On Demand

The strategy journal dedicated to one subject:

How to build the "Agile, Real-Time Enterprise", an organisation capable of fluidly adapting itself to change. **On Demand.**

Issue 1.00 – Introducing On Demand

Events

November, London

Building the Agile, On Demand Enterprise

- Conference
- Workshops
- Executive Briefings

Meet leading technical and business experts; network with peers and new industry contacts.

Produced by the Genesis Forum, and distributed free to subscribers of:

www.ondemand-strategy.com

Vendor partners

Contents

Neil McEvoy

Neil McEvoy is CEO of the Genesis forum, an industry initiative of Service Oriented Architecture vendors describing the business benefits of their technologies. He is the Chief Architect of the On Demand framework, the platform for autonomic business models that match demand and supply perfectly, and engender agile businesses.

He can be reached at www.ondemand-strategy.com

Neil's news and links

www.ibm.com/ondemand
A clear overview of Big Blue's positioning around On Demand.

On Demand has arrived!

The era of organic versus mechanistic business is upon us

You have to hand it to IBM, their sense of timing with regards to predicting and betting heavily on new growth markets is evident in their TV adverts. A few years ago, as the dot com boom was starting to swell we enjoyed their e-business adverts that practically re-invented Big Blue single handed, they did indeed position them as the e-business company.

Now we have On Demand. It has arrived.

It certainly has. On Demand is the simplest and clearest description of the single most important business strategy your company needs to start asking and answering questions about. The latest TV advert in particular sums up the imperative. The business executive is with his psychologist, pleading for help because his dreams are filled with crowds of customers relentlessly demanding things from him. The psychologist responds with "You're not agile enough; You can't adapt." Brilliant.

The power of the On Demand idea comes from its simplicity: Customers want exactly what they want when they want it, and they want it in exactly in the form and volume they want. You get the idea.

It's what "pure supply" should aspire to: Providing a perfect match to demand instantaneously when it is expressed, and if you can't engineer your business to be a set of 'lego pieces' that the customers themselves can assemble to meet **exactly their needs**, you're staying in the 20th century pal!

This first issue is primarily to introduce the core ideas of On Demand. It is a very simple motivation to design your business strategy, but there are quite a lot of new ideas and technologies to understand that are required to make it work.

Therefore each of these will be discussed in fine detail in our subsequent issues of On Demand, and an initial outline of what these will be is described at the end of this issue.

We are always keen to hear from potential contributors, so if you have any ideas and thoughts you'd like to share with us, don't hesitate to make contact!

Neil McEvoy

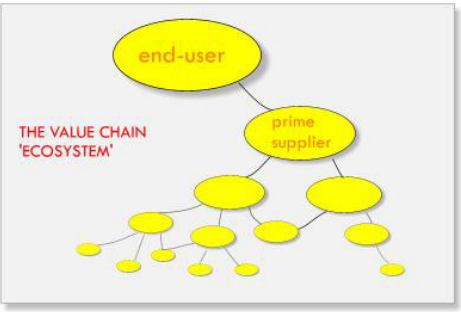
Anatomy of an On Demand organisation

The era of organic versus mechanistic business is upon us

So what does an On Demand organisation look like?

It's primary characteristic is one recognition of being network-centric, in contrast to 20th century organisations which favour isolation through top-down, command and control paradigms.

There is no such thing as an On Demand business, only an On Demand supply network. No business exists without buying parts from other companies, whether they are people or car tyres. How effectively **all** of these pieces fit together determines how fluid your organisation will be.



The more you move away thinking about your organisation, and more in terms of "your network", the easier it becomes to understand the need for, the benefits of, and the methods of achieving an On Demand "real time" enterprise.

On Demand capability is achieved through this network being fluidly interconnected in a "loosely coupled" manner, meaning that it can re-organise itself in real time to the preferences of its customers and other market factors.

