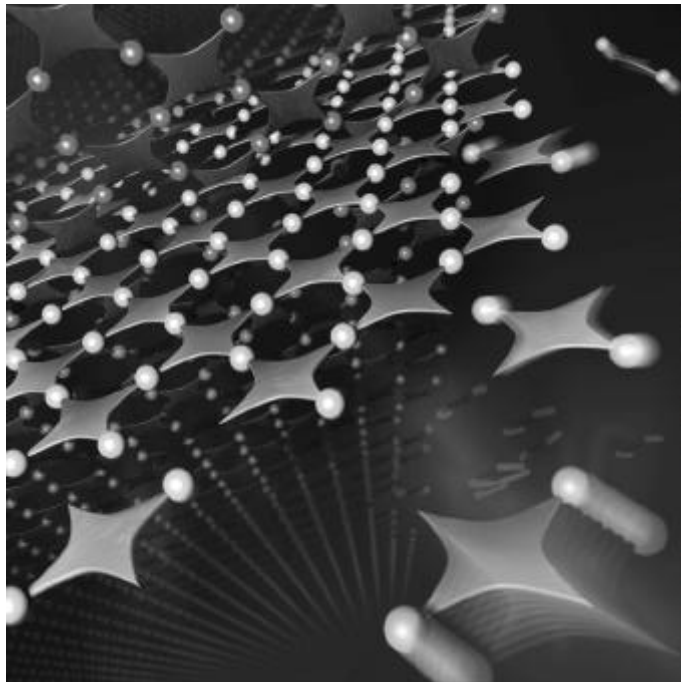


R/3<sup>®</sup> System

Benefits of the Business Framework



©Copyright 1997 SAP AG. All rights reserved.

Written by Technology Marketing, SAP AG

No part of this description of functions 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®, NT®, EXCEL® and SQL-ServerV are registered trademarks of Microsoft Corporation.

ActiveX™ is a trademark of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

OSF/Motif® is a registered trademark of Open Software Foundation.

ORACLE® is a registered trademark of ORACLE Corporation, California, USA.

INFORMIX®-OnLine *for SAP* is a registered trademark of Informix Software Incorporated.

UNIXV and X/Open® are registered trademarks of SCO Santa Cruz Operation.

Java™ is a trademark of Sun Microsystems, Inc..

ADABAS is a registered trademark of Software AG.

SAP®, R/2®, R/3®, RIVA®, ABAP®, SAPoffice®, SAPmail®, SAPaccess®, SAP-EDI®, SAP ArchiveLink®, SAP EarlyWatch®, SAP Business Workflow®, ALE/WEB®, SAPTRONIC® are registered trademarks of SAP AG.

## Contents

<b>Benefits of the Business Framework</b> .....	<b>3</b>
1. Introduction .....	3
2. Componentization of standard business software .....	4
2.1 The software industry has a maintenance problem not a development problem.....	4
2.2 Business and technology change at a different pace.....	4
2.3 "Separation of concerns" - the number one design principle .....	5
2.4 Speed and flexibility are crucial distinguishing factors .....	5
2.5 The whole is more than just the sum of the parts .....	5
3. The Business Framework .....	6
3.1 Business Components .....	7
3.2 SAP Business Objects .....	8
3.3 BAPIs .....	10
3.4 Integration .....	11
4. Conclusion .....	13
5. Bibliography .....	14

# Benefits of the Business Framework

As a vendor of enterprise business software with more than 10,000 installations of the R/3 System, SAP fulfills a very diverse range of company-specific software requirements. R/3 is an integrated software system that supports a variety of different applications from logistics through financial accounting to human resources. As the pressure of competition becomes greater and greater, innovation cycles in the IT industry become shorter and shorter; at the same time, the move toward market globalization means that software must become increasingly flexible, integrated and open. SAP has, for this reason, developed the Business Framework architecture which enables the R/3 System to be integrated in a family of software components.

