

Technology Infrastructure

Research and Advisory Services

Application Deployment and Integration

RESEARCH PAPER

Epionet Epiowave

Abstract

Epiowave, which comes in two editions: Developer and Enterprise, is the generic name for a suite of tools used in the designing, building, and management of Web service applications and application components. Layer abstraction to the degree implemented in Epiowave is a prime factor in the ability to re-use, either directly or with little change, the applications developed in this environment. Not only does it demonstrate the conceptual understanding of Component-Based Development (CBD) but is technically implemented to a high degree. The components developed with Epiobuilder are COM+ and the whole solution is to that extent Microsoft-centric. Any organisation that is prepared to understand the complexities that development for Web services will bring, and also prepared to realise the benefits, will require applications developed with a true n-tier architecture. The best way of evaluating this solution is to repeat an application build that has used other tools and methodologies.

<p>STRENGTHS</p> <ul style="list-style-type: none"> • Builds manageable and personalised applications. • Abstracts application layers to a very high degree. • Breaks down the barriers between business requirements and development constraints. 	<p>WEAKNESSES</p> <ul style="list-style-type: none"> • Microsoft and COM+ centric.
<p>FUTURE POTENTIAL</p> <p>Has the potential to become a major entrant in the market. This would be further eased by the inclusion of Java.</p>	

► FUNCTIONALITY

Product Analysis

Epiowave, which comes in two editions: Developer and Enterprise, is the generic name for a suite of tools used in the designing, building, and management of Web service applications and application components. There are three elements contained within Epiowave:

- **EpioBusiness Server** – a runtime Web service application management environment.
- **EpioBuilder** – which contains the application framework architecture, the implementation system along with the Web services library, and the application builder.
- **EpioDesigner** – which is a complete toolset for requirement analysis, prototyping, and usability. It incorporates third-party products and has a strong emphasis on re-use backed by the incorporated process methodology.

The components developed with EpioBuilder are COM+ and the whole solution is to that extent Microsoft-centric. Whilst the focus for development and management is on applications based on the Web service model the lack of EJB support has to be considered as a weakness. However, two points should be made here; the first is that the transparency provided by Web services will remove some of those weaknesses, and secondly, it is the only obvious weakness found in what is a very strong product.

Central to the strength of Epiowave is the fact that it goes far beyond the development of applications, it provides an end-to-end solution from design to support of the applications, taking in prototyping, the actual development, testing, and deployment along the route.

The aim for modern development is to provide re-use capabilities for the developed applications, but this re-use can be limited if the context in which the user sits is ignored. In the broadest terms, this can mean that applications are developed based upon an expected 'average' user, and do not effectively take into account true individualisation. In the vast majority of instances, this view is forced by the granularity of the components available for application development.

Epiowave avoids this by providing the tools that can not only develop fine-grained components, but also the management capabilities of adapting those components at runtime based on the context or role of the user. Whilst there are many other development environments that can create fine-grained components that can be built into applications, Epiowave provides the environment for truly personalising those applications, which can be beneficial for reasons of security or productivity, and this has to be considered a major strength of the product.

Even before the application build and component development phase starts, Epiowave scores heavily by the use of UML modelling and Mind Mapping software to carry out requirement analysis. This is not an isolated process, as the methodology used provides a pathway that ends with the ability to abstract the required components from the design requirement. Components that already exist within the repository can be identified as being suitable either 'as is' or by adaptation. Those components that do not meet the requirements can then be quickly developed using EpioBuilder.

Product Operation

The EpioBusiness Server provides a central resource for application configuration, management, and deployment across the Internet backbone. Architecturally, it contains two core elements:

- System Services.
- Application Services.

System services manage those elements that are common to all applications; such as user management, security, and Web servers. The Epiobusiness Server System Services provides an environment at both the administrator and the developer level. The former is likely to use the environment for typical tasks such as user details and account information, and the management of group policies. System services fully supports hierarchical inheritance of any created character, with change in group policy being automatically filtered to associated group members. System services also includes a full set of reporting tools for administrators to log full details on application usage and system metrics.