I.e. **On Demand.**

Being part of a network

With this perspective, your overriding strategy should therefore be to design from the point of view of becoming a more fluid component of a larger network that is beyond your direct control. Up till now, most strategy and re-engineering exercises have constrained the focus of their efforts only to the organisation that is implementing them.

This presents a considerable shift in terms of how you think about defining your business processes and what technology will be required to implement them.

Neil's news and links

www.ibm.com/ondemand
A clear overview of Big Blue's positioning around On Demand.

Collectively focused (federated) workflow

Most organisations view their business processes as something they own and subsequently design and implement them with a combination of technology and data formats unique to them. The only point at which they consider a broader perspective is at the 'hand-over points' where they pass sub-components of workflow to their supply chain partners.

Hundreds of millions of dollars are spent on EAI (Enterprise Application Integration) projects to connect system to system to enable 'straight through processing' of workflow. This is a fundamental requirement for the fluidity necessary for immediate, On Demand fulfilment of customers needs.

However, achieving this has simply proven too expensive and difficult for corporates, mainly because the traditional point to point EAI model, where ERP is integrated to CRM which is integrated to the web site and so on, generates even more complexity so speedy and low cost process engineering becomes impossible.

Since the problem centres on semantics, where each different system describes the one same entity (e.g. the Customer) in a slightly different language, then the solution is surprisingly simple: Just use one format!

On Demand networks achieve low cost straight through processing of workflow through eliminating multiple different versions of customer data and instead work from a singular, shared information set. Instead of expending great amounts of time and money moving the same piece of information between numerous systems, changing it each and every time, the network members simply query and update the same source of data as and when needed.

This is known as a 'federated architecture' as the collective union of all parties involved maintains it.

Organic self-organisation

The most challenging dimension of On Demand organisations is that they require decision-making to originate from the bottom, not the top.

The 'command and control' model for organisational management is one that is practically universal, and that has gone without challenge since its inception in the mills and factories of the 18th and 19th centuries. This is primarily because it fits well with the most fundamental of human compulsions, that of the urge to exert influence over the environment in which we work to best suit our own personal needs. Naturally since power coalesces at the top, then this is the area of the business that exerts the most influence over the shape of the entire organisation.

However, customers rarely deal with the people at the top, so the need to change to meet customer needs must be mangled through various "reporting procedures" for months before action is taken to implement the change customers have requested. Even then, the nature of the change to be implemented must be mangled back down through the chain before it reaches the point it is needed. This painful process can take weeks, months or even years.

Try it out: Phone a large call centre and ask to speak to the head of customer services. I tried it and was told that this person doesn't actually speak to customers. Wow. A Head Of Customer Services who doesn't speak to Customers. Fascinating.

The solution is very simple: Empower the customer touch point, whether it be machine based or a human being, with the authority and ability to implement change requested from customers when it is needed and in the form that is requested.

Combining these two foundations creates a business process framework that avoids becoming entangled in both corporate IT and within management hierarchy oriented more towards personal egos than satisfying customer demand.

Instead, what the customer wants to happen just happens; quickly and effectively. On Demand.

Becoming “truly” customer-centric

The biggest challenge you will face with designing On Demand business models will be the wrestling match you will have with your own sense of control.

You see, claiming to be truly customer-centric when you’re actually not is really quite easy; being ‘purely’ customer-centric requires a leap of faith and a process of “letting go”. Why?

Because normally what happens is i) we say we’re customer-centric but then ii) we design how business processes should work to suit us. For example, we offer a useful white paper on our web site, but then make the potential customer fill in a lengthy form with umpteen **must complete* fields. Why? Because **we** want the information, not because it suits the customer.

This example highlights new ideas you should start considering that XML Web services and other technologies will have a huge impact on.

Why do you need a customer to key in their information and then store it in a database? Technically you don’t need it because it already exists, in like a million different places. You don’t need another copy, you just need access to another source so that you can use it when you need it.

