

# Component Based Internet Development

Ronald Lips

*In the past few years, software development has been able to reuse custom made and/or third party software components in multiple applications by using a component based approach. The use of components has reached very deep into the way Internet applications are being developed as well. Coming from a mix of HTML and CGI based development, Internet applications have become more and more componentized. The use of third party components, combined with purpose-made custom components is now a more or less accepted way of developing these applications.*

This article focuses on the way the Baan Company and more specifically Baan WWWapps as part of the Baan Development organization, is developing Internet applications using component based development. The article starts with a short introduction of the Baan WWWapps department and its mission. After that, general issues with regard to component based development are being discussed. The article then continues with a focus on components that are very closely related to Internet development, both at a functional level as well as at a technical level. It demonstrates a balance between 'Make' or 'Buy' of components, which is always the consideration to make when starting component based development.

## Baan WWWapps

Part of the Baan Development organization, Baan WWWapps is responsible for developing Internet applications that focus on the Extended Enterprise. The Extended Enterprise can be seen as the collection of all relations a company can have with other companies as well as the business processes that exist with regard to these relations. For instance in a Supplier relation between two companies, the internal requisitioning process in one company may eventually lead to a sales order in the ERP system

of the supplier that will fulfill the requisition.

The automation of these processes that are beyond the boundaries of a single company is the domain of Baan WWWapps. The Internet is used as the medium for supporting these business processes. To summarize, Baan WWWapps develops Internet applications for the E-Business and E-Commerce environment, both for a Business to Business situation as well as a Consumer to Business situation. The result is a suite of applications called Baan E-Enterprise.

## Multi-tier architecture

Closely related with component based development is the use of a multi-tier architecture in which components may exist at every tier of this architecture. Generally, a multi-tier architecture at a minimum has the following logically and usually also physically separated layers:

### *Presentation layer*

This layer is responsible for presenting data as well as logic that applies to that data to the end-user of the application. In an Internet environment, HTML combined with client-side scripting is used to achieve this. Examples of components

that exist in this layer are ActiveX controls and Java applets.

### ***Business Logic layer***

The Business Logic layer is responsible for the implementation of all rules that form the core of the application. The business rules are the functional domain of the application. Functional specifications are usually expressed in terms of business rules that need to be implemented. In an Internet environment, these rules are mostly implemented in server-side components, which are sometimes called business objects.

### ***Storage Layer***

The Storage layer is responsible for storing the data that is modified in the application into some form of persistent storage. Usually a relational database is used for this purpose. Components that exist in this layer may be provided by third parties. Examples of such components are Active Data Objects (ADO), ODBC and other data storage components.

The multi-tier application model is the natural model for Internet applications. There are a number of reasons for this:

### ***Avoid or minimize the installation of software at the client side***

Installation of software may mean long download times as well as setting additional requirements for the client machine. This means that the client machine needs to be dedicated to user interface logic, leaving the other layers on the server. Of course, when Internet connections become faster, this may become less important.

### ***Create browser independent applications***

Differences between browsers forces Internet applications to settle for the most common denominator when it comes to the user interface experience. Allowing installation of for example ActiveX controls on the client will enrich the user interface experience, but will reduce the reach of an application: certain parts of the user interface will not be able to run in certain browsers. Of course, the most common denominator is HTML, so this means limiting the client to user interface logic created using HTML. Business logic will require the use of more advanced programming

languages and therefore needs to be confined to the server.

### ***Scalability***

Scalability is very important for Internet applications, as they will need to be able to serve an unknown and in theory unlimited number of users. The multi-tier application model is a good context for creating scalability. Of course, the multi-tier model will not give you scalability (you need extra provisions for this, for instance the use of Microsoft Transaction Server), but it will give you the right model for creating scalable applications.

The majority of the logic of an Internet application will always be implemented at the server side.

## **Middleware**

Another issue closely related to component based development is the question which middleware needs to be used. Middleware is needed to allow the communication between components, but also the communication of for instance a user interface with components or, in the case of an Internet application, the communication between the Web server and components. The two most common choices that are at your disposal when it comes to middleware are the following:

### ***CORBA***

Common Object Request Broker Architecture. A middleware standard that has been set by the OMG. The OMG standard only exists on paper and is dependent on an implementation of third parties (for instance Orbix). This also poses a problem when it comes to using multiple implementations, as they may not always be compatible with each other. CORBA is available on multiple operating system platforms.

