

Java-Based Applications and SAP's Business Framework

Tejinder Singh¹, Associate Prof. Dr. Ruchira Bhargava²,

¹Research Scholar of Computer Science, JJTU, Jhunjhunu Rajasthan, Department of Computer Science,
Lecturer of BFC, Bathinda (Punjab),

²Ph.D Coordinator, HOD-CSE/IT, JJT University, Jhunjhunu Rajasthan

Abstract: - As the leading vendor of enterprise business standard software, SAP has recognized the need to adapt their R/3 system to current trends in software development and to meet market needs for Speed of development, flexibility, openness and interoperability. In this paper, we first present SAP's approach to object-oriented and component-based technology by describing the Business Framework, the concepts of Business Objects, BAPIs, and the Business Object Repository. On this basis, we then analyze current communication architectures and products enabling the interaction of external Java-based software applications with SAP R/3, point out the advantages and disadvantages of different solutions and finally elaborate on potential strategies and steps for driving the evolution of SAP R/3 in order to further increase interoperability, openness and flexibility.

Keywords: - Enterprise, business, interoperability, flexibility.

I. Introduction

The integrated business software system SAP R/3 Covers a wide range of business software requirements from accounting and controlling through Human resources management to project management. Being SAP's vital software product, R/3 has to be reworked continuously in order to meet the changing market needs and to face competition. Recognizing the demands for interoperability, flexibility, openness, and the trends in software development being based on object-oriented and component-based paradigms, SAP has designed a new architectural framework for R/3 which has made the relatively monolithic system open up and become an integration platform for using so-called "business objects" across business processes and applications. This component- and business object-oriented architecture is called the Business Framework. While a high level of system integration has been maintained in the Business Framework, it has become possible to recombine business components such as "Product Data Management" and "Consolidation and Pricing" originating from different R/3 releases. Furthermore, integration with third party Components (e.g., produced by customers or partners) have become much easier. Figure 1 shows the basic architecture of the Business Framework. The Business Framework consists of:

II. SAP Business Components:

SAP Business Components represent a self-contained software application domain in the enterprise. Each business component consists of a number of SAP Business Objects, whose interfaces, the BAPIs, jointly constitute the interface of the component. Business processes can be implemented entirely within one business component, or they may cross several components. Because of the encapsulation of functionality and the accessibility through clearly defined and stable interfaces, each SAP Business Component has its own software Lifecycle, e.g., each component can be maintained independently of other components.

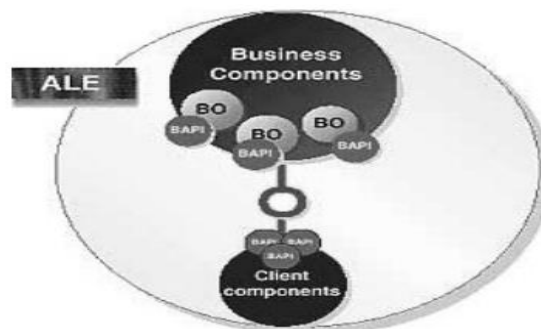


Figure 1: Business Framework

1.1 SAP Business Objects

SAP Business Objects are abstractions of real world objects from the business domain, such as "Employee", "Product", or "Customer", encapsulating implementation details of data and functionality. The concept of SAP Business Objects shows the Core characteristics of an object-oriented approach: business objects as instances of business object types, encapsulation of data and functionality, inheritance between business object types, and polymorphism in hierarchical inheritance relationships. SAP Business Objects are structured into the following layers (Figure 2):

- **Business object kernel** : contains object data and the core business logic,
- **Integrity layer:** contains business rules and constraints needed to manipulate the object data in a consistent way,
- **Interface layer:** defines the services provided by the object, the input event control and the output events published by the object.
- **Access layer:** describes the technologies (COM/ DCOM, Java, CORBA) that can be used to access the object.



Figure 2.1 Multiple Layers of an SAP Business Objects.

III. Java-Based Access to SAP R/3

Having presented the basic concepts of SAP's Business Framework approach, we will now concentrate on questions regarding the interoperability provided by this approach. Since Java is one of the most interesting and successful modern object-oriented programming languages, we will emphasize Java-based communication architectures and products for the interaction with R/3.

1.2 Characteristics of Different Communication Architectures

Choosing the "right" communication architecture is not trivial and plays an important role with regard to

- The integration of different programming languages, operating systems, and computer architectures,
- The integration of existing applications (compatible),
- Portability of source code, • Performance, and scalability. For a Java-based application, there are four possible technologies on which communication with the SAP R/3 System can be based:
- Remote Method Invocation (RMI), Java Native Interface (JNI), Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA).

3.2 Available Software Products for Java-Based Access to SAP R/3

For some time now, a number of products have been offered which enable the access of Java-based applications to SAP's R/3 System. Among them are:

- Visual Edge's Object Bridge and Madrid for R/3.
- IBM's Visual Age for Java Enterprise and
- SAP's Java Remote Function Call (JRFC).

Moreover, there are several libraries for other programming languages (such as C, C++, Delphi, or Perl) which contain programming constructs for the communication with the R/3 system. In the following, we focus on the presentation of the Java-based products.