This is one of the backbones of the Service Oriented Architecture, the idea of shared services infrastructure. Why have a million copies of a piece of information all over the place, when you can have just one perfectly up to date one that everyone shares in an equal manner. This benefits everyone:

- The customer: They don’t have to repeatedly key in data
- The business: They completely eliminate the need for “data cleaning” and other useless functions

More sales, less cost. On a very large scale.

So what does this mean you need to start thinking about?

1. Start looking for groups of business partners (E.g. your supply chain) where you can start working from a singular data-set for customer and process information

2. Stop “hard-coding” your business processes. Instead, “let go” and stop worrying about owning and controlling specific data but instead references to the data

Continue this process and you reach a conclusion where customers will directly be able to program their own specific workflow that is “attached” to your business: How you should answer the phone when they call, when and how you can contact them, what type of cheese they prefer.

This entire goldmine of customer information is likely to exist in one place: Where it can be controlled by the customer external to any one commercial organisation. When you give up the sense of control and instead just start worrying about getting permission to use it in a hassle-free way, then you truly become customer-centric, because the customers themselves have become the ‘pure’ designers of your business processes. You’re there just to cultivate useful assets that they can assemble into a solution wholly unique to their needs.

e-Marketplaces and organic supply networks

Building commerce platforms that enable bottom up, self-organising growth

Soon, autonomous XML Web service "agents" representing the purchasing business processes of customers will begin seeking out demand to meet their needs, dynamically negotiating deals via agreed languages and concluding transactions without human intervention. Understanding how this framework fits together will be essential for global, 21st century business.

Your company will then fulfil this demand through integrating the products and processes of your own organisation and those of your business partners. How fluidly this is achieved will be determined by the nature of the relationships you have with these suppliers. If they are fixed and inflexible, you will be too.

As companies become more like 'Lego pieces', the easier this fusion of processes can be achieved, enabling the smallest to the largest organisations to assemble themselves to present a solution to a larger problem they themselves cannot solve. This fusion is the catalyst for growth.

Federation based business models

Customers like less work to do when ordering stuff.

Giving them forms to fill in and new numbers to call is giving them reasons not to buy from you. 'One click' ordering has shown us the way and it will become the commercial model to aim for when designing customer-centric business processes. This is enabled by process data federation, which naturally blends and shares this information with partners.

Working from existing network data in this manner not only makes things easier for your customers, it also makes them easier for you too. You can launch new, superior business models quicker for less cost, as you are not repeatedly creating new processes, but instead calling on existing ones and plugging them together.

When you release control of customer information to customers themselves, not only do you save millions in "data cleaning" and other useless functions, but you also gain "Exact Demand". Giving customers control over data and processes specific to them encourages them to be far more proactive in stipulating their needs in a form that your Web service agents will be able to process.

This information will be stored via Semantic web concepts and standards, making it both machine-readable but also consistent with how the marketplace wants to process it.

Neil's news and links

www.chaordic.org
The Chaordic Commons, home of the inspirational thinking of Dee Hock, CEO Emeritus, VISA.

www.johnhagel.com
John has pioneered thinking with regards to loosely coupled supply networks.

Self-organising common frameworks

This is why two areas are very important to understand further are:

- Open standards for describing business processes in a common language that everyone can share and that are extensible (they can be improved if they need to be)
- A decision-making framework for enabling a consensus approach for making these changes

Open standards

Open standards provide the means to eliminate complexity that results in tens of millions of dollars being required for EAI and other intensive process enablement projects. The problem is that many systems describe the same logical thing (e.g. the customer) in many different formats. Siebel stores it in one format, SAP another, but they're both still referencing the same customer. EAI provides the translation to map one to the other. However, multiply this by hundreds of different systems in hundreds of different companies and you get a sense of the problem.

This has been further aggravated by continual point-to-point EAI strategies. I.e. Siebel to SAP, SAP to Oracle, Oracle to BAAN, BAAN to Microsoft and so forth.

However through all of this integration, there is still only ever one actual customer!