### ***DCOM***

Distributed Component Object Model. A standard that has been created and implemented by Microsoft for the Windows platform. Other vendors have taken it upon them to port this to other platforms like UNIX, Motif and the Macintosh. As this standard is implemented for the Win-

dows platform, this means that choosing DCOM will allow application developers to use the large amount of components available on this platform.

Baan WWWapps is using the Microsoft platform for application development. Development is primarily based on Microsoft Internet Information Server and Microsoft Site Server Commerce Edition. Therefore, being on the Microsoft platform, DCOM was chosen as the middleware for Baan WWWapps applications. This means that a component in the Baan WWWapps world is always a DCOM object, usually implementing some business logic running at the server.

## Choosing a language

Component based applications development when using either CORBA or DCOM as middleware, means that a binary standard is chosen for the implementation of components. A component can be implemented in the language of your liking and by adhering to the binary standard other components will be able to communicate with this component.

In the DCOM world this may mean that a Visual Basic script is able to communicate with a component that has been implemented in C++. To choose a language for component based development using DCOM means that any COM enabled language will do. The following are the most common choices (though there are more available):

### *Visual Basic*

Choosing Visual Basic means that you can use a development environment that gives you very high productivity combined with medium execution speed. Visual Basic components in the past few years have become fast enough though for the implementation of business logic, as the compiler used is the same one that is being used for Visual C++ development. Therefore, in most of the cases, Visual Basic will do. Visual Basic is not an object-oriented language and therefore constructions like inheritance are not possible. This does not have to be a big disadvantage, as DCOM does not provide you with these features either.

### *Visual J++*

Using Microsoft's variation of Java will allow you to combine the use of a good object-oriented language with a high productivity development environment. However, experience has learned that the Java-COM integration is not yet fully matured. Moreover, the future of the Java-COM integration is unclear at present, looking at the latest legal developments in the United States in the lawsuit against Microsoft with regard to this integration.

### *C++*

Not surprisingly, C++ will bring you the highest execution speed, but low to medium productivity. In the opinion of Baan WWWapps, C++ should only be used when execution speed is critical. Usually, this is not the case with the implementation of business logic.

In the Baan WWWapps development environment, Visual Basic is used as the standard programming language for creating components, which are predominantly components providing some specific piece of business logic (business objects).

Choosing a non object-oriented language for component based development should not be a big problem. There is a fundamental difference between object reuse and component reuse. Object-oriented languages can be very well used for implementing frameworks to simplify a specific task. For instance, one may implement a persistence framework that allows an application developer to make the state of an object persistent in a very easy way. In such a case object reuse is key. However, this does not mean that such a framework enables component reuse. Component reuse plays a role at the binary level and can be seen separately from frameworks.

Component reuse should also be a decision between Make or Buy. Very often generic components have been implemented by third parties and can be reused as they are.

## Team Development

In a multi-team development environment, components developed by one team are reused by other teams. This means that there needs to be a delivery

mechanism from the team implementing a component to the other teams. A number of key features are required in this respect:

### **Binary delivery**

Components should be delivered at the binary level only. No source code of the component should be delivered to other teams. This forces component developers to adhere to the contract of the interface and provide high quality components to other teams.

### **Sample code**

Components should be accompanied by sample code showing the way that specific component needs to be used.

### **Documentation**

Components should be accompanied by proper documentation describing the interfaces exposed by the component.

### **Easy installation**

When a developer decides to use a particular component, installation of the component, including creating all necessary Window registry entries, should be automated.

To achieve the requirements above, Baan WWWWapps uses the Microsoft Repository that is part of Microsoft Visual Studio. Using this repository, component developers can publish components including samples and documentation in the repository.

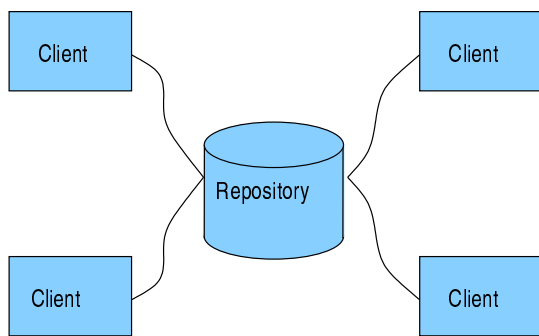


Figure 1: Development situation

Developers who want to use the component can install it from their development environment. The repository takes care of the installation. After installation, the developer can immediately use the component in Visual Studio. The repository can be im-

plemented using a SQL Server database as well using a Microsoft Access database. Baan WWWWapps uses SQL Server. The development situation using the Microsoft Repository is depicted in Figure 1.