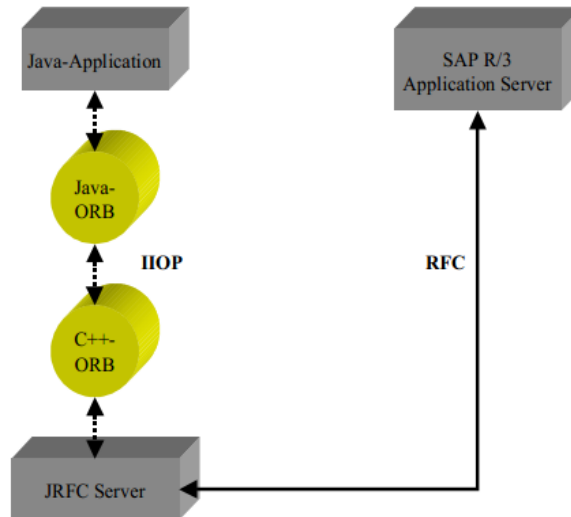


Figure: Access to R/3 via JRFC

2. Suggestions for Improvement

Although there are some first signs of interoperability and openness, the drawbacks stated before remain. In order to address these problems, one can think of a number of potential steps to be taken to driving the evolution of R/3. The following suggestions for improvements are not only directed to SAP as the manufacturer of R/3 itself, but also aimed at the great number of SAP customers and others who should jointly articulate their demands for increasing the openness of R/3 and its possibilities for integration with various platforms and programming languages.

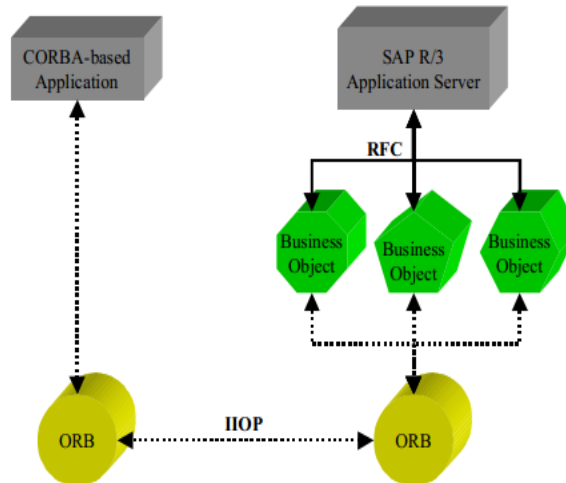


Figure 4. Direct connection to R/3 via

CORBA.

IV. Conclusion

After some years of paying lip-service to Java/CORBA without doing much to prove its Support, SAP has been engaging strongly in this area for some time now. An example of this is SAPGUI in Java. This ORBlet communicates with the SAPGUI server via IIOP. The SAPGUI server itself communicates with the R/3 application server using the SAP-specific R/3 Protocol. The great advantages of this application are platform independence, ease of use (provided that the user is able to handle a web browser) and the significant simplification of software distribution. A new version only has to be installed once and is then automatically distributed by the web server. The publication of IDL interfaces for SAP Business Objects and of the initial reference needed for communication could solve this problem and allow interoperability of R/3 and formerly unsupported platforms and programming languages such as Ada, COBOL or Smalltalk.

References

- [1] Benchmarking Partners (1996): "The SAP R/3 Business Object Repository"; in: Strategic Technologies, 1 (1) 1996
- [2] Hamilton, G., editor (1997): "JavaBeans API Specification"; ver. 1.01, Sun Microsystems Inc., July 24, 1997, <http://www.javasoft.com/beans/docs/beans.101.pdf>
- [3] Herrmann, J., Reipa D. (1999): „Der Monolith öffnet sich: eine objektorientierte SAP R/3-Schnittstelle für Java“; in: OBJEKTspektrum, 2/99, pp. 63-69
- [4] IBM (1999): General IBM web site containing information about Visual Age for Java and Access Builder for SAP R/3, <http://www.software.ibm.com/ad/vajava/sap.htm>
- [5] Korthaus, A. (1997): "Business Objects as Constitu-ents of Future Distributed Business Information Sys-tems"; discussion paper 1/97, University of Mannheim, 1997, <ftp://ftp.wifo.uni-nnheim.de/pub/PEOPLE/korthaus/CBOpaper.ps>
- [6] Microsoft Inc. (1999): "Distributed Component Ob-ject Model (DCOM)"; General Microsoft web site con-taining links to information about the DCOM Technol-ogy, <http://www.microsoft.com/com/dcom.asp>
- [7] OMG (1998): "CORBA/IIOP 2.2 Specification"; OMG Technical Document Number 98-07-01, <http://www.omg.org/corba/corbaiiop.html>
- [8] OMG (1998): "CORBA/IIOP 2.2 Specification, Chapter 3—IDL Syntax and Semantics"; OMG Technical Document Number 98-02-08, <ftp://www.omg.org/pubs/formal/98-02-08.pdf>
- [9] OMG (1998): "CORBA/IIOP 2.2 Specification, Chapter 4—ORB Interface"; OMG Technical Document Number 98-02-09, pp. 10 ff., <ftp://www.omg.org/pubs/formal/98-02-09.pdf>
- [10] OMG (1998): "CORBA/IIOP 2.2 Specification, Chapter 11—ORB Interoperability Architecture"; OMG Technical Document Number 98-02-16, pp. 14 ff., <ftp://www.omg.org/pub/formal/98-02-16.pdf>
- [11] OMG (1998): "CORBA/IIOP 2.2 Specification, Chapter 13—General Inter-ORB Protocol"; OMG Technical Document Number 98-02-18, pp. 33 ff., <ftp://www.omg.org/pub/formal/98-02-18.pdf>
- [12] OMG (1997): "Naming Service Specification"; OMG Technical Document Number 97-12-10, <ftp://www.omg.org/pubs/docs/format/97-12-10.pdf>
- [13] SAP (1997): "BAPI Introduction and Overview"; SAP white paper, Version R/3 Release 4.0, Dec. 1997, <http://www.sap.com/products/techno/bapis/edu/docu/caalbe/caalbe.htm>