Open and shared standards provide the means to only have one technical description of one actual customer, and for everyone to use it, in a manner that personally and commercially benefits everyone involved. One you adapt your systems to use this one reference, that's it, no more EAI projects, you can start focusing more on improving the value of the processes themselves.

Change management

There is much derision of the design by committee process, however if you want a group of autonomous people to collaborate in this way you need to give them something very important to humans: A sense of equal control over decisions to change.

The most effective re-organisation of a network system occurs when all members are in equal union in terms of opinion, ambition, ability and political control.

This is why the Chaordic model designed by Dee Hock, CEO Emeritus VISA, is so important. Dee recognised that to deliver truly valuable customer experiences, a network of partners, no matter how big or small, had to interoperate in a fluid manner sharing customer and process information equally, and that to do so, each member had to know they weren't exposed to biased decision-making structures.

Implementing the loosely coupled federation model that VISA is based on, Dee created one of the most commercially successful organisations of the 20th century.

And provided the means to build commercial processes that can maintain shared e-marketplace standardised languages that all business partners can comfortably adapt their systems to use.

The Service Oriented Architecture

Enabling loosely coupled business systems

The SOA is the architectural model that makes all this stuff fit together, from business models to technology platforms.

It's principle function is to create an abstracted service layer where physical assets, from people to disk storage, are "represented" on the network in network terms, by service 'agents'. With the appropriate security in place, these services expose the properties and processes of the resource, enabling it to be 'consumed' by other service agents.

This interlinking happens at "computer speeds", and by using a commonly agreed semantic framework for how these conversations should be initiated and carried out, machine to machine processing of incoming demand can occur in real-time.

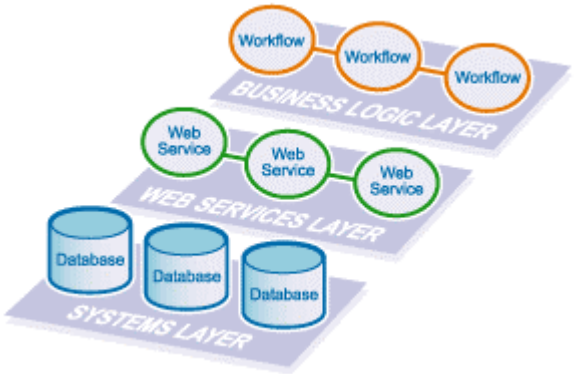
Neil's news and links

Sun Jini
www.sun.com/jini

Blue Titan
www.bluetitan.com

Intamission
www.intamission.com

Object Management Group
www.omg.org



Loosely coupled federation

Loosely coupled federation is principally about object-oriented abstraction, but with a particular remit of bottom-up self-organisation. If services are self-aware and self-determining, then they can dynamically unite to serve a higher purpose, forming a collective entity capable of representing an entire network and all of the methods of its constituent members. Since this is the key to unlocking new business growth, it's kinda important.

The framework for fluid process flow is achieved when customer-centric business processes are encoded with an information architecture that is shared across all the organisations that play a role within fulfilling customer demand.

Defining industry wide ontologies and implementing them consistently via Topic map frameworks presents a universal information "ecosystem" that more naturally reflects how processes work. This shared "information bus" engenders an abstracted workflow-centric layer that is inherently superior to traditional point to point EAI methods, and creates a virtualised service architecture which is loosely coupled from its enabling technology, offering greater resilience and making fluid adaptability possible.

Shared Services infrastructure

The scope of the benefits achieved by Loosely Coupled Federations coming about are directly proportional to the volume of usage of shared services for universal traits, such as Identity. Since it is this elimination of duplication that drives customer value, then the more the better.

Application of the Service Oriented Architecture within an organisation begins the process of disintegrating the islands of information that exists within the modern corporation due to the proliferation of multiple application systems that “lock in” business information.

Creating a simplified, service-centric architecture eliminates the plethora of complexity that restricts agile operations and customer-centric, dynamic change management. The end result is an organisation malleable to the changing forces of the ecosystem it exists within, and thus a truly adaptive enterprise.

