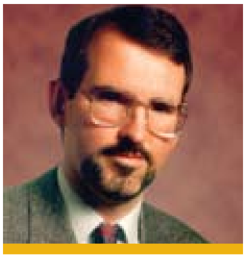
SAPinsider This article appeared in the Oct ▪ Nov ▪ Dec 2007 issue of *SAP Insider* and appears here with permission from the publisher, Wellesley Information Services (WIS), **www.WISpubs.com.** WIS Wellesley Information Services

What Should – and Should Not – Be Running on the Same System in Your SAP Environment?

Guiding Principles for Reshaping Your SAP ERP Landscape



Dr. Franz-Josef Fritz (franz-josef.fritz@sap.com) has a Ph.D. in mathematics and 30 years of experience in all areas of IT. Workflow and business process management have been particular areas of interest for much of his life. He has worked at SAP since 1993 as Program Director and Vice President with responsibility for the Business Process Technology and Internet Business Framework departments. Since 2003, he has been responsible for several areas within SAP NetWeaver Product Management.

✔ **NOTE!**
While this article does highlight some key issues for SAP ERP landscapes, it is not within its scope to look at the full breadth of planning and decision-making factors for setting up and evolving an entire system landscape.

While innovation is a driving force in any enterprise, for some parts of the business – and their underlying software – there *is* such a thing as too much change. SAP customers tell us that for some essential business processes, they want as much stability as possible. For this reason, SAP is keeping the core capabilities and processes of SAP ERP 6.0 stable for an extended period of time, and is extending its maintenance period.[1]

This does *not* mean the end of innovation for your SAP ERP system, however. ERP *enhancement packages* – periodic deliveries of additional features and functions that run on top of SAP ERP – will enable customers to selectively use new services and processes, while allowing other customers to ignore these pieces or select more relevant parts of the enhancement packages for their business. Ensuring stability – while at the same time retaining the flexibility to nimbly respond to technology developments – is forcing customers to rethink the landscape around their SAP ERP system and consider:

- What are the key building blocks of a system landscape around SAP ERP and SAP NetWeaver?
- What are the benefits and drawbacks of running ABAP-based and Java-based applications in the same system (the so-called "double stack")?
- What applications should be running separately in my system landscape to maximize flexibility and innovation – but minimize unnecessary change?
- Where will these separations help ease maintenance efforts?

This article covers two larger developments – the new SAP NetWeaver capabilities and the evolution of ABAP and Java application servers – and traces their potential impact on what runs where, and why.

Platform Stability and Flexibility: How New Capabilities of SAP NetWeaver Affect Your System Landscape

All the focus on stability for SAP ERP requires that its underlying platform – SAP NetWeaver 7.0 (formerly SAP NetWeaver 2004s; see sidebar) – also remains relatively stable. Like SAP ERP 6.0, SAP NetWeaver 7.0 will be on long-term maintenance, including support for new operating systems, database management systems, and browser releases.

With any maintenance or support offering, the goal is to sustain a very high level of compatibility, and changes during maintenance cycles are designed to avoid disrupting any SAP applications or custom-built applications.[2] But for an extensive product like SAP NetWeaver that is also constantly embracing new technology advances – including further big steps in enterprise SOA with all of the new interoperability standards, new approaches for model-based development and composition, and new Java versions – there are limitations on what can be accomplished in a maintenance release alone.

For this reason, SAP is delivering selected parts of a new SAP NetWeaver release (SAP NetWeaver 7.1) in 2007. The new developments in SAP NetWeaver – along with changes in the use of ABAP and Java environments – also mean that there are some new factors to consider when evaluating the structure and maintenance of your landscape: Where does it make sense, from a functional or technical perspective, to separate systems? Where will you gain the

[1] For more details on the SAP ERP strategy and maintenance periods, please visit http://service.sap.com/upgrade-ERP.

[2] This level of compatibility applies as long as the custom applications are built according to the compatibility rules and fully utilize the abstraction layers provided by the SAP application server infrastructure.

most flexibility to adapt to technology changes with minimal disruption?[3]

In all cases, the way you structure your landscape – especially when you can separate the elements of your landscape that change most quickly – can simplify maintenance and enhance the stability of your core processes while maximizing innovation and new opportunities. Let's begin this discussion by examining the changes coming with SAP NetWeaver.

What Is New in SAP NetWeaver for 2007?

The key driver for delivering innovation in 2007 is the promise of enterprise SOA to enable business process flexibility on top of a stable core, to utilize existing investments to the largest extent possible, and to leverage new business opportunities by building and adapting composite applications.