## 1. Introduction

As a vendor of enterprise business software, SAP uses and develops new technologies in order to secure crucial competitive advantages for companies attempting to conquer shares of the market as well as capture new markets. The benefits to be derived from the increasingly extensive use of software components are outlined below:

- Component-based technology provides companies with a reliable means of implementing new functionalities smoothly and quickly. New components can work together with existing components in an integrated solution. **Fast**
- Existing solutions can be reconfigured extremely flexibly and, above all, dynamically. A software component is replaced with a new version without having to adapt the other components. The key benefit for companies that have to act as quickly and as flexibly as possible is that only parts of a solution and not the whole solution are transferred to a new release. **Flexible**
- Adopting a component-based approach permits applications from different vendors to be used for a defined task. Companies are thus free to choose the software products which complement their existing solutions. **Open**
- Industry-specific requirements can be met easily and flexibly by integrating components that are specially designed for the industry in question [5]. **Industry-specific**
- Last but not least, a component-based solution can be easily expanded to take account of customer-specific functionality because the interfaces of prefabricated components can also be used by customer-specific components. **Expandable**

This is why SAP has, for some time now, been using a component-based technology which is designed to enable the evolution of the R/3 System into the Business Framework. SAP uses both existing and new technologies for



this purpose. Companies which are already using R/3 productively can thus benefit directly from this development. Companies that have not yet implemented R/3 can, however, also derive strategic advantages from the Business Framework. These will be discussed later on.

## **2. Componentization of enterprise business software**

A number of different fundamental considerations played an important role in the development of the Business Framework strategy. These will be presented as five assumptions which take a closer look at the basic requirements of companies that use enterprise business software.

### **2.1 The software industry has a maintenance problem, not a development problem**

Companies that opt for enterprise business software tend to think in the long term. This type of software generally has a long life cycle, which explains why investment protection is an important issue for such companies. The technology on which the software is based and the functionality implemented in the software are developed as part of a continuous process. It goes without saying that the companies concerned would like to profit from these ongoing developments. The result is a maintenance problem: How can a company, with an IT infrastructure that has evolved over the years, continue to take advantage of the new technologies and enhancements offered by the enterprise software vendor? This question is particularly relevant if the software has been modified to incorporate company-specific functions.

### **2.2 Business and technology change at a different pace**

Enterprise business software is impacted by the way in which technological innovation cycles are becoming shorter and shorter. Examples of this include client/server systems [9], the Internet, Java as well as object-oriented communication mechanisms. This list could, in theory, be continued indefinitely, but there is a danger of overlooking one very important aspect: The business processes and basic principles supported by this abundance of technology have themselves hardly changed. For example, enterprise business software still continues to support order receipt as a business process, the only difference being that order receipt processing is also possible from a Java applet via the Internet or via DCOM from a Visual Basic program in conjunction with the standard software application. The business relationships (customer purchases product from vendor and receives an invoice) have remained unchanged.

These software components provide companies with direct access to the world of electronic commerce. The necessary technical innovations are implemented in the form of components without interrupting the business processes.

### **2.3 "Separation of concerns" - the number one design principle**

Since business changes more slowly than technology, it is appropriate to separate the responsibilities for different tasks within the enterprise software. The advantages of this can be easily illustrated using an example from the aircraft industry: Certain parts of a commercial aircraft are used for propulsion, others for the comfort of the passengers and others for navigation purposes. If a technical improvement is made to the navigation system of the aircraft, an existing component can be replaced with a new one without having to change the other components. This "separation of concerns" can also be applied to software: Examples can be found in interface technology, message brokering, programming languages, databases and, of course, in a wide range of applications such as logistics, financial accounting and human resources. As in the aircraft industry, software components can only be combined if they are based on a common design. In the same way as an aircraft must be viewed not in terms of its individual components but by the way in which these components interact, software components must speak a mutually understandable business language to ensure that they cooperate with each other successfully.

### **2.4 Speed and flexibility are crucial distinguishing factors**

In the past, companies basically had two possible ways of using software to support their business processes. The software could either be purchased as a complete standard package or could be developed and maintained in the form of a solution tailored to the requirements of that particular company by the in-house IT department [6]. The first alternative offers two strategic advantages: Firstly, purchased software can be implemented considerably faster and, secondly, there is no need for an expensive in-house development department. On the other hand, a company can only fulfill all of its IT requirements if it develops its own in-house applications or combines existing software products in accordance with its specific requirements. The challenge in the area of enterprise business software is to combine the two alternatives, that is the speed of a packaged solution and the openness and flexibility of a tailor-made in-house system.

### **2.5 The whole is more than just the sum of the parts**

The componentization of business applications is becoming increasingly important not just within individual companies. Components also work together across different companies to add a new dimension to communication and information. A key term here is "value chain", which runs from the supplier to the manufacturer through to the customer and along which different companies cooperate. The Internet is a particularly good example of a basic platform on which the distributed components of different companies can interoperate. The result is a cross-company infrastructure which is unprecedented in terms of its efficiency, degree of integration and availability of information [12].

