

Open Market Components



A CBDi Forum Report

An assessment of the developing market for component based application software, with particular reference to the impacts of the Internet and predictions for development and segmentation in the component marketplace. We introduce the concept of the open market component, which we predict will emerge as the dominant commodity software concept.

David Sprott, cbdiforum.com January 2000

The Forum for Component Based Development and Integration

January 2000

Introduction

Post Y2K there will be widespread rejection of conventional IT solutions which have been characterized by poor quality, inflexibility and failure. There will be huge demand for cost effective software products that can deliver the adaptable platform for mission critical e-business applications. As the new COM+ and Enterprise JavaBeans server platforms that support component based architectures

become available, the CBDi Forum predict the component software market will grow rapidly, establishing a rich supply of high quality, low cost software. This will transform the dynamics and economics of enterprise software solutions.

The key to this transformation will be the widespread availability of productized business software components which are designed to be highly adaptable, capable of supporting widely varying business situations and deployment across diverse technology and application architectures. This report introduces the concept of the open market component, and predicts how and why the open component market will grow rapidly to become an important part of the overall software market.

Summary

The idea of software components is highly attractive, offering the possibility of cheaper, higher quality software products that form the basis for more adaptable business processes. However whilst the uptake of software components has been widely predicted, the mass market has taken longer to materialize than expected. Today we can observe the prerequisites for widespread adoption are now largely in place with the more mature component based technologies and platforms now available and implementing the concepts, together with a rapidly forming supply chain responding to the upsurge in customer demand post Y2K. There is widespread acceptance that e-business processes will need to be highly adaptable and that component based architectures are an appropriate response to avoid the lock-in effects of application packages.

In addition to driving the business priorities, the Internet will also provide a significant stimulus to the uptake of components. Most e-business application architectures will be component based. Perhaps even more important is the widespread and rapid activity surrounding standardization of "business meaning" to enable business to business processes. This standardization activity is creating an extraordinary level of consensus on basic business terms that will reduce the difficulty of integrating components from diverse sources. This in turn creates a significant opportunity for software developers to create mass-market components that can be more easily adopted than the equivalent application package and, being narrower focused, will be more easily replaced to provide upgraded or different functionality as the business processes change. We predict that businesses will increasingly prioritize adaptability and that the footprint of the typical software application acquired will become smaller and be focused on sets of tightly integrated services, which have been explicitly designed to integrate and collaborate with a wide range of applications and components.

The widespread dissatisfaction with the de facto application packages will fuel the reappraisal of how application functionality is architected and acquired in both business and IT terms. We recommend that businesses should look to what we are terming "open market components" which are sub-application units of functionality that have been productized such that they are easy to use, easy to integrate and economic. Early examples of open market components are already widely available for many common business functions covering e-commerce, e-business and many more prosaic processes and entities.

Some other industry analysts have been less than rigorous in their definition of components, leading them to conclude that the market for components is already thriving. We advise that it is critical to differentiate between coarse grain or monolithic, hard to integrate applications that have been loosely called components, from those components that have been specifically designed to be adaptable and easy to integrate and reuse.

The market for open market components is now commencing to grow very rapidly, stimulated by the post Y2K upsurge. The organizations creating components range from platform providers, ISV's (independent software vendors), to specialist component builders and end users. All of these types of supplier are and will be encouraged by the opportunity to productize their existing applications in whole or part, and to catch the market tornado as it forms. These providers will also be attracted by the huge potential size and diversity of the market. Whilst there will be clear leaders in, for example e-commerce shopping components, there are huge opportunities for suppliers to productize their expertise in one or more of literally countless niche industry sectors.

Stimulated by the above factors, a virtual marketplace for software components is already in formation, with several players creating trading exchanges. Today these intermediaries are playing an important role in creating the market, and their role will develop rapidly to create a seamless, virtual supply chain. For end users the opportunity exists today to create more adaptive application architectures using pre-existing components and is a recommended business productivity action.

Background

Respected industry analysts have been predicting for several years that the market for software components is either about to take off or has already arrived. Yet contrary to these optimistic reports the mass market has failed to materialize beyond simple widgets and controls. The entire software industry has been hugely diverted by the Y2K effort and there has not yet been a rich supply of high quality software components that make a compelling case for buyers. In addition there is a normal gestation process for new concepts that typically goes through several cycles of hype, interest and experimentation before the early majority embrace the new

ideas. In the case of software components therefore we should not be surprised to see this natural process in play, because there are important pre-requisites that need to be in place before the broader market will move to adopt.

