and other types of electronic forms is that the Interactive Form data is not stored in the workflow container. (With simple forms, data is transferred to and from the workflow container directly.) Instead, here it is part of the business object, stored in the MSS databases, and accessed via the characteristics associated with the form. Here, the data is used by other parts of the MSS application, and there is program logic that evaluates and verifies the form data. Clearly the data belongs to the application.

✓ **Note!**

There are some exceptions to this, where data is only needed by the process. For example, if just before triggering the process, the initiating manager specifies the particular users who will need to perform subsequent steps in the process, the data is not needed by the application. This would most likely be part of the workflow container, since the data is only used for the process control (but it can be tracked in the workflow log).

## Conclusion

In your efforts to make life easier for your managers and business users, you will appreciate how effective Interactive Forms based on Adobe software is in simplifying the user interface for non-specialist activities such as MSS or ESS. But more importantly, by implementing forms-based processes, you won't have to sacrifice the integrity of your underlying business processes, which by nature can be pretty sophisticated.

<sup>6</sup> To learn more about the full advantages of using the Universal Worklist, see my article "Implement a Central Inbox for One-Stop Access to Work Items from Any Business Process" in the October-December 2003 issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)). The email notifications described in this article are also completely supported without any additional development.

You may also have noticed how the separation of the process logic from the data makes it easier for the business process owners (those responsible, but not necessarily participating in the processes) to keep control of the process. The hard-coded business logic has been isolated in the BAdIs so that it can be reused and adjusted independently of the process design. Similarly, the user interface is in the control of the graphic designers and process owners. This is not coincidence — moving as much ownership and control of processes as possible to process owners rather than the IT department is one of the main objectives behind Business Process Management (BPM) in SAP NetWeaver.<sup>7</sup>

For more information on the topics covered in this article, see the following sections in the the SAP Service Marketplace: **mss, isr, workflow** (e.g., <http://service.sap.com/mss>). To learn more about Interactive Forms, visit [www.sdn.sap.com](http://www.sdn.sap.com), navigate to the Web Application Server developer area, and click on the Interactive Forms "quick link." And to find out more about the Adobe partnership with SAP, view the white paper entitled, "SAP AG and Adobe Forms: Enabling Collaboration with Interactive Forms" at <http://service.sap.com/adobe>. ■

*Author's note: I'd like to extend my special thanks to Dirk Becker and Jeremiah Stone for their support in this article.*

<sup>7</sup> For more information on Business Process Management, see my article, "A Primer on Business Process Management (BPM) in SAP NetWeaver" in the July-September 2004 issue of *SAP Insider* ([www.SAPinsider.com](http://www.SAPinsider.com)).

Alan Rickayzen has been with SAP since 1992 and in data processing since 1988. Since 1995, he has been performing development work as well as process technology consulting for various major US customers and, as a result, has amassed a good deal of technical knowledge in collaborative process technology. Alan Rickayzen is co-author of the book *Practical Workflow for SAP*, available at [www.sap-press.com](http://www.sap-press.com), and may be contacted at [alan.rickayzen@sap.com](mailto:alan.rickayzen@sap.com).