## **Web Components**

In an Internet environment a number of component types can be distinguished. For Baan WWWWapps the most important types are listed below:

### **Business objects**

Business objects are COM objects that implement a certain piece of business logic and run on the server.

### **Design-time controls**

Design-time controls (DTCs) are ActiveX controls that only exist at design-time. At run time these controls result in VBScript or Java script that runs on an Active Server Page (ASP) on the Web server. DTCs are a feature of Microsoft Visual InterDev and can only exist on Active Server Pages at design-time.

### **ActiveX Controls**

ActiveX controls are user interface elements that provide some user interface functionality on the client. Example is a Calendar control. ActiveX controls are usually browser dependent.

### **Pipeline components**

A pipeline is a feature of Microsoft Site Server Commerce Edition and represents a specific business process in which one or more steps can be distinguished. An example is the processing of an order that has been submitted by the end-user. Each step in the pipeline needs to be implemented by a so-called pipeline component, which is a COM object that implements specific pipeline component interfaces. A number of standard pipeline components are available on the market. Customization of an existing or a newly created pipeline can be done by creating new pipeline components.

In the next few paragraphs a number of these component types will be discussed in the context of the features an E-Commerce / E-Business application needs.

The general context of a business object is depicted in Figure 2. As much has been said about business objects already in previous paragraphs, I will not discuss them any more at this place. This architecture is valid for all components that will be discussed here.

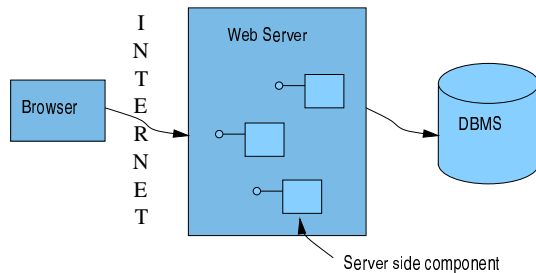


Figure 2: General context of business objects

## E-Business features

Basic features needed in the Baan WWWapps E-Enterprise applications are the following:

### **Authentication**

The ability to validate user name and password of a particular user and to provide security at this level.

### **Personalization**

Having identified the user, personalization is the ability to provide the user with Web content that match his/her preferences. Preferences should be stored in a user profile.

### **Ordering process**

The ordering process is central to E-Commerce applications. This process should be based on server side components in a pipeline.

### **Backend integration**

Integration of BaanERP with the E-Enterprise applications. On one hand, information in BaanERP should be made available on the Internet. On the other hand data created in E-Enterprise (for instance a Sales Order) should be inserted into BaanERP to follow the standard ERP procedure. Thus, E-Enterprise can use the power of BaanERP on the Internet.

Using component based development these features are implemented into E-Enterprise, using a mix of

Make or Buy: some components could be used 'as is'. Other components have to be developed.

## Personalization & Membership

The Personalization and Membership (P&M) features of Microsoft Site Server provides a set of components that can be used to identify the user and to provide the user with personalized content. P&M is fully integrated with Windows NT Security and offers a so-called membership directory that can be accessed using the standard ADSI interface that has been recently introduced by Microsoft and that will be part of NT 5.0.

The membership directory stores user data in the form of user attributes. New user attributes can be added in order to provide new information about the user. Examples of user attributes are E-Mail address, address information, user preferences etc. All user related data can be accessed through a COM object that is called the Active User Object (AUO) that abstracts all information in the membership database as well as access through ADSI. Information in the AUO can be accessed as simple key-value pairs (dictionary).

Security as it will be implemented in E-Enterprise 1.1 is typically a situation where Baan WWWapps has decided to use a third party component instead of implementing this functionality in a custom component.

## Pipeline components

As mentioned earlier, pipeline components are part of a pipeline that implements a specific business process, like submitting an order. In Microsoft Site Server Commerce Edition two types of pipelines are distinguished:

### **The Order Processing Pipeline**

As discussed before, this type of pipeline processes an order that has been submitted by an end-user.

### **The Commerce Interchange Pipeline**

The CIP is designed for exchanging data between two systems. The CIP consists of a Transmit

pipeline that takes data from the sending side and sends it to a Receive pipeline that exists on the receiving side. On the receiving side pipeline components exist that for instance take a purchase order that is sent by the Transmit pipeline and put it in to a backend system and make a sales order from it.