Today we can observe the intersection of four critical factors that will predictably lead to widespread and mass-market acquisition of high functionality software components.

Technology:

The Windows 2000 and Enterprise JavaBeans platforms are the first mature platforms specifically designed to support enterprise scale component-based applications. Over the next two years most new applications will be delivered onto one of these platforms, which both make use of components easy, and from a technical perspective, mandatory. The component-based platforms will be complemented by widespread adoption of XML (eXtensible Markup Language) as an internal and external interoperability backplane, which will enable heterogeneous component connectivity. These two factors now enable the reality of distributed component environments without the complexities of earlier infrastructures.

Mature productized component concepts:

For a long time there was confusion over exactly what a component really is, as the concepts evolved from objects and other conventional software engineering approaches. Today we have evident consensus and maturity of the core component concept that has been established through the efforts of software industry leaders such as Microsoft, Sun and IBM, together with large numbers of tool providers and application developers. The software industry has invested massively to deliver server side development tools, component catalogs and management products, middleware, component based application servers as well as component based application products. Whilst we might be cynical that once again the software industry is creating a new wave of technology, which is now looking for a business problem to solve, the reality is that component approaches are by now well established and proven to deliver significant business benefits. Also, after years of component standards wars (COM, CORBA, Java etc), it is evident that multiple standards will not be an inhibitor to progress, rather a positive stimulus for competition. The industry now understands that platforms are commodities, market-leading applications are the differentiators. So the platform provider that enables the widest availability of component supply will be the winner. Remember OS/2?

Mature Supply Chain:

The most attractive aspect of software components is the opportunity to acquire software as parts that can be assembled and integrated into applications, potentially providing high quality solutions delivered faster and cheaper. However the realization of this promise has consistently been the most disappointing aspect of software components. Today we can see that independent software vendors (ISV's) are now moving rapidly to address this need, stimulated by the rapidly maturing technology and products referred to above. In addition component channel marketing organizations such as ComponentSource are well established and already facilitating the component supply as well as providing a reliable, branded, channel to market, facilitating the acquisition and support processes.

Customer Demand

The market for IT products and services, once predicated on the elegance of the technology, is now driven primarily by business requirements. IT procurement criteria are dominated today by an application product's support to critical business needs. However the mood of the procurement market is changing post Y2K. There is acute awareness that IT failure is common, and high probability. IT projects of all types are characterized by poor quality software, inflexible applications, inefficiency, high cost and an unpredictable IT application delivery process. The IT industry may justifiably be accused of constant reinvention of the wheel. Post Y2K, senior business management are now prioritizing the creation of effective business processes which deliver highly competitive and effective e-business relationships and e-commerce products. This undeniable trend will cause strong and sustained growth in the market for IT products and services driven by the transition to electronic business. Consequently there will be a ready market demand for components that address these key business needs and deliver high quality, adaptable applications with lower risk.

The Internet Changes Everything

Microsoft famously discovered to its cost that the Internet has the potential to exert dramatic change in very short timescales. That same effect will now impact the application market over the next 36 months, because with the pre-requisites discussed above almost in place, software components in conjunction with the Internet will create the opportunity to simplify application architectures.

The urgency for applications to support e-commerce and business to business processes will generally preclude the creation of

entirely new applications. In fact the typical Internet application architecture is based on dedicated application servers that integrate the new external relationships together with existing core business applications and databases. These application servers are almost without exception component-based architectures using either the Enterprise JavaBeans or COM standards. Already many ISV's are moving rapidly in an attempt to establish market-leading components that perform the functions common to e-commerce and business to business applications. This will be an area of huge growth over the next 24 months. As supplier and consumer organizations then evolve their core business applications, it will be natural to extend their usage of the component based platforms.

A second important trend is that XML has become universally accepted as the standard for document interchange. As a direct result XML has enabled widely supported activity which is making good progress in delivering standards governing common business meanings. The CBDi Forum predicts that these initiatives will very soon establish widespread consensus on the meaning of commonly used business concepts, driven by the demand to implement virtual business integration. This standardization activity is being driven at an extraordinary rate due to the high level of interest in business to business automation. However, don't be misled into thinking that XML is just an interoperability standard; the implications are far more profound. What will become clear very soon is that this widespread and rapid adoption of standards for business meaning will provide a massive stimulus and opportunity for software developers to create components that conform to the various standards. The more consensus there is on semantics, then the greater the opportunity there is to provide common functionality that can easily be integrated into existing application and product portfolios. Two years ago we commented that creation of semantic standards, whilst extremely important, would probably take years because of the difficulties of gaining consensus. Today we are happy to report that this forecast was incorrect and that the Internet driven surge in standardization of business meaning will create massive opportunity for common component functionality.