To achieve this, the new pieces delivered in 2007 as part of SAP NetWeaver 7.1 are:

- SAP NetWeaver Composition Environment 7.1
- SAP NetWeaver Process Integration 7.1
- The Enterprise Services Repository
- SAP NetWeaver Mobile 7.1

In addition, there will be two add-on components for the current SAP NetWeaver 7.0 release that do not require the new 7.1 release:

- SAP Enterprise Search 7.0
- Collaboration features for SAP NetWeaver Portal 7.0

All of these pieces are designed to interoperate with existing releases of SAP ERP and other parts of the SAP Business Suite (see sidebar on the next page). However, those pieces that rely on a new release (7.1) of SAP NetWeaver will need to run on a separate system with a separate maintenance cycle. (This does not exclude the possibility of sharing the same physical host systems – with or without virtualization layers as an additional means of separation.)

Which Pieces of SAP NetWeaver Should Be Combined into One System?

While there is no single answer to fit all customer situations, I can offer some advice and guidelines.

1. Look at Speed of Evolution

Naturally, some systems change more quickly than others. SAP ERP is the stable core for running critical business processes and will not undergo major release changes in the foreseeable future (once you have arrived on SAP ERP 6.0 as your core platform).

On the other hand, updates and innovation in SAP NetWeaver Composition Environment and SAP NetWeaver Process Integration are most important for an end-to-end enterprise SOA approach. Likewise, you can expect that SAP NetWeaver Portal and SAP NetWeaver Business Intelligence will move at a faster pace than SAP Business Suite components – and this is true not only in SAP's release cycle, but also in customer landscapes. For this reason, SAP is delivering selected parts of SAP NetWeaver release 7.1 in 2007.

The new offerings for SAP NetWeaver are designed to interoperate with existing releases of SAP ERP and other parts of the SAP Business Suite. Note that those pieces that rely on SAP NetWeaver 7.1 will need to run on a separate system — with a separate maintenance cycle.

Why Did SAP Change the Release Names of SAP ERP and SAP NetWeaver?

You may have noticed that SAP is introducing a new way of referring to its latest releases – no longer in years (SAP ERP 2005), but in release numbers (SAP ERP 6.0).

There are pros and cons for every release-numbering scheme, and no single scheme satisfies all possible requirements. It became clear, though, that using the release year – as we did until recently with some SAP offerings – is only helpful in the early phase of a release shipment. Over time, the year loses its meaning; after all, does anybody really care about the release year of SAP R/3 4.6C anymore?

From an end-user perspective, the release-numbering scheme should not matter at all – ideally a new release should behave just the same as the previous one as long as users deal with the same scenarios and capabilities.

From a technical perspective (installation, administration, and change management), release nomenclature *does* matter, of course. But it is also necessary to look at support package levels and other details to successfully manage your system life cycle; this is true not only for SAP, but also for each operating system, database management system, or any other major piece of software.

From that perspective, release numbers should be closely related to what you find and see in technical tools – not just in which year they were released.

Therefore – if you care about release numbers at all – referring to these releases as SAP NetWeaver 7.0 (also known as SAP NetWeaver 2004s) and SAP ERP 6.0 (also known as SAP ERP 2005) makes the most sense.

In an "ideal" world, the release numbers of all corresponding and interacting pieces of software would be on the same level. This is an oversimplification, of course; it would imply that all pieces move and change at the same speed – which is rare in the "real world" and not even desirable from a business process perspective.

[3] This article focuses on new developments in SAP NetWeaver for SAP ERP-centric system landscapes. Completely new applications for customers requiring on-demand delivery models are out of this article's scope.

2. Look at Responsibility and Governance

Different customers have different organizational structures, building on regions, product lines, and customer segments. These structures drive and are often directly reflected in many decisions about setting up and connecting systems to run the business – and they still remain key considerations in how any IT landscape is structured.

3. Look to Balance the Simplicity of Centralization with the Flexibility of Distributed Systems

Clearly, the simplest approach is to install and run as much functionality as possible on a single system – a consideration that is especially important for smaller businesses. In fact, SAP has such offerings (such as the SAP Business All-in-One solution[4]) for companies where this is the dominant need. On the other hand, flexibility is better served by distributing units that move with different speeds and where decoupled maintenance cycles can limit the impact of maintenance on the rest of the landscape.

These guidelines are extremely pertinent to the changes going on in the SAP ERP and SAP NetWeaver offerings.

Of course, factors such as security and availability – and many others – remain crucial to any landscape, though they are beyond the scope of this article. However, in the "distribution vs. centralization" debate, these three guidelines are pertinent to the changes going on in SAP ERP and SAP NetWeaver.