### 3. The Business Framework

These five assumptions support the underlying concept of the Business Framework which offers a business solution for companies of all sizes in all branches of industry. This solution comprises integrated software components and features a flexible infrastructure that enables a company to respond to new requirements quickly without interrupting the course of business. The aim of the Business Framework is to provide SAP customers with greater flexibility and agility while maintaining the existing level of system integration [2]. The Business Framework focuses primarily on the first assumption "The software industry has a maintenance problem, not a development problem". Companies can use the Business Framework to combine individual components such as "Human Resources", "Consolidation", "Availability-to-Promise", and so on, even if these components do not belong to the same R/3 release. This means that a company about to upgrade its Human Resources component will not be forced to upgrade the other components as well.

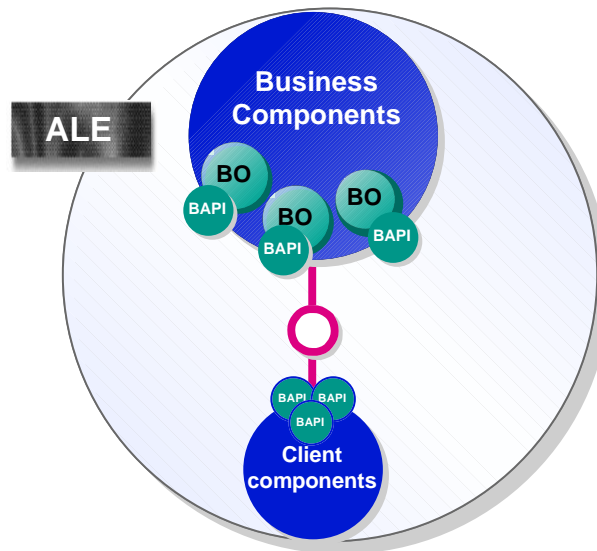


Figure 1: The Business Framework

In order to make this possible, SAP uses both proven and new technologies (see Figure 1):

- Business Components**
  - Business Components:**  
A Business Component supplies a dedicated, encapsulated business functionality via stable interfaces. A component of this type has its own development, implementation and maintenance cycles. A number of these components can run on their own dedicated databases.
- BAPIs**
  - Interfaces:**  
The Business Components communicate via BAPIs, a new object-oriented interface technology. BAPIs (Business Application Programming Interfaces) are methods which are based on SAP Business Objects, and offer a new level of interoperability between network-enabled, encapsulated Business Components from SAP and other vendors.

❑ **Integration technologies:**

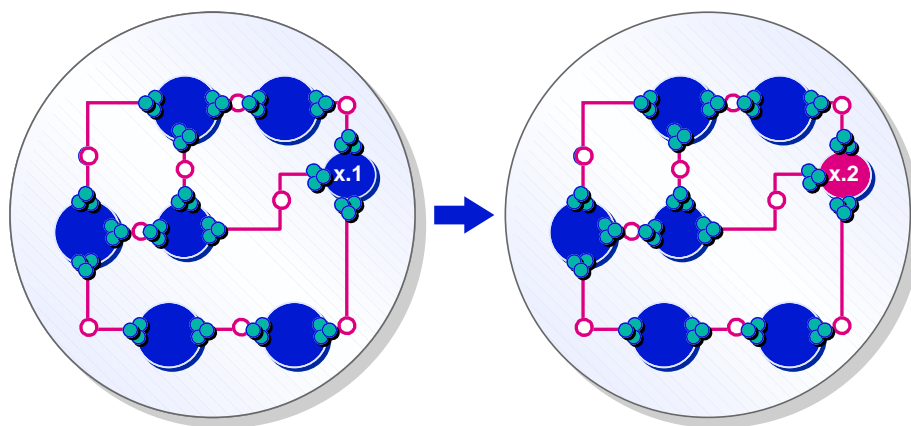
SAP uses Application Link Enabling (ALE) and SAP Business Workflow to ensure full integration of the system. This permits cross-component mapping of business processes and means that, from a business point of view, it makes no difference whether a process is handled in a single R/3 system or by a number of cooperating Business Components.