Just as system services deals with the control and configuration of common system elements, so application services deals with common elements across application functionality. Whilst there will be no complete commonality of functionality across every application (it would be the same application), there are functionality elements such as interface code that has to be repeated across multiple applications. The abstracting of these elements into a separate layer for re-use is, in Butler Group's opinion, a clear indication of the effectiveness of the Epiowave solution.

Layer abstraction to the degree implemented in Epiowave is a prime factor in the ability to re-use, either directly or with little change, the applications developed in this environment. Not only does it demonstrate the conceptual understanding of Component-Based Development (CBD) but is technically implemented to a high degree.

One example, database connectivity, demonstrates the savings that can be made with this level of abstraction. Rather than create a connection within the application code, the database connection is created as an object that can be re-used with any future application. A similar methodology is used for file system management, and it is also possible for the developer to build an administrator interface for a specific application that is independent of any changes made.

Consideration of that last point more fully reveals a further strength of applications developed within Epiowave. Providing an administrator with the flexibility to manage applications without having to map application functionality separately to a management tool or console, not only creates a saving in time and expense, but will also become necessary when service-based delivery of applications becomes the norm. Logically, and rationally, applications will not only need to have management capabilities exposed, but they will have to do so in a manner that allows for instant administration. A user who calls a Web service application may want to manage certain aspects of that application, but if a new management interface implementation has to be created, it defeats the instant nature of Web services.

EpioBuilder could be considered as being the RAD part of the Epiowave product set, and although it contains the expected elements of such a tool; such as a development GUI, it goes beyond many RAD environments with the inclusion of additional functionality.

EpioBuilder utilises the EpioBuilder Framework, which is a template-based architecture for building Web-based applications. As already discussed, although not explicitly stated, it uses an *n*-tier design methodology. Split into three basic elements:

- The Business Object Framework.
- The Business Services Registry.
- The Presentation Framework.

The EpioBuilder Framework provides a template for creating components, a repository for objects, and a client neutral presentation layer.

The final module of Epiowave is EpioDesigner. Whilst this can be considered as being outside of the mainstream of application development *per se* it forms a most essential part of the whole package, although it is not a prerequisite for the other two modules. It has been implemented to remove the barriers between business application requirements and business application development.

Key elements within EpioDesigner include:

- Creation of an outline of business requirements.
- Building a model of those business requirements.
- Prototyping to establish user interface requirements.
- Prioritisation of business services.
- Documentation management of the full lifecycle design process.

EpioDesigner is currently in beta 2 and is expected to be released shortly. During preparation for this Research Paper, the beta version was reviewed and the functionality promised appears to be available and to be stable. The designer has been used within a live implementation as a further proof of its usability.

Product Emphasis

The uptake of CBD has been hindered by many factors over the past few years, but the introduction of the Web services model along with a greater understanding of the need to reduce the cost of application management as well as development is now ready to drive it forward.

Epiowave is positioned within the Web service space, but is not limited to the uptake of that model. Even at the most conservative level, it would be difficult to imagine why the benefits of this type of development (and that includes the design phase) should be overlooked. The creation of true *n*-tier applications is what the market needs, even if some of the market does not yet realise the fact.

► DEPLOYMENT

Application deployment is an administrator task handled by the System Services element of the EpioBusiness Server, and the applications can be managed centrally in terms of configuration, with the users able to create application instantiations based upon the policies dictated by the group in which they reside. The requirement for knowing what client devices are being used is removed by the internal client system analysis system, which details the device calling the application and makes alterations to both the system and the application (for this instance of the application) to take account of any device restrictions.

Security is handled through a number of layers implemented by the administrator, but the initial entry point to the system is through a single authentication point, which can be handled by the Windows NT administration capabilities. This reduces the complexity of account management for those organisations where application-user groups can be mapped to a central administration console. Where this is not possible, the single point authentication can be handled by EpioBusiness Server directly.

As the production of a bug-free application is still a phenomenon eagerly awaited, the implementation of a real-time debugger that automatically gathers live performance and behaviour information and reports on errors is an important element in any application deployment. EpioBuilder provides the tool for this, and it works against the specific application within a specific environment, effectively providing continuous testing.

