

Technology Infrastructure

Butler Group Subscription Services

Web Services TECHNOLOGY AUDIT

hyfinity

Morphyc and PxP

Abstract *PxP is a platform for developing and deploying Web service solutions. It utilises XML in an architecture – known as Morphyc – that is designed to simplify the development of complex business solutions. The Morphyc architecture contains two core components: XPlatform and XStudio. The latter component is used to construct applications that will execute on the former. The solution brings together Web services, Component-Based Development, and Business Process Management in a total solution that answers many of the problems concerning the development and deployment of applications that work in an inter-organisational environment. There are two possible issues that might hold back the future development and use of this product. The first is the need for organisations to move away from 3GL procedural application development, the second is the supporting literature from hyfinity does not make clear the delineation between the offerings.*

KEY FINDINGS

- | | |
|--|---|
| <ul style="list-style-type: none"> ✓ Strong development and deployment methodology. ✓ Incorporation of business process as a key part of the application. ✓ Completely XML-based. | <ul style="list-style-type: none"> <i>i</i> Ships with open source supporting technology. X Confusing terminology used within supporting literature. |
|--|---|

Key: ✓ Product Strength **X** Product Weakness *i* Point of Information

LOOK AHEAD

Further enhancements of graphical tools for development involvement with business users. Implementation of vertical solutions that can be sold as a re-usable solution. hyfinity already has another complimentary technology called MVC, based on the same Morphyc Architecture. MVC can be used for developing XML Web Forms applications, based on XML Schema and XForms.

► FUNCTIONALITY

Product Analysis PxP is a platform for developing and deploying Web service solutions. It utilises XML in an architecture – known as Morphyc – that is designed to simplify the development of complex business solutions. The Morphyc architecture contains two core components: XPlatform and XStudio. The latter component is used to construct applications that will execute on the former. XStudio is a collection of tools, including the eXtensible Design Environment (XDE), which is used for the visual construction of distributed applications.

The use of the word ‘construct’ for developing applications with XDE is well chosen, as it follows the paradigm of component assembly, where functionally discrete pieces are put together to form a functional whole.

Product Functionality XDE uses a series of elements that match the assembly paradigm. These elements are:

- **XML Assets** – simply a collection of XML files representing industry standards such as ebXML or XBRL.
- **Patterns** – which is a representation of the best practice for assembly.
- **Runtime Patterns** – these are the physical implementation of the Patterns, and will utilise the actual XML Assets.
- **Blueprints** – these take a collection of Runtime Patterns and create associations between them to form the end application.

The XDE contains a GUI, which allows the developer to take the separate elements and build applications. Thus the XDE takes the XML Assets and the Runtime Patterns (themselves derived from Patterns) and associating these two elements in a manner defined in a Blueprint. PxP is an implemented example of a Blueprint available from hyfinity. It defines the associations for a peer-to-peer Web service for connecting business processes both intra- and inter-organisationally. Although the concept might be a little difficult to understand, it is a perfect methodology for answering the requirements of highly dynamic application creation in a service-based environment.

There are further elements that need to come under consideration when considering the processing. These are Nodes and Logicsheets. The Nodes communicate by passing XML messages via HTTP or Java protocols. This means that Nodes do not have to exist within the same physical location and can be distributed across an entire network (such as with trading partners).

Logicsheets are used to make decisions based on the content of the XML message. As both the messages and the Logicsheets are in XML format a declarative system is created, obviating the need to recompile code when changes are made.

XPlatform is the runtime platform for applications assembled with XDE. As XPlatform is constructed from XML and Java it can execute on any platform that supports the Java Virtual Machine (JVM). Architecturally, XPlatform contains three components and is representative of the runtime version of a Node. These three components are:

- **XEngine** – is a forward-chaining rules engine used to process XML messages. XEngine utilises the Logicsheets to enable this forward chaining.
- **XAgent** – acts as an incubator for XEngine. Agents collaborate by exchange of XML messages. They incorporate dynamic runtime discovery, which allows them to be deployed anywhere, either within a JVM as native Java services or as a Web service via XGate.
- **XGate** – is a generic gateway providing access using different protocols. As well as providing the discovery mechanism for XAgents deployed as Web services, it provides the same discovery mechanism for other Web services.