ALE

In line with the assumption "Speed and flexibility are crucial distinguishing factors", Business Components can be implemented fast by means of model-based customizing using the Business Engineer [3, 7]. SAP achieves flexibility through an open system and by supporting de facto and de jure standards such as COM/DCOM from Microsoft, Java from Sun Microsystems, CORBA from the Object Management Group, OAGIS from the Open Applications Group, UML from Rational Corp., and so on. The degree of success already attained by opening up the R/3 System is demonstrated by SAP's 230 or so partners who offer complementary software. When combined with R/3, this provides greater flexibility and speed for the companies concerned.

### 3.1 Business Components

Business Components supply the user with a specific business functionality in an independent software product by means of object-oriented interfaces. Release 3.1 included 28 different components for Internet applications, for example, for sales order creation via the Internet, Internet KANBAN production control or for creating requirement requests. Release 4.0 will contain a further 15 Business Components which, rather than providing new functions as in the Internet applications, will make existing R/3 functionality available in the form of components. Examples of this type of Business Component include the Business Engineer, Human Resources, Availability-to-Promise, Product Data Management, Consolidation and Pricing.



**Figure 2: Evolutionary introduction of new technologies**

The components in a software solution can use a wide range of different technologies. A number of components can, for example, be implemented in Java and distributed on the Internet. Others use the object-oriented enhancements of the programming language ABAP [11]. The evolutionary introduction of new technologies, that is, one component at a time, is made possible

Evolution



by the componentization of the overall system (refer also to Figure 2). Development of a completely new overall system which would require full migration of the current R/3 System can thus be avoided.

**Principle** Have you ever tried to make your own jigsaw puzzle? Imagine that you have a blank sheet of paper in front of you together with a pencil and a pair of scissors. Where would you start? You would almost certainly begin by drawing a picture before cutting up the paper. It would probably never enter your head to first cut the paper up and then draw a picture on the individual pieces because the "overall picture" would not fit together. The first method is obviously the right one and the principle behind it is implemented in the Business Framework.

**R/3 Reference Model** The transition from R/3 to a family of software components is taking place in accordance with the R/3 Reference Model, as this contains the overall picture upon which the componentization of the Business Framework is based. All business-specific data, that is the R/3 business rules and business processes as well as the SAP Business Objects, are stored in the R/3 Reference Model. The assumption "The whole is more than just the sum of the parts" is relevant in this case because it is not just within a company that components can map integrated business processes. This interaction is also possible along the value chain between components that are installed in different companies. An example of this is an Internet KANBAN application which is used by a supplier to check the KANBAN status of a manufacturer via the Internet. This makes it easy for the manufacturer to request a just-in-time delivery of material by the supplier.

### 3.2 SAP Business Objects

**Advantages of a top-down approach** The business level of R/3 is the focal point of the SAP object strategy. As of Release 3.0, the R/3 System incorporates over 170 different SAP Business Objects which are documented in the Business Object Repository [4]. In order to adapt object-oriented technology, SAP adopted a "top-down" approach which involves encapsulating business data and functions that already exist in the system in objects. This means that the business objects created can be used where they are of the greatest, that is direct, benefit to the company:

- Speed in implementing the software because business objects provide a fast and straightforward means of understanding business procedures when modeling and configuring business processes [10].
- Flexibility in the implemented systems through the integration of easily configurable workflows on the basis of business objects and SAP Business Workflow.
- Openness thanks to the object-oriented BAPIs of Business Components that use diverse communication standards even across system boundaries [8].

Implementing the object methods themselves in an object-oriented programming language does not take place until the very end. The stability of the object interfaces enables different programming languages to be used, including the new object-oriented ABAP enhancements.

ABAP objects

The alternative to a "top-down" approach is a "bottom-up" approach. This type of strategy, which has been propagated for some time now although it has not proved entirely successful for developing large software systems, focuses on technical objects that are implemented by means of an object-oriented programming language. A major disadvantage, as compared with the "top-down" approach, is that the development process has to start basically at the end. With "bottom-up" strategy, a software system has to be completely rewritten because the existing functionality cannot be used. Consequently, it will be some time before a system of this type is capable of competing with existing systems and their rich, comprehensive functionalities.