As you look at these issues, here's another consideration: the development language of your applications. Take another look at where and how your enterprise is using ABAP and Java; there are new opportunities for restructuring here as well.

ABAP and Java: The Future Role of the "Double Stack"

SAP has built several generations of business software on its own business application language, ABAP. With the emergence of Java, SAP embraced the language for its openness and interoperability. Together, ABAP and Java are referred to as the "double stack."

The evolution of these two languages calls for another examination of your systems: Where and when do you optimally use these languages? And where is it beneficial to run ABAP and Java separately?

The Current State of ABAP, Java, and SAP Solutions: A Quick Summary

Across the SAP ecosystem, it took some time to fully understand the advantages and disadvantages of

[4] To learn more about SAP Business All-in-One, see Sheila Zelinger's article in the "Small and Midsize Business Solutions" special feature in this issue of *SAP Insider* (www.SAPinsideronline.com).

An Overview of New SAP NetWeaver Features

What follows are brief descriptions of new features being delivered in the components of SAP NetWeaver releases 7.0 and 7.1.

SAP NetWeaver Composition Environment 7.1

SAP NetWeaver Composition Environment 7.1 is designed to enable fast innovation in building new applications. This tool supports the creation of new "delta applications" using the newest version of the Java standard, as well as Eclipse-based tools and advanced composition capabilities. These resulting business applications leverage the enterprise services that are delivered with SAP ERP and the enhancement packages on top of SAP ERP, and can be built by specialized software solution providers (SSPs) or by customers themselves.

The key benefit of SAP NetWeaver Composition Environment 7.1 is its much smaller footprint compared to the complete SAP NetWeaver stack, allowing faster change on the customer side and providing the opportunity for SAP to deliver even more innovative technology in the coming years – without affecting the stability of core application systems, such as SAP ERP.

SAP NetWeaver Process Integration 7.1

SAP NetWeaver Process Integration (PI) 7.1 is the successor release of SAP NetWeaver PI 7.0 (also known as SAP NetWeaver Exchange Infrastructure, or SAP NetWeaver XI). Like SAP NetWeaver Composition Environment 7.1, the focus of SAP NetWeaver PI – with its capabilities for message-based communication between different systems and companies and its cross-system process orchestration capabilities – is on broad support of the newest versions of open standards, including Web Services Reliable Messaging (WS-ReliableMessaging). It also extends its reach to high-volume scenarios that require higher throughput.

Enterprise Services Repository

Both SAP NetWeaver Composition Environment 7.1 and SAP NetWeaver Process Integration 7.1 can host the Enterprise Services Repository (ES Repository), which is the centerpiece of SOA governance. SAP built this repository to contain all the process and service definitions needed to build integration processes and composite applications on top of system and solution landscapes. The ES Repository also provides the means for customers and

Java and ABAP. There was also some evolution in the way SAP thought and talked about Java. Here is a short summary of where we are today:

- ABAP is and remains the primary language for business logic and applications created by SAP.
- From its construction, ABAP is on a higher abstraction level than Java, which means that it is better suited for building applications and less suited for lower-level technology layers, such as message handling or creating technical tools.
- Conversely, Java is closer to traditional imperative programming languages like C and C++ and therefore is more suitable for lower-level technology, such as integration tasks.
- Java is, in principle at least, open and standards based. As such, it has a broader community than ABAP could possibly have. Therefore, it is attractive in some situations – such as when portability or a very small footprint matters most – to write applications in Java, even if the task would be somewhat easier with more business-level languages like ABAP.
- An extensive community uses Java for a variety of purposes, from low-level technology pieces up to applications, especially UIs (in the early days, Java was focused on more interactive Web interfaces). So it remains important for SAP and customers to be able to leverage Java-based software.
- Because Java has many benefits for the creation of UIs and composite applications on top of existing services and processes, SAP has opted to build the SAP NetWeaver Composition Environment on top of Java, offering Java-based development, model-based composition, and combinations of the two approaches as needed.

The Evolution of the "Double Stack"

In the early days of SAP R/3, ABAP was the only language in the SAP application space – the use of C and C++ on the kernel level was mainly invisible to the customer.

The advent of Java introduced not only a new programming language, but also a new development environment, new administration functions, and new tools and techniques – and new needs – for diagnostics and debugging.

It was not easy to hide this additional complexity. Combining ABAP and Java into a single application server infrastructure could only partially solve this problem. The driving idea was to "hide Java behind

Where and when do you optimally use ABAP and Java? And where is it beneficial to run these languages separately?

partners to define and model additional services and processes that are not provided by SAP.