Today's applications are immensely complex precisely because they have to manage the integration of a broad set of data, relationships and functions. In tomorrow's world, components will offer services that can be used in a much simpler manner by many different users across entire organizations, as well as increasingly across organizational boundaries. The concept of a service is becoming widely understood as the key to creating adaptable applications. The service is simply the capability offered by the interface of a component. Ideally an application architecture should comprise sets of independent services providing common support to disparate processes. This makes it much easier to add or change functionality because the services are designed to be independent of each other to the maximum extent.

A good example of this are the e-business components that have recently become available from various sources providing commonly required functionality for e-business such as online storefront, order management and product catalog. Components of this type are designed to provide comprehensive support to very specific services in a relatively narrow footprint, and to be integrated with existing back-office and custom storefront applications. Many organizations are recognizing that the service approach enables progressive evolution of applications, allowing rapid response to urgent business priorities, such as creating e-commerce applications. The same strategy enables a mixed buy "and" build strategy.

For example consider the common requirement to establish a business-wide service to support a unified dot-com view of an organization to the outside world, to resolve the issues created by perhaps many different business units operating disparate applications which manage customers and accounts in different, uncoordinated ways. Components offering customer relationship services can often be rapidly implemented to complement existing applications, providing narrowly focused services that a) provide a façade to existing applications, and b) can be reused by existing systems to provide the same service, branching around pre-existing business logic. This approach is infinitely superior to a CRM package acquisition, enabling something closer to a custom solution without the build time or costs. The same approach can be adopted for a wide range of purposes precisely because the interfaces with the services are rigorously defined. This is an interesting example as there are excellent customer relationship management components available that offer the opportunity to acquire the core services, which are designed to be integrated with existing applications and provide much greater flexibility than a CRM package. Today the functionality range of such open components is increasingly broad, supporting capabilities such as trading platforms, shopping carts, credit authorization, address formatting, post-code lookup and many more.

The progressive move to a service based architecture will lead to tomorrow's applications being more sharply focused on narrower information and process domains implemented by components supporting independent services. Relationships between the components will increasingly be implemented in messages between the components.

The Open Market Component

The move to the e-business world will lead to a reappraisal of what an application is in both IT and business terms. Today we see the "application" as the unit of acquisition. In future the unit of acquisition will be much more flexible allowing organizations to assemble IT support to business processes from a variety of sources with confidence that the component parts will work together efficiently and with integrity. The existing concept of the application becomes largely redundant, locked in the age of monolithic packaged applications. The component based environment allows functionality to be introduced, upgraded and replaced on a continuous process improvement cycle, in a timescale that matches the business change cycle time. In this environment new components are acquired or bought, existing components reused and new components built only where the required functionality cannot be reused or

acquired. Clearly collections of components will be still be marketed as a complete solution, but increasingly there will be pressure on the vendors to offer much more flexibility and easier upgrade, and more choice by the use of open components.

The key to the component environment is the concept of the "integrity unit". These are individual parts that are the unit of change. Each integrity unit is complete and the dependencies on other components and the environment are precisely understood as a series of contracts, that ensure that the integrity unit may be introduced, upgraded and retired independently of other components. No longer will it be necessary to implement an entire monolithic application at one time, an activity that can only be carried out occasionally and always introduces huge risk of destabilization.

The integrity unit as described is however a technical concept, and we require further refinement in order to have a unit of functionality that can complement and integrate with existing applications. In addition to the characteristics of the integrity unit we need components that are designed to have the minimum of external dependency, to be as generally applicable as possible and to be productized such that they give the user confidence in terms of quality, longevity, maintainability and ease of use. We call the component that meets all of these criteria the "open market component". In Figure 1 below we illustrate how the technical concept of an integrity unit is complemented by qualitative characteristics that in combination create a productized component that will be usable in many different circumstances.