Disadvantages of a bottom-up approach

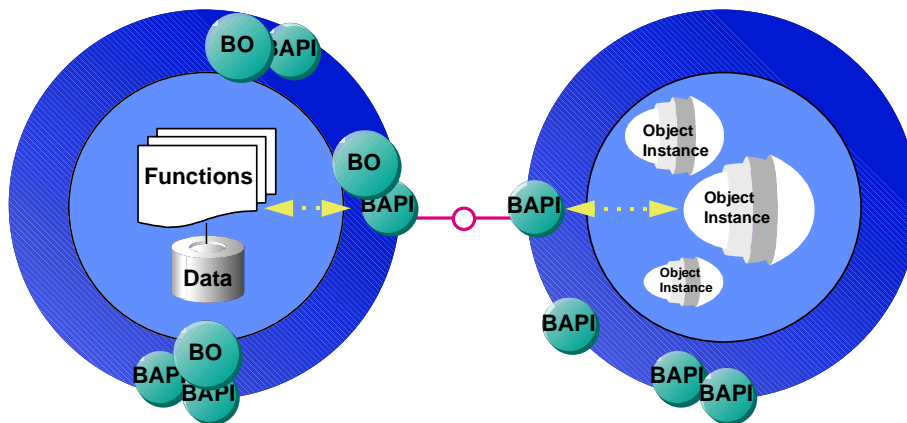


Figure 3: Object-oriented interfaces of Business Components

SAP separates business from technology by means of business objects, the stable elements in the Business Framework. SAP Business Objects, such as "Employee", "Application", "Applicant" and so on, represent real business entities which, compared with technology, develop at a much slower rate. This is why the interfaces of Business Components were structured using business objects and the SAP Business Objects mentioned above are visible in the interface to the "Human Resources" component (refer also to Figure 3). This allows the communication technology used between objects to evolve (COM/DCOM, CORBA, and so on) and new technologies (Java, Visual Basic, and so on) to be used within an object without the business processes having to be changed.

SAP Business Objects as stable elements

In very basic terms, a Business Component represents the functionality of a set of semantically related SAP Business Objects which are supplied to a company in the form of a separate, executable software product.

### 3.3 BAPIs

Business Application Programming Interfaces (BAPIs) are open, object-oriented interfaces which permit straightforward and, above all, stable communication between software components in line with basic business principles [1]. They enable the business processes in the R/3 System to be accessed easily and thus provide a basis for integrating software components (including the customer's own developments or products from third-party vendors) irrespective of the programming languages and communication mechanisms used.

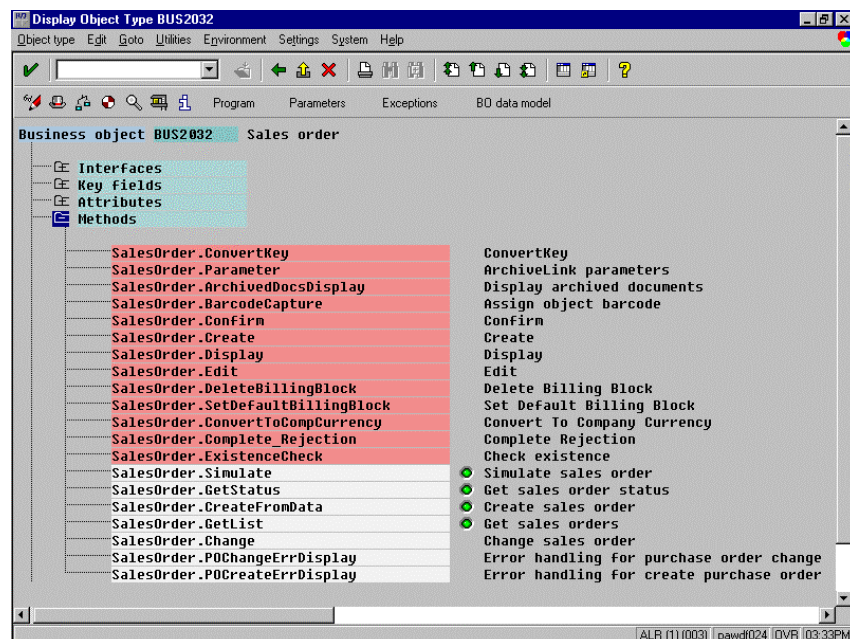


Figure 4: The BAPIs of the SAP Business Object "SalesOrder"

**Examples** BAPIs are methods of SAP Business Objects. BAPIs of the SAP Business Object "SalesOrder" are, for example, the two methods "SalesOrder.CreateFromData" and "SalesOrder.GetStatus". BAPIs provide R/3 functionality at an object-oriented interface and thus form the basis for the SAP Internet strategy or for customer-specific extensions to R/3 in languages such as Visual Basic or Java. BAPIs, therefore, play a very important role in the process of creating component software within the Business Framework (refer also to Figure 4).