Although some of the terminology used may be unfamiliar, the benefits of using the Morphyc architecture and development methodology are easier to understand. The reliance on XML as implemented within hyfinity's solution provides numerous advantages over other, more traditional approaches to development and deployment of applications.

Applications are no longer tied to a rigid process infrastructure. The applications developed and deployed can dynamically adapt to changes in the process requirement. Not only is this of benefit from an internal standpoint, but it also means that inter-organisational application deployment is considerably reduced in complexity.

The cost of maintenance of deployed solutions is also reduced. The architecture is such that the costs of maintaining the middleware on which other types of applications are deployed is removed from the equation.

There is little or no requirement (depending upon current installations) for implementing new infrastructure. As an asset-based system, organisations can rapidly create new solutions for changing market forces without the need to 'start from scratch'. Existing assets can be simply re-used with new rules and associations.

Given the reduction in development time, maintenance cost, and the lack of requirement for integration, the ROI model is most encouraging, with a standard ROI of between six and twelve months. Naturally, much of this will depend upon the type and size of solution developed and deployed.

The application assembly model used matches the document flow within and beyond the organisational boundaries, making development and deployment much more intuitive.

When one considers Component-Based Development (CBD), and/or the Web service model, much thought has to be given to the issue of granularity. Many of the solutions on the market, which deploy processes as a Web service (for example), are coarse grained. In many cases this appears to be effective, but really avoids the true benefits of working within a service-oriented model.

The coarser the granularity, the less the opportunity for re-use of the assets. The downside is that finer grained components are more difficult to manage (due to their number) and are less easy to assemble into meaningful applications (due to issues of complexity). This is overcome with Morphyc by the use of Patterns and Pattern aggregation.

A prime requirement for any solution that utilises XML is the ability to handle any well-formed XML document regardless of the 'standard' used. Morphyc has this facility and will handle any XML standard (with well-formed documents) such as ebXML, BPML, and BPEL4WS.

XDE has a form-based tool available, which provides the facility for the graphical design of XML forms. The implementation of this tool reduces the development time, increases development consistency, and provides for XML standardisation. Forms-based applications can be generated in various formats complete with roundtrip capabilities.

Platform independence is achieved through the separation of specification and runtime. Additionally, this also helps reduce development and maintenance costs. In true three-tier CBD form, representation of data, logic, and presentation are kept separate and only brought together at runtime.

hyfinity would point out that the use of native XML is the reason why these benefits are all available. Butler Group believes that this is an understatement. The use of native XML is only the enabler. It is the hyfinity development team's understanding of CBD and, more specifically, CBD in relation to process components. There are many other CBD environments that would point to their XML capabilities, but where many of these fail (to a greater or lesser extent) is in the bringing together of CBD (especially at the finer-grained level), the document paradigm (in the ability to utilise any well-formed document), and the process element (which places process and document inclusion as the driver for the deployed application).

Whilst we would consider this a strength of the product in one respect, and this is the most important respect, it does come with a downside. By its very nature, the hyfinity solution enforces good system design and architectural principles. In a perfect world, this would not be a problem, but we feel that there is a trade-off to be made, especially in these early days, for re-educating developers as to the methodologies employed.

This may appear to be splitting hairs – if something is good then it should be recognised as good. However, the reality of development has to come under consideration, and especially the inclusion of the process management element will not sit comfortably with many development teams.

There is an upside, apart from the benefits already established. The graphical environment takes much of the initial development away from the traditional development area and allows non-technical people to design the applications that the organisation requires.

The architecture of Morphyc is extremely light. The only requirements are for a JVM, and XML Parser, and an XSLT (eXtensible Stylesheet Language Transformations) engine. The XPlatform component of Morphyc ships with an open source environment, which includes Tomcat (there is a requirement for a Web server if nodes are to be published as a Web service), Xerces (an XML parser), Xalan (an XSLT engine), and Sun Microsystems' JVM.

In commercial environments, IBM's WebSphere is the prime recommendation from hyfinity, and the technology is tested and proven against this platform; there has been no testing against BEA's WebLogic, but there is no apparent technological reasons why this implementation mix should not work. The solution has also been tested under Windows, Linux, and AIX.