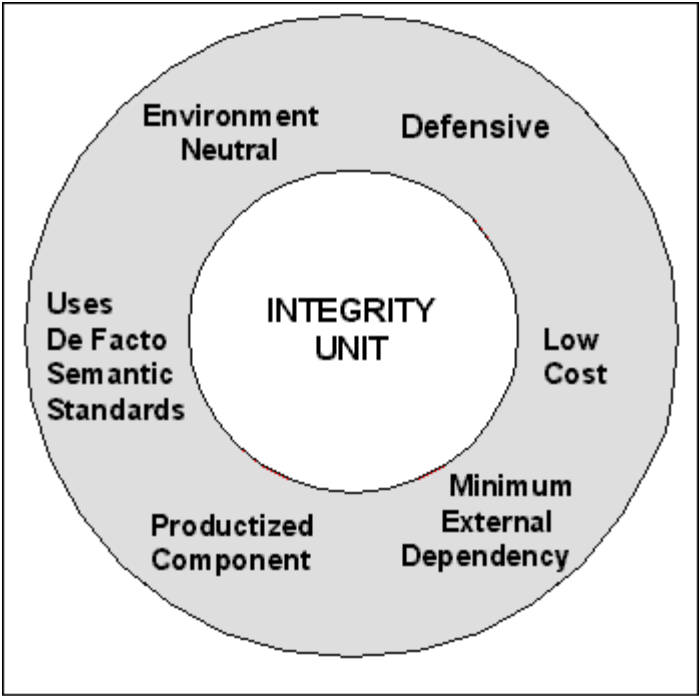


Figure 1 - The Open Market Component

The high quality open market component will be stand alone, and have full internal consistency and be designed to collaborate with a wide range of products and technologies. Whilst the advantages of components from a technical perspective are quite widely understood, the specific advantages of open market components need to be examined. From the analysis overleaf in Table 1, it will be seen that the technology layer is almost taken for granted, and that the real advantages of open market components accrue from advanced behavioral and usage characteristics. In some cases the open market component will be entirely independent, encapsulating all functions and data to provide the offered services. This is the simplest form and is already commonly available. In other cases, and we can expect this class of open market component to become more prevalent, the component will implement the de facto and de jure semantic standards referred to above, that will minimize integration effort. A further refinement which again will become increasingly common, will be the open market component that makes use of other services that are themselves de facto standards.

Open Market Components	Proprietary Components
Widely applicable Adaptable: capable of being deployed and used in many situations.	Operates within a specific framework Can only be used in conjunction with other parts of a framework

<p>Easy to integrate: uses open semantic standards</p> <p>Supports all current de facto technical connectivity standards</p> <p>Environment neutral</p> <p>Does not dictate process</p>	<p>Requires significant effort to integrate with other components or applications</p> <p>Does not conform to widely used semantic or technical standards</p> <p>Requires user to conform to specific process and data definitions</p>
<p>Stand Alone</p> <p>Can be used independent of other applications or components</p> <p>Defensive, non reliant, good immune systems</p> <p>Appropriate level of granularity that equates to service usage</p>	<p>Interdependent</p> <p>Can only be used in conjunction with other parts of a framework</p>
<p>Productized Component</p> <p>Published, branded, supported component</p> <p>Shrink wrapped, easy to use, upgrade and replace</p> <p>Widely available</p>	<p>Productized Application</p> <p>Larger grain integrity units require greater effort to use, upgrade and replace</p> <p>Often requires significant implementation support</p>
<p>Economic</p> <p>Low cost, designed to be bought and used by many</p> <p>Cheap to buy, cheap to install and use and low cost of ownership</p> <p>Critical mass market acceptance provides benefits of commodity quality and pricing</p>	<p>High Cost of Ownership</p> <p>Generally lower numbers of unit sales lead to niche market and higher costs</p> <p>Higher integration complexity increases integration and cost of ownership</p>

Table 1 - Open vs Proprietary Components

The converse of an open component is a proprietary component. Many of the business managers that bought ERP and similar packages in the last years of the '90's are now perhaps painfully aware of the constraints that today's application packages exert on business adaptability. With a typical package the integrity units are too large to allow rapid replacement and are generally highly interdependent. The software application products that provide integrity units based on tightly scoped services that have minimum interdependence between integrity units will be the most attractive to purchasers. It is highly likely therefore that as the benefits of componentized applications becomes obvious, that customer pressure will drive the application package vendors to reduce the size of their integrity units and to open up their products to integration with market leading application components.

If this list of quality characteristics for open market components seems hard to achieve, it must be said that the cost of creating a componentized product is less to do with software creation and more to do with productization and getting to market. As long ago as 1975, Fred Brooks identified that the cost of creating a true programming systems product is fully nine times the cost of the basic programming effort. He identified the programming systems product as being fully documented, tested, generalized and integrated into a wider application system. For some time now we have been aware that the cost of productizing the component based application product is significant, in particular because of the overhead of documenting the interfaces to the extent required and creating the appropriate levels of generality and independence.