The speed of change that can be implemented is directly related to the volume of new code that must be created. Going hand in hand with Model Driven Programming is the use of shared software services infrastructure.

Equipping programmers with the appropriate tools and available network components allows them to dramatically reduce the time required to instantiate new business systems. The bulk of the code can be “stamped out”, not only dramatically reducing the time to completion but also automatically integrating pre-approved software components so that corporate policy is inherently enforced throughout all new application development.

Having published new code to the application network, the policy changes must be dynamically reflected throughout the entire network infrastructure.

Federation of data throughout distributed networks is a foundation component to enabling the abstracted business process layer that makes fluid adaptability possible. Furthermore, it is the consistency of information achieved through federation that makes it possible to design truly customer-centric processes, such as multi-network single sign-on.

Model driving programming

If duplication is eliminated, then creating new business models can simply concentrate on the new information and processes that are required, not repeatedly re-inventing the wheel. Model driven programming allows developers to “stamp out” pre-defined patterns of code that automatically utilise these shared services. This reduces costs and time to market.

Model Driven Software addresses problems with development time, code-generation, uncertain requirements, flexibility, and change. Without interrupting projects.

Model Driven Software is an innovative and more productive way to develop software, addressing the key areas of software development risk, notably; flexibility to change, and development speed. This is achieved through the real-world open-standard implementation of the recently released OMG Model Driven Architecture™.

Since faster implementation of change is a goal of agile businesses, then the ability to create new business models and enabling code faster offers an obvious benefit.

Autonomic infrastructure

If faster and more effective change is the strategic objective (to become more agile) then autonomous infrastructure delivers a number of building blocks:

Allow infrastructure to adapt itself. Computers are good at making and implementing “dumb” decisions very, very quickly. People are good at extremely complex problems (like other people!) but are very slow. Since much of the demand you experience will be originated from other computers, then you need to have your systems ready to interpret it and make suitable commercial decisions. For example, buy more temporary bandwidth when your web site experiences a peak of traffic.

Eliminate downtime. Downtime is mainly caused by human error working on dumb decisions.

Free the people! Eliminating more dumb work that people need to do results in them having more time to do what they enjoy the most and contributes the most growth potential: Generating more value for customers.

Since these autonomic operations are enabled by machine-to-machine “conversations”, then a universal language is required. Hence the term “Semantic web”. It’s not a separate Web, it’s the language of the Web, that applications will need to “speak” to take part in a global Web services e-marketplace. This is why the critical success factor is not the sophistication of the language (the open standards) but more the fact that everyone in the network speaks it.

The implementation of a virtualised service architecture within the corporate offers immediate potential when focusing purely on the IT platform itself. Indeed an adaptive organisation will require an enabling, adaptive IT infrastructure. Building fluidly adaptable IT systems that can dynamically provision, unite and deliver resource to users and other applications presents the framework for:

- Autonomic adaption of IT to meet real-world business needs
- Considerably more effective utilisation of paid for IT assets

Applications of automated IT can range from the mundane such as provisioning users email accounts through to harnessing and blending unused resource for the purposes of large-scale intensive modelling and calculation applications.

Security and performance management

Naturally these services must be secured from technical and commercial abuse, so the platform must feature an inherent capability to secure, test, monitor and invoke the process engine to re-adapt itself when needed.

Furthermore, Web services must be able to expose and work with service level agreements so that business processes can feature intelligent decision-making capabilities to comfortably work within the varying performance and reliability environment of distributed network services.

Raw Communications

An On Demand business in action

Eiren O'Keefe

Eiren is the Director of Operations and Development for Raw Communications. Eiren has successfully built an On Demand business through the application of all the key concepts covered in this executive briefing, and is a leading authority on the critical issue of applying them for business benefit.

www.rawfinancial.com

Eiren's news and links

msdn.microsoft.com/architecture
Good for all things .NET architecture related. A wealth of information is provided by Microsoft to keep most anyone occupied for a good, long while.

www.webservices.org
WebServices.Org provides timely information on Java and Open source based options.

www.internet.com
For access to a multitude of topics from a single location Internet.com makes it quick and easy, though you can find yourself lost in the links for hours on end.