**Principle** Yet what distinguishes a BAPI from a conventional interface? The interfaces used to date have one major disadvantage which we shall refer to as the "postcard effect". With a postcard, no distinction is made between the contents and the mode of transport. A text is written on a card which is, at the same time, the information carrier. If the information carrier changes, the recipient must use a different method to read or process the text. R/3 Release 3.1 was supplied with more than 150 BAPIs that offer a new quality in standard business interfaces. These open BAPIs, which were also published on the SAP Web site [1], combine the most important advantages of object-oriented communication, namely the encapsulation and abstraction of internal implementation details, with the business know-how of SAP.

The basic difference between BAPIs and conventional interfaces is the way in which the business contents and the underlying communication technology are separated from each other. This can be compared with a letter in an envelope; the "postcard effect" is, in this case, eliminated. It is quite easy to place a letter in another envelope or to use different or new communication technologies such as Microsoft's COM/DCOM or the CORBA standard from the Object Management Group. The envelope itself is irrelevant as far as understanding the contents of the letter is concerned. In this example, BAPIs correspond to the letter - they are independent of the programming languages (used to implement the functionality represented by the BAPI) and the communication mechanisms (used to invoke the BAPI).

This clear separation of business and the underlying technology enables new technologies to be used in innovation cycles that are becoming shorter and shorter without having to change the more slowly developing business represented by the BAPIs. BAPIs thus provide a means of translating the "Separation of concerns - the number one design principle" into action.

### 3.4 Integration

Integrating the different Business Components is an extremely important process for the companies concerned. Application Link Enabling (ALE) and SAP Business Workflow, for business processes which are tailored more specifically to individual requirements, implement the integration rules in the active R/3 System. ALE is already being used by many different customers. In Release 3.0, ALE was used to integrate business processes which are executed on several R/3 systems that are physically remote from each other. For example, a company's head office in Europe, its Sales and Distribution division in the USA and one of its plants in Australia could each use a separate R/3 System. Reliable, message-based integration is made possible by ALE even if the releases of the systems are different. This means that a manager can obtain all relevant corporate data immediately and seamlessly even if the original data was not administered on his local R/3 System. The tools and technologies provided in Release 3.0 for integrating different systems are further enhanced in the Business Framework for integrating Business Components.

SAP designs, develops and tests the Business Framework as an integrated system and supplies it to the companies concerned as a set of Business Components. This integrated approach enables the Business Components which are made available separately to be recombined in a straightforward manner to form an integrated solution. This is illustrated in Figure 5. The customer here has opted for a solution which includes the Human Resources component (HR) and Availability-to-Promise (ATP) as independent Business Components. The customer can upgrade the Human Resources component

**Development, shipment,  
and configuration of  
components**

to a new release independently of the Logistics (LO) and Financial Accounting (FI) components. Furthermore, they are also able to perform Availability-to-Promise processes efficiently using hardware, a database and an operating system platform specially optimized for this purpose.