At this stage we make a rough assessment that this cost for a significant productized component to be at least four times the cost of

creating the base programming systems product as identified by Brooks. Of course this represents a considerable investment, but for the successful creators of high quality components, there is the prize of creating the killer applications of tomorrow. It should also be noted that the entry cost of creating such a product is dramatically lower than that required to enter the conventional application market. In addition we fully expect and predict that there are large numbers of software developers who already have excellent products that will recognize the opportunity to componentize their products in whole or part, and bring them to market in a manner that enables them to deliver the product to a much larger market.

It is also important to recognize the wide functional applicability of this concept. Open market components are not restricted to particular sets of functionality or domains. Table 2 illustrates with examples, that open market components may be fine-grained complements to a user interface, a common utility, a generic business component or a domain specific component. All of these various types of open market component will exhibit the common characteristics that uniquely enable them to be used in widely variable circumstances that equally allows them to support highly volatile business environments.

The mistake that some commentators often make however is to assume that any software application that uses component concepts should be regarded as part of a component market. We recommend that software that is constructed as components but shipped as applications should not be regarded as part of the component market. This definition immediately excludes almost all of the ERP and CRM application package vendors who in reality are selling monolithic applications, of which only some selected functionality can be accessed using service based interfaces, but which do not offer the benefits of ease of upgrade or replacement. The same definition also excludes those component-based frameworks that form a complete application or suite of applications, which are highly dependent on the use of a common infrastructure and interdependent on collaborative services. The component based application level framework may offer good flexibility, but will probably not offer ease of integration of components that do not conform to the framework.

These distinctions are important for sellers and buyers of software. It is all too easy for vendors to claim that their products inherit all the benefits of components, whilst in reality they are fall a long way short.






Open Market?	Class of Component	Example Domains	Example services
	Application or Framework	Logistics ERP	Take Order, Ship Product Update Customer
	Domain Specific Business Service Component	Future Trade Telecom Pricing	Make Trade Price Interco Transaction
	Common Business Domain Service Component	Party Customer Post Code	Change relationship Update Customer Details
	Common Utility Service Component	Check handling UI Control	Print Check Spin Button
	Fragment or Sub-Assembly	All	Get Customer Data

Table 2 - Types of Components

The Software Component Market

Over the next five years, the entire software industry will move progressively to utilize component architectures. This is inevitable, as the de facto server platforms will make use of components mandatory. Whilst component based software should inherently be better quality as discussed in this report, only software products that are explicitly designed to be marketed as components, with standalone characteristics being a major part of their offering as opposed to complete applications, should be considered as forming the component market. As usual, as any new market takes off there is considerable hype, and it is not uncommon for software industry marketeers to latch onto the terminology long before their product research and development department deliver anything like adequate capability. Be warned. For this reason we encourage the use of the term "open market components" specifically to make positive differentiation between casual usage of the term component and those that are investing considerable effort to create whole component products.

The open market software component market will be supplied by a variety of organizations including Independent Software Vendors (ISV's), Systems Integrators, as well as some package vendors. In addition we should expect to see an upsurge of independent or small consortia of home workers, as highly experienced IT staff are laid off following the Y2K peak in demand for skilled personnel. Each of these types of suppliers will be encouraged to create components to supply a generally booming market in which there is huge opportunity for, initially killer application components, but thereafter an infinite number of niches in every vertical sector imaginable. Some suppliers will bring deep functional expertise which they encapsulate in open market components that perhaps bring capability in areas such as address handling, currency conversion, party management or trading risk analysis. Others will bring highly efficient and easy to use open market components for executing simple technical tasks.

Whilst the open market component suppliers may find customers through their own efforts, capitalizing on existing relationships or making creative usage of the Internet, most will increasingly use the services of component marketing organizations. The role of the component distributor emerged in the early '90's as the UI control market grew to support the sale and distribution of initially OLE and then subsequently ActiveX controls. Today the business model for the comprehensive marketing and facilitation of open market components has greatly matured over the typical ActiveX distribution operation, comprising sophisticated channel creation activity through promotional marketing to comprehensive after sales support. What we must however consider is that the market for open market components is potentially massive. There are countless business operations, services and functions in a wide variety of industry and commercial sectors that each represents an opportunity to create successful software products. Very soon there will be hundreds of thousands of open market components available and the problem will then be finding the ones that fit your need. We can expect that the component marketing channel organizations of today, will respond by increasing the sophistication of the search engines in line with the growth in the components available. This will allow potential customers to easily identify the components that match their needs, using appropriate search arguments that scan meta data which describes the component functionality and prerequisites.