Neil McEvoy

Eiren: Something I get asked a lot is "Is this On Demand stuff real? Is anyone actually implementing "real" projects", usually with the inference that they think no one is because it's vapourware.

However, Raw is a great example of a business utilising powerful technology architectures to accelerate business growth, perhaps you could share an overview of your current strategy with us?

Eiren O'Keefe

Our strategy was naturally borne of our commercial requirements. Since the beginnings of RAW the technology that underpinned the business had been purchased and developed externally and we were rapidly approaching a point where our external service providers had different focal points that didn't always intersect with our own. This is not saying that they were poor or inattentive but simply their business focus had changed over time just as ours had. The original platform was based on bespoke technology, protocols, applications, etc. and not only was it difficult to implement but also to maintain and nigh on impossible to integrate effectively with any customer based systems – our platform sat as an island within environments rather than an integral piece thereof.

The original foundation was built on Windows Server 2003 Enterprise Edition using Visual Studio.NET and C# as the core language. All of the underlying data across the board that supports the platform was done as XML, which has facilitated rather rapid and easily managed changes to the data set, including modification, removal and addition of elements.

The code itself that comprises the platform takes advantage of WS2K3's underlying support for XML which further reduced our development overhead in that the general plumbing was well provided for (though as always there were some exceptions that required a bit more coding on our part) and the interconnections between the various components were designed from the ground up as loosely coupled web-services.

This platform (for those that are interested) facilitates broadcast of media (audio/video/data/etc.) via satellite/circuit to our "appliances" (the servers bespoke with our code and architecture) at tier one financial institutions globally facilitating live delivery of results announcements, morning calls, etc. to the desktop. The current evolution (which we're developing at this moment and will demonstrate at the SIA show in NY w/c 16 June) now encompasses our existing web presence. This too is being re-written from the ground up and we're utilising a web services architecture with loosely coupled reliable messaging (built on MSMQ for routing/reliability aspects over public networks including NAT traversal with grace) such that it will now become a federated structure.

This supports self-publication of content from institutions including digital rights management for entitlement purposes. The services that provide this too will be exposed utilising web services to further direct integration within customer environments. There are plenty of administrative interfaces that people must deal with as is and through our comprehensive web services approach rather than force customers to use yet another admin interface, they can easily integrate these aspects within their current administrative consoles, portals, etc.

As for the functionality and what's placed within this iteration I've approached it with an accounting value approach against the functions combined with triage style analysis to identify those functions that are of most actual value. While the bean counters see no value in the web services aspect and how this has been designed our customers do and it is this approach that allows us to rapidly evolve the platform; in some cases providing underlying functionality that we simply expose once the business has got its head around it -- we knew they'd want it but they didn't know it yet.

It has and is quite a challenge and there are still a number of hurdles for us to jump however the underlying approach of treating everything as a message has paid significant dividends thus far and will continue to do so. It is however a diligence that requires a solid up front plan which we've modelled extensively utilising UML but not gone over the top with it – we've done what was necessary only versus trying to model every, single aspect out across the board.

Again, the dividends paid here are quite exceptional as now our rich client application and web presentation are able to take advantage of reflections -- where a new application/interface/etc. is required we quite literally select the classes and they are presented in the new app/int ready to rock with validation and functionality present -- no coding required.

Whilst many knock Microsofts messages (and they do deserve it in some respects) we quite simply would not have been able to deliver such a rich platform so rapidly and effectively using any other approach/technology -- we looked into it and would have spent more time coding the basic plumbing than presenting the business solution, which is ultimately what this is all about.