The native Windows deployment is slightly problematic as the recommendation is for the use of a JVM, which would mirror the J2EE implementation. However, Microsoft's sporadic support for the JVM is well documented, and a future dedicated .NET version is in the pipeline, but with no release date yet available.

Performance enhancement is also something that needs to come under consideration in an environment as heavily dependent on XML as this is. There has been numerous tests against hardware-based XML accelerators, which plug directly into the Morphyc architecture, and although there has been no problems with any of the accelerators tested, hyfinity has a partner in this area with DataPower.

There is no requirement for a database for either development or deployment, but for transactional systems the use of an RDBMS or native-XML database is required and the product has been tested against IBM's DB2, Microsoft's SQL Server, and Software AG's Tamino.

The future developments for the product set are extensive and include:

- The implementation of WorkSpaces, which will enable different XML spaces to be defined and used during rule execution. This will provide separation between incoming, outgoing, and work-in-progress documents during the execution of each XEngine.
- Integration Adapters to allow for graphical communication with third-party products such as WebSphere, MQ, SonicMQ, etc., without having to write custom engines in Java.
- A higher-level graphical tool to remove the need for XML-based programming.
- Graphical debugging tools that will follow a message's progress through a sequence of nodes. It is also the intention to give the debugging tool the ability to understand rule correctness and evaluate the outcome of small subsets of functionality.

Product Emphasis The Morphyc architecture is such that it forms a persuasive argument for consideration as a development and deployment environment. As it is completely XML-based, it does away with many of the issues surrounding integration of applications, especially when one looks inter-organisationally.

To a large extent it is an implemented solution for the early promise of XML technologies, when XML was going to be the 'silver bullet' integration solution. Butler Group always considered that as a pipe dream, but hyfinity has gone a long way to removing the pain of integration with a strong blend of XML, CBD, and process management.

► DEPLOYMENT

Implementation of the product will be carried out by business analysts, architects, and developers with good XML knowledge, and a strong background and understanding of distributed systems. Administration post-deployment requires no specialist skills beyond those available with any competent SysAdmin, and even then the administration task is eased by the inclusion of graphical tools.

These tools come in the form of a Dashboard, which is Web-based. The tools provided include the functionality for tracking, auditing, and testing deployed nodes.

The resources for deployment will typically come from SIs, consultants, and in-house IT and business personnel. Currently, hyfinity can provide resource for consulting, education, and knowledge transfer.

Time for implementation is impossible to quantify as it completely depends upon the type, size, and scope of the project. However, given the required skill set, a small project could be up-and-running in a matter of weeks.

The education element consists of a five-day training course currently delivered workshop style at the client's base. The generic course contents can be tailored to meet any specific organisational needs. hyfinity also provides XML training through partner organisations, which forms an introduction to using the architecture, technology, and approach to development and deployment.

Ongoing support is an area which requires some more work on behalf of hyfinity. The model is for first-tier support to be provided by partners, but until these relationships are put in place, support is provided through a dedicated Web site, e-mail, and dedicated support staff.

The runtime platform requires a JVM and XML processing technology. The product ships with open source implementations of everything that will be required, but it also runs harmoniously against enterprise applications such as WebSphere from IBM and BEA's WebLogic.

The XStudio component requires Microsoft IE6.0 or above as well as MSXML4.0. A further consideration is that XStudio requires software for Scalable Vector Graphics (SVG), XStudio ships with Adobe's SVG Viewer.

The product runs on any platform capable of supporting a JVM (1.3 or above). hyfinity has certified its products against:

- Microsoft Windows 2000.
- Microsoft Windows XP.
- Linux.
- IBM AIX.

The product can also operate against the following databases:

- IBM DB2.
- Oracle.
- Microsoft SQL Server.
- Software AG Tamino

Access to the databases is through JDBC or a suitable bridge for RDBMSs. Database object libraries can be used through the implementation of custom engines. Native-XML databases utilise HTTP, SOAP, and Java APIs.