Already there are components that execute specific, basic tasks, for instance tax calculation or doing electronic payments. In Baan E-Enterprise 1.1 pipeline components are used to achieve backend integration with BaanERP, but also to implement sophisticated pricing algorithms that are not available in the standard components.

In this case both Make or Buy of pipeline components occurs. The general architecture of a pipeline is presented in the figure below.

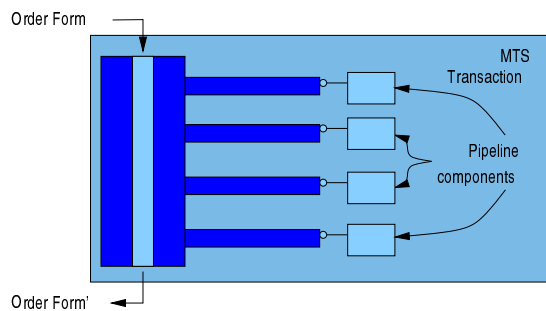


Figure 3: Architecture of a pipeline

## Backend Integration

Backend integration provides Baan E-Enterprise applications with access to BaanERP. In BaanERP so-called Business Object Interfaces (BOI) exist that provide an API to specific pieces of functionality of BaanERP. These BOIs can be accessed by using COM objects that act as proxies for a specific BOI.

The E-Enterprise Backend component provides simplified access to these BOIs by implementing functionality that allows the conversion of the result of a BOI call to XML. This result in XML can then be presented to the end-user using an XSL style sheet. A design-time control has been created that allows a developer or a system administrator at the customer site to create a new Web page that provides information from BaanERP. Using this DTC, the developer can indicate which BOI needs to be called and which data needs to be retrieved from BaanERP. Moreover, the XSL style sheet can be indicated that needs to be used to present the result to the end-user.

For this situation the decision has been made to implement the component ourselves. The architecture of the backend integration is shown in the picture below.

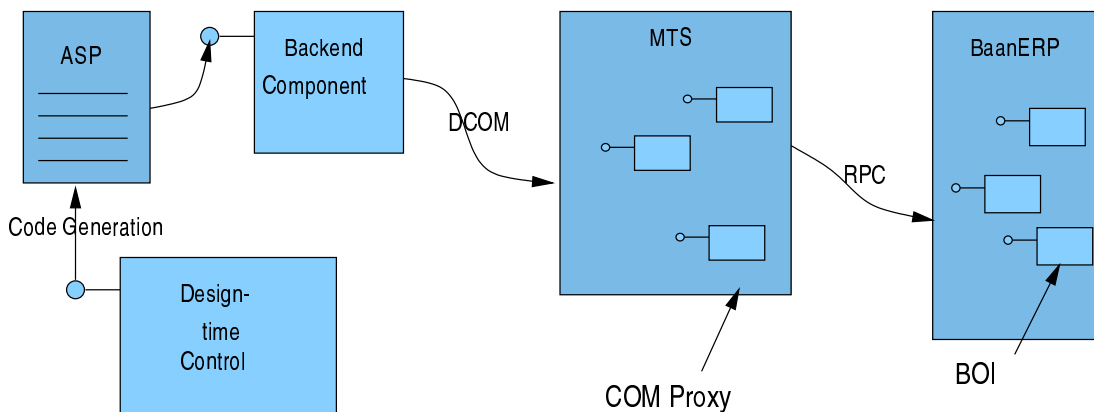


Figure 4: Architecture of the backend integration

## Summary

In this article an overview has been given of the way Baan WWWapps does component based development. In Baan WWWapps this development is completely based on the Microsoft development platform, using COM as middleware. It also shows that component based development is always a decision between Make or Buy. With exception of business objects, when using the Microsoft environment the Buy decision occurs very often because of the rich functionality of this environment. Developers then can focus on the implementation of core business logic of an application using infrastructure provided by third parties. Components are implemented at the level of business logic and providing access to BaanERP.

People who are interested in learning more about component based development using COM, can also contact the COM Interest Group (CIG). This is a user group that has been started by a number of COM developers in the Netherlands. The initiative is non-commercial, sharing knowledge is key to this group. Founding members of the CIG are Baan Company, BCI, Rational, Software AG, TU Delft and Microsoft. Information can be obtained by sending an e-mail to the author of this article (rlips@baan.nl).

*Ronald Lips is a Product Architect for Baan. He works at the Baan WWWapps department, which focuses on the development of Internet applications for the Extended Enterprise. Baan's web site can be found at <http://www.baan.com>*