The first iteration which completely replaced an entire MPEG-1 based broadcasting architecture, provided significant enhancements to functionality and improvements to existing functionality, was developed in about four months time and deployed last year 5 August 2002 as the first global implementation of a WS2K3/WM9 Series based architecture.

Distribution partners

On Demand will be made available for download from distribution partner sites.

www.looselycoupled.com

www.searchwebservices.com

Next issues

On Demand will explore a number of business, industry and technology areas in considerable depth throughout the following next issues. Due to the volume of information that must be communicated to clearly describe how to build the agile enterprise, topics will be categorised and distributed as follows:

1. Business strategy

The strategic imperative for the On Demand Real-Time Enterprise. What is it, how does it work, what are the benefits, who else is adopting it? A succinct discussion of the big picture.

1.1 Loosely coupled supply networks

1.11 Defining loosely coupled supply networks

Understanding the application of the SOA to business process architecture to engender fluidly adaptable workflow for orchestrating supply and demand.

1.12 Implementing shared marketplace ontologies

Creating federated unions with business partners as a catalyst framework to engineer new growth.

1.2 The Real-Time Enterprise

1.21 Building Executive Dashboards

How to build suitably simple but powerful Real-Time modelling, programming and reporting interfaces to business systems for non-technical people, that provide immediately up to date information and process control.

1.22 Building best-practice autonomic processes

Designing flexible business processes that can utilise Web services to automate repetitive procurement and management functions in a manner that protects corporate strategic policy.

1.3 Strategic drivers

1.31 CRM Masterclass

A challenging review of what loosely coupled, customer-controlled CRM architecture means for sales and marketing programs.

1.32 Six Sigma Services

How to integrate management frameworks such as Six Sigma with XML Web services to provide the IT aligned, customer-centric business design and reporting methods.

1.4 Industry strategies

1.41 e-Government On Demand

An in-depth review of the current progress of e-enabling government services, and what role XML Web services can play in accelerating its success.

1.42 Multi-network gaming

How can federated identity and process information contribute to greater uptake of cross-network media services, such as multi-player gaming between Digital TV, mobile and Web.

Distribution partners

www.looselycoupled.com

www.searchwebservices.com

2. Technology strategy

Expert contributors from the smallest to the largest specialists will provide us insightful articles across the following sections:

2.1 The Service Oriented Architecture: Core model

2.11 Loosely coupled federation

Designing service-centric environments that can sustain application functionality across multiple, distributed network systems.

2.12 Messaging Oriented Middleware

Building messaging based application communications via the publish/subscribe model of the SOA.

2.2 Information architecture

2.21 XML and Topic Maps

Understanding the fundamentals of defining and implementing meta-data for autonomic operations.

2.22 Implementing shared marketplace ontologies

Building b2b integration via open, standardised languages such as BPEL4WS.

2.3 Shared Web services infrastructure

2.31 Identity and policy federation

Successfully replicating policy data across distributed systems to achieve fluid business processes enabled by Identity Federation.

2.32 Shared software component infrastructure

Building composite applications that invoke network available shared Web services.

2.4 Application development strategy

2.41 Model Driven Software

How to practically adopt OMG based Model Driven Programming to “stamp” out common code patterns that utilise existing shared Web services components.

2.42 Platform decisions

A thorough discussion of which platform (.net, Java etc) is best suited for different situations.

2.5 Infrastructure and legacy systems

2.51 Auto-discovery and registration

How to automatically explore, discover, identify and register the business processes that exist within legacy systems such as SAP or Siebel.

2.52 Networks and hosting

Architecting physical infrastructure and management systems to ensure performance is sustained between communicating Web services agents.

2.53 Data-centre automation

Applying standards frameworks to the provisioning and supply of common IT infrastructure from internal and external data-centre infrastructure.

Distribution partners

www.looselycoupled.com

www.searchwebservices.com

3. ROI and implementation

ROI and implementation will seek to address the simplest but toughest questions: How can you make significant investments into these new business models and technologies that successfully generate a profitable return.