Installation of applications is carried out with the application packager tool, and this utilises a Wizard-based installation tool for installing applications across multiple servers. The Wizard also handles application updates, back-up and restore, and version control.

Deployment against various data sources is available by two methods. Due to the true *n*-tier nature of the developed applications, the developer can create the connection objects for a specific data source within the application and this is used at runtime, or the connection object can be created as an application service to be managed by the administrator. Technically, there is no difference between the two as far as the development goes. Conceptually, if the application is to be deployed against a known data source then administration control can be removed (it can be restored at any time in the future); however, deployment against unknown data sources requires the intervention of an administrator to implement the correct connection object. This is an example of the benefits behind abstracting application services.

The *n*-tier nature of the developed applications increases the scalability of the deployed solutions. Epiowave applications are relevant to environments that need to support a large number of concurrent users, high-speed performance, and high-volume transactions. Multiple application servers can be utilised, with sessions being independent of any single instance.

► PRODUCT STRATEGY

Any tool or toolset within the application development market is going to have problems with a go-to-market strategy. The introduction of Web services has caused the major vendors to look at development tools in a new light and many are making them a central part of their Web service strategy. The good news is that this has at least created some traction in the market, and it is this that Epionet will have to leverage.

Building on work that Epionet has carried out for a major multinational corporation, the strategy is to demonstrate the benefits, already shown by the corporation's project, by carrying out similar application specific development implementations for other major organisations.

This will build an ROI model that will carry weight in the market, and provide the foundation for getting Epiowave noticed. This particular case study provides interesting figures, with the measurement being taken against a project undertaken with the corporation's current tools and methodologies and repeated with Epiowave.

The Epiowave development showed a gain in productivity of 650% and up to an 86% cost saving. There were also 80% less bugs in the finished Epiowave development. The deployment of the finished application was a simple exercise, yielding further benefits in terms of convenience and cost. Whilst there is always a reduced cost in repeating a project for proof-of-concept purposes, the figures shown above were weighted to take account of that fact, and even if that weighting was not 100% accurate it still shows a remarkable increase in productivity and an equally remarkable decrease in cost.

Epionet is also actively seeking technology partners to help bring Epiowave to market.

► COMPANY PROFILE

Epionet is headquartered in Dublin, Ireland; and comes from a project-based background, carrying out work for clients that included:

- Modus Media.
- Siemens.
- Bertelsmann.
- Build On-Line.

In 1998 the focus shifted away from this work and the seeds of Epiowave were sown. 1999 saw the implementation of the concept and 2000 saw the release of v1 of Epiowave (then named EpiorADD) with the first client taking the solution on board.

2001 was spent both developing v2 of Epiowave and also producing stand-alone applications for the market, designed with Epiowave.

► SUMMARY

That Epiowave is playing in a difficult market is impossible to deny. That it deserves to succeed is, in Butler Group's opinion, open to no doubt. Epiowave is an excellent technical interpretation and implementation of the best concepts of CBD and development for Web services.

Any organisation that is prepared to understand the complexities that development for Web services will bring, but also prepared to realise the benefits will require applications developed with a true *n*-tier architecture. Given the level of abstraction that Epiowave applications possess, the inclusion of this solution on any shortlist should be a formality.

► CONTACT DETAILS

Epionet

11 Marino Mart
Fairview
Dublin 3
Ireland

Tel: +353 (1)833 9558

Fax: +353 (1)853 3241

www.epionet.com

Important Notice:

This report contains data and information up-to-date and correct to the best of our knowledge at the time of preparation. The data and information comes from a variety of sources outside our direct control, therefore Butler Direct Limited cannot give any guarantees relating to the content of this report. Ultimate responsibility for all interpretations of, and use of, data, information and commentary in this report remains with you. Butler Direct Limited will not be liable for any interpretations or decisions made by you.

**For more information on Butler Group's
Research and Advisory Services contact:**

Europa House, 184 Ferensway, Hull, East Yorkshire, HU1 3UT, UK
Tel: +44 (0)1482 586149 Fax: +44 (0)1482 323577 www.butlergroup.com

Published on behalf of Butler Direct Limited by Addax Media Limited www.addaxmedia.com