There is no prerequisite for integration in order to provide a solution. Existing business rules can be re-used with the product. Interoperability between systems with either a Java or XML interface is an inherent aspect of the product.

The viability of the solution is dependent upon organisations moving from a 3GL procedural development approach, to one based on a distributed XML-document model. Eventually this, and the requirements to move towards a full Web services or Service-Oriented Architecture (SOA) model, will require the infrastructure to support these paradigms. This is not due to any limitations in the hyfinity solution, simply a reflection that not all currently implemented infrastructures are viable for these new models.

► PRODUCT STRATEGY

The target market for hyfinity is in the medium to large enterprise space. Given that the solution fits well with developing and deploying applications for peer-to-peer trading partnerships in a highly dynamic manner, this is seen as a prime opportunity. Another aspect of this is the opportunity to sell into established trading relationships of smaller organisations. Again, the nature of the solution is ideal for this.

The short-term geographical aim is to establish the product within the UK market. Partnership discussions for extending into mainland Europe are currently underway, and it should not be long before the channels for hyfinity are in place.

A more medium- to long-term aim is to move into the key US market. Currently a partnership exists for related XML technology with the hardware XML-accelerator provider DataPower. Although the company is in the early stages of product placement, there are already a number of pilot projects being undertaken using the solution, and once these come to fruition there is no reason why a wider uptake is not forthcoming.

The key driver for hyfinity, and one of the key pieces of functionality of the solution, is the requirement to use XML as an integration methodology. Although XML *per se* is not the complete answer to integration, when it is combined with inter-process operability, as with this solution, it goes a long way to answering the requirements that were typically viewed as an integration 'problem'.

There are couple of key technology partnerships for the product. DataPower with the XML hardware accelerator and XML Security Gateway, is one, and Altova provides XML Spy as a tool within XStudio for general-purpose XML asset creation such as XML messages, XSL Stylesheets, and XSDs.

Licensing is based on the following model: all products are based on the Morphyc architecture with its two key components, XStudio and XPlatform. XStudio is priced per developer seat, and XPlatform is priced per CPU. A 'typical' implementation will be in the £50K-£150K price range.

Annual maintenance and support is charged at 20% of the list price per annum. This provides for telephone, e-mail, and Web-based support services and product upgrades.

► COMPANY PROFILE

hyfinity is a private company headquartered in Solihull in the UK. Formed in 2001, the company has now generated a product set with which it is ready to go to market. Its long-term business model is to have its software embedded and resold through a partner network.

Currently, hyfinity employs just eight people who have all been concerned with the development of the product. All company administration is handled externally. Now the product is live and market tested, the plan is to grow the company, based on the uptake of the solution.

Even though the real marketing of the product did not commence until Q4 of this year, the company realised revenues in excess of £250,000 in 2002, with forecast figures for £300,000 in the current year. The turnover in 2002 generated a Net Profit in excess of £100,000 in 2002. Considering this came from projects with early customers such as:

- Benfield Premium Finance.
- Darlington Building Society.
- HPI.
- Nissan.
- Software AG.
- Wildnet.

the company has demonstrated an ability to show that future revenues and profitability could be substantial. As with any company of this size, the main early thrust has to get proof points in place, and this appears to have been successfully achieved.

► SUMMARY

The hyfinity solution based on the Morphyc architecture is one that answers many of the problems facing businesses today. The promise of XML has to be realised and the hyfinity solution does just that. Development and deployment of applications that bring together the elements of Web services, CBD, and BPM is no easy matter, and hyfinity appears to have a solution that will be of real benefit to organisations who require dynamic, distributed applications.

The implementation of such a solution (productised as a Blueprint) with PxP is just one indication of the depth of functionality that this product has. Perhaps the only *caveat* that Butler Group would add is that some of the conventions and naming within the supporting material is confusing and fails to give the clear message that such a strong solution warrants.

► CONTACT DETAILS

hyfinity Limited

Blythe Valley Innovation Centre
Central Boulevard
Blythe Valley Park
Solihull
West Midlands
B90 8AJ
UK

Tel: +44 (0)121 5069111

Fax: +44 (0)121 5069112

E-mail: info@hyfinity.com

www.hyfinity.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
Subscription 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