As the market expands, also expect that new business models will emerge. Already we have various channel business models being deployed including broking, match making, distribution, and we should expect more models to emerge including forms of auction, stimulated by increasing demand. Expect also that the search engines and marketing tools will increasingly be integrated with the component assembly, integration and management tools on the desktop creating a seamless supply chain from supplier to consumer in both initial supply and ongoing support. This will increasingly look like a virtual software supply business.

The platform providers understand that the winner in the server market share war will be the vendor that has created the most successful application base. The platform providers, particularly IBM, Microsoft and Sun are therefore exerting considerable influence in creating the component market, providing investment in order to stimulate application developer and channel creation activity. The market for what we have termed open market components is currently just a small fraction of the total market for application software. However we are of the opinion that there will be significant growth in this market, because of all the factors outlined in this report, together with the Internet impacts on the broader application market. These factors lead us to conclude that the market for open components will probably grow significantly faster than many current predictions once the prerequisites are in place.

Conclusion

The transition to open market components is part of a major shift in the way that software is created and delivered that commenced in the early part of the last decade. The shift is enabled by the new component technologies, but is actually more about a revolution in acquisition practices and process. Instead of the individual, craftsman like approach to creating software systems, with high variability of result, the component software environment is a mature, repeatable product based delivery process that comprises of a sophisticated supply chain and channel to market. This transition will accentuate and accelerate the current shift away from in house software development, but will place significant market pressure on existing suppliers to deliver higher quality products very economically.

The Internet will be a huge driver of change, stimulating market demand and supply and driving the creation of standards. We can confidently predict that the huge demand will not be easily satisfied using existing software delivery approaches, and together with demands for rapid delivery, will stimulate massive reuse through productization.

For all classes of software supplier, systems integrators, ISV's, and package builders, there is opportunity to create componentized products that become de facto standards in specific industry sectors. What is particularly interesting is the huge number of individual opportunities that will exist in all of the various functional and industry sector areas. The opportunity to take existing expertise and capability and deliver that to a world-wide market exists right now, as the market leading brands will be established over the next 24 months. For component marketing organizations the opportunity is to facilitate the supply chain of high quality, business component builders and to establish tomorrows marketplace that links the supply and demand. For consumers, which of course includes systems integrators, ISV's and package providers as well as end user organizations, the opportunity exists today to acquire high quality components which can deliver important new business functionality faster and cheaper.

**David Sprott, Principal Analyst and Co-chair, The CBDi Forum
(The Forum for Component Based Development and Integration)**

Email: david.sprott@cbdiforum.com

References:

1. Examples of domain specific initiatives include the HR-XML Consortium, a non-profit group dedicated to the development and promotion of standardized human-resources-related XML vocabularies, which is creating an XML framework designed to enable web-based workforce management and recruiting services. There are also global initiatives such as the Electronic Business XML Initiative (ebXML), a joint effort of the United Nation/CEFACT and OASIS. ebXML is an open, vendor-neutral initiative to establish a global technical and semantic framework that will enable XML to be used in a consistent manner for the exchange of electronic business data. In addition there are vendor led initiatives such as Microsoft's BizTalk which is a community of standards users, with the goal of driving the rapid, consistent adoption of XML to enable electronic commerce and application integration.

2. The Mythical Man-Month, Frederick Brooks, Jr.
Addison Wesley 1975, ISBN 0-201-00650-2

News Item from CBDi Forum News (www.cbdiforum.com) 1st December 1999: Organizations from Around the World Gather to Launch ebXML Global Electronic Business Initiative

CBDi Forum Reports including Semantic Interoperability, Microsoft's Interoperability Strategy, Framework for the Financial Sector, and others in the CBDi Forum Journal INTERACT, May 1999.

The CBDi Forum is the leading information resource for advanced application delivery, focused on component-based development, application integration and legacy rejuvenation. The Forum assists IT and business managers to keep up to date with the realities of e-business application delivery based on practical end user, and industry expert experience in architectures, processes, projects, skills, technologies and techniques. Business and IT managers may join the Forum free of charge at:

www.cbdiforum.com

cbdiforum.com Ltd
PO BOX 536
Guildford
United Kingdom
Telephone: +44 1483 571151



© 2000 cbdiforum.com Ltd