3.1 ROI strategies

3.11 Short-term areas

Finding areas in the business where the application of the SOA will yield short-term ROI.

3.12 Investment strategies

Creating federated unions with business partners as a catalyst framework to engineer new growth, and share risk, costs and implementation requirements.

3.2 Vendor review

3.21 Battle of the giants

An in-depth review of the strategies of the major vendors such as BEA, Sun, Microsoft, IBM, HP and Oracle, and their SOA capabilities compared against the framework described in IT strategy.

3.22 End-user case studies

Complimenting this will be profiles of some of their customers, outlining what their outsourcing and development approaches were, and how well they worked or didn't.

3.3 Implementation strategy

3.31 Skills breakdown

A structured description of the architecture and development skills required to implement loosely coupled business systems.

3.32 Project methodologies

What project management disciplines will contribute effectively to a successful implementation?

3.33 *Automated directory population*

How to automatically explore, discover, identify and register the business processes that exist within legacy systems such as SAP or Siebel, and how this can accelerate successful On Demand deployment.

3.4 Solution scenarios

3.41 The On Demand Call Centre

Tackling the toughest issues of running the multi-channel contact centre through the application of distributed business and technology models.

3.42 Knowledge and Content Management

Applying Semantic XML Web services in a modular fashion to eliminate the need for large, expensive content management systems for e-business and intranet KM projects.

Further information

If you would like to attend these events, have suggestions for presentations or would like to discuss sponsorship options, contact:

Charlie Williams
Ark Group
+44 (0) 208 785 5903
cwilliams@ark-group.com

or visit:

www.ondemand-strategy.com

Events

Conference

In partnership with the Ark Group (www.ark-group.com) Genesis is running a large conference describing How to Build the Agile, On Demand Enterprise, featuring keynote presentations from world leaders in the field. This is scheduled for (7-8*) November 2003, at the (Hilton Hotel), London.

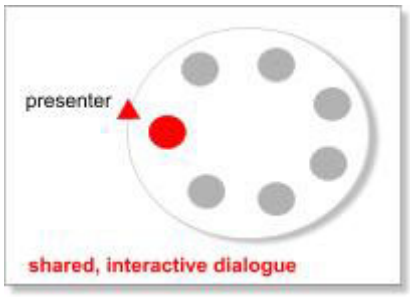
Workshops

The conference will be followed by a series of interactive workshops:

WK1.1	**Information Architecture Acceleration Primer**
	- Introduction to standards such as BPEL4WS
	- XML Topic Maps
WK1.2	**Service Oriented Architecture**
	- Building loosely coupled systems
	- Defining shared Web services infrastructure strategies
	- Implementing identity federation
WK1.3	**Rapid business application development**
	- Model driven programming
	- Supply network integration through XML

Executive Briefings

Genesis Executive Briefings are designed to combine learning and relationship building with potential business partners.



In contrast to the one to many 'broadcast' model of most seminars, each presentation is conducted with a small focused group, with the emphasis on short, snappy presentations followed by engaging dialogue discussing the real-world implementation specific to your business and those of your peers.

This additionally presents the opportunity to network with fellow executives and discuss the potential for partnership presented by the technologies and models discussed.

1. **Enterprise Services Architecture:** The most comprehensive and insightful agenda describing the Service Oriented Architecture and how it is enabled by number of key foundations.
2. **Business/ management strategy:** What new fluidly adaptable business models can be enabled by the Service Oriented Architecture, discussing issues such as Six Sigma and other management sciences.
3. **On Demand Call Centre:** Aimed at large telcos and enterprises operating call-centres, this session focuses on the On Demand model applied to the strategically important call-centre scenario.
4. **E-government On Demand:** A particularly exciting briefing that includes discussion of e-democracy, strategies for economic stimulation and corporate/government partnership opportunities.
5. **On Demand TV:** Focused mainly at xSP sector: Telco, mobile, Digital TV, and overlap with media companies. The schedule is about using distributed services infrastructure for media and games distribution etc.