SAP NetWeaver Mobile 7.1

This is a major new release with many important architectural enhancements. Key areas include:

- Higher scalability (i.e., support for a very large number of devices and high data volume on the local device)
- Rapid, model-driven development of mobile applications based on Web Dynpro and SAP NetWeaver Developer Studio
- Deployment, administration, and monitoring of a large number of mobile devices with low effort
- Reduced cost of upgrade due to new version management

For more information, refer to the Mobile Matters column "Overcome the Top 6 IT Challenges of Mobile Business: New Major Release of SAP NetWeaver Mobile Paves the Way" by Hansen Lieu and Arvind S. Pawar in this issue of *SAP Insider* (**www.SAPinsideronline.com**).

SAP Enterprise Search 7.0

The goal of Enterprise Search is to simplify searches across metadata, documents, and business objects. Enterprise Search finds and relates the right information for better decision making and features:

- The ability to search on structured and unstructured data
- Integration into the existing infrastructure
- Access through various UI channels

For more information, see "A Unified Search Across All Your Enterprise Information and Business Processes" by Karsten Hohage in the January-March 2007 issue of *SAP Insider* (**www.SAPinsideronline.com**).

Collaboration Features for SAP NetWeaver Portal 7.0

This is an add-on delivery that can be installed on top of existing SAP NetWeaver Portal 7.0 installations and provides:

- Web 2.0-based building blocks, such as discussion forums like the ones found on the SAP Developer Network (**http://sap.sdn.com**), including an intuitive user interface integrated into the portal, moderated forums, and search and email notifications on postings
- An AJAX-based portal user interface, offering fast and intuitive navigation and a personalized selection of roles (compatible with existing portal roles

ABAP," which turned out to be pretty challenging and not fully successful end to end. Some flexibility – a key benefit of Java – was reduced to simplify some aspects of configuration and administration.

The end result? Currently, some deliverables like SAP NetWeaver Process Integration and the SAP Solution Manager do rely on this dual setup. Others – like SAP NetWeaver Business Intelligence – will allow the distribution of ABAP and Java functionality on two different systems, although they still require fairly tight coupling in terms of configuration and maintenance. The same applies to SAP ERP self-services with their ABAP back end and Java front end.

A natural separation of duties has emerged for the ABAP and Java stacks: Service *provisioning* is the main focus on the ABAP side, and service *consumption* is the main focus on the Java side.

However, most customers prefer to run the ABAP and Java pieces on separate systems, even if those systems have to be tightly coupled from a lifecycle perspective.

Rethinking ABAP and Java for the Future

While there are ongoing discussions on how to solve this puzzle, a few patterns have emerged when considering how best to use ABAP and Java:

1. To focus on the respective, specific benefits of ABAP and Java from a low-level perspective (e.g., transaction management or usage of operating system threads), using separate systems or instances for ABAP and Java makes sense.

2. On the administration and maintenance level (or more generally, all aspects of lifecycle management), service-based unification is the way to go. By using services and the principles of enterprise SOA, the result will be a common set of interfaces, hiding different implementations that are optimized for specific purposes. This will reduce TCO as well as the cost of learning for customers, and will also address issues that could not be solved with the dual-stack approach.

3. Fast-changing pieces of the overall SAP NetWeaver platform, like the SAP NetWeaver Composition Environment, are better served by an open, community-based technology stack, with Java as the programming language and Eclipse as the main driving force.

4. Reliable, long-lived application processes are better served by the proven, robust, and stable ABAP environment.

So we come to a natural separation of duties:

- Service *provisioning* is the main focus on the ABAP side

- Service *consumption* is the main focus on the Java side

With this approach, any interface between the two stacks will be open and standards based – and well able to hide the differences between the languages, as well as the stacks surrounding these languages.

Final Recommendations

Taken together, the recommendation to run ABAP and Java pieces in separate systems – when possible and where supported by SAP at this point in time – goes hand in hand with the recommendation to run stable parts of the landscape separately from fast-moving and more innovative parts.

In both cases, the goal is to enhance the stability of your technology and processes while maximizing the opportunities to move forward and explore new possibilities in your business. All are important considerations as you plan for the future of your system landscape.

AdditionalResources...

...from SAPinsider

- "Under Development: SAP's New ERP Strategy for Developers: 4 Ways to Make the Most of Your SAP ERP System" by Karl Kessler (*SAP Insider*, July-September 2007, **www.SAPinsideronline.com**)
- The **Administration and Infrastructure 2008** conference in Orlando, March 26-28, 2008, offering in-depth strategies for planning your SAP ERP and SAP NetWeaver landscape (**www.sapadmin2008.com**)