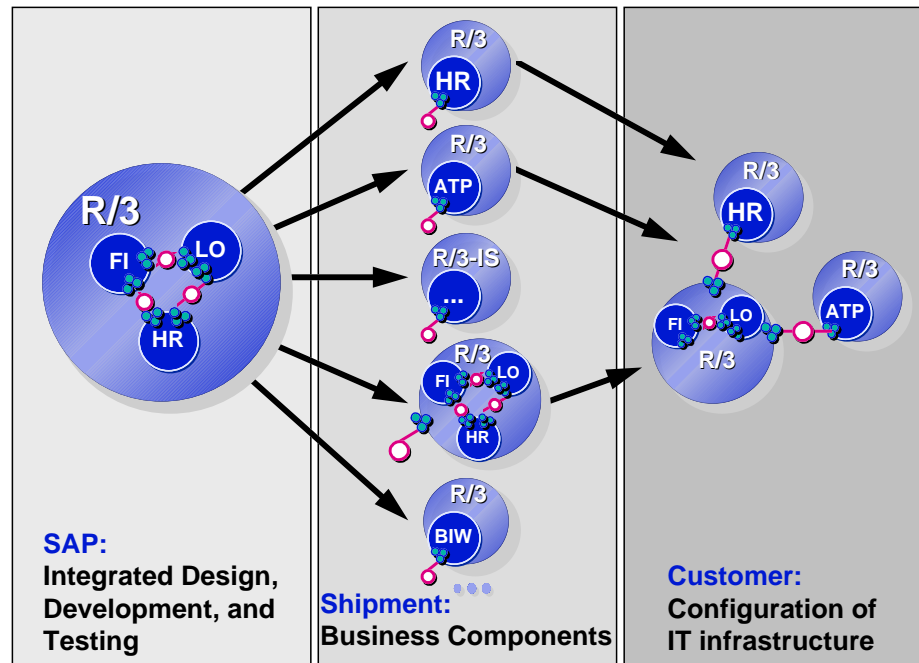


Figure 5: Development, shipment and configuration of Business Components

Only components with a common business context can communicate easily with each other. This is why SAP developed the Business Framework, which creates precisely this open standard business infrastructure:

- ❑ The Business Components are based on a joint business concept and have a common business context.
- ❑ The R/3 Reference Model describes the common features of all SAP Business Components and establishes a joint conceptual basis for all entities.
- ❑ The joint business infrastructure contains basic business functions such as a factory calendar and supports several currencies and country versions.

### Semantic synchronization

The Business Framework was not developed in line with a "best-of-class" approach where any components from different manufacturers can be combined. The Business Components in the Business Framework are based on common business principles in accordance with which business processes are synchronized semantically. The components "understand" each other because they have the same business background. If, for example, HR and FI are run as separate Business Components, the organizational structure of the company must be modified in both components even if one of them was not available at the time the change was carried out. This semantic synchronization would not be possible without integration technology.



## 4. Conclusion

Value-added potential is created by striking a balance between Business Components, interfaces and integration. The number of components must, on the one hand, be large enough to attain a significant level of mobility and agility but, on the other hand, small enough to keep the system that has been created manageable. The interfaces must, on the one hand, be integrated deeply enough to provide a high degree of flexibility but must, on the other hand, be open enough to enable the standard software and components of other manufacturers to be reused easily. Integration must, on the one hand, be tight enough to secure realtime business advantages but, on the other hand, loose enough to avoid bottlenecks occurring as a result of too many interdependencies between the underlying components. The Business Framework offers a balanced solution that maximizes the advantages of this approach.

Using Business Components is just one way in which SAP offers companies a completely new level of agility, flexibility and integration. Preintegrated Business Components can be implemented as and when required. A new, higher degree of interoperability is attained through BAPIs, which are new business-oriented interfaces. Finally, Application Link Enabling allows Business Components from SAP and other vendors to be integrated.

More information on SAP and the Business Framework can be found on the Web at <http://www.sap.com>.

**The Business Framework -  
a balanced solution**

**Visit us!**



## 5. Bibliography

- [1] Deimel, A.; Graf P.: "The BAPI-Catalog", Web site, [www.sap.com](http://www.sap.com), SAP AG, 1997
- [2] Graf, P.; Tolkmit, G.: "The Business Framework", White Paper, mat. no. 50016302, SAP AG, 1997
- [3] Jansen-Niedermaier, C.: "R/3 Business Engineer", White Paper, mat. no. 50014850, SAP AG, 1997
- [4] Johnson, M.: "The SAP R/3 Business Object Repository", White Paper, Benchmarking Partners, 1996
- [5] Kagermann, H.: "Lean Government", SAP INFO Focus Public Sector, mat. no. 7503675, SAP AG, 1997
- [6] Keller, E.: "Build vs. Buy: New Decision Criteria for Applications", Research Note, Gartner Group, 1997
- [7] Meinhardt, S.; Popp, K.: "Konfiguration von Geschäftsprozessen bei der Einführung von Standard-Anwendungssystemen". In: HMD Theorie und Praxis der Wirtschaftsinformatik, Heft 193, S. 104-122, 1997.
- [8] Orfali, R.; Harkey, D.; Edwards, J.: "The Essential Distributed Objects Survival Guide", John Wiley & Sons, 1996
- [9] Plattner, H.: "Client/Server Architecture". In: "Handbuch Informationssysteme", Gabler Verlag, Wiesbaden, 1993
- [10] Taylor, D.: "Business Engineering with Object Technology", John Wiley & Sons, 1995
- [11] Wiegert, O.: "Object-Oriented Concepts of ABAP", White Paper, mat. no. 50015999, SAP AG, 1997
- [12] Zencke, P.: "A logical extension of the R/3 architecture", SAP INFO Nr. 50, SAP AG, 1996