

REQUIREMENTS AND TRENDS IN ADVANCED TECHNOLOGIES FOR THE LARGE SCALE ENGINEERING UPTAKE

Alain Zarli*, Olivier Richaud*, Eamonn Buckley**

*Centre Scientifique et Technique du Bâtiment

290, route des Lucioles, B.P. 209, 06904 Sophia Antipolis cedex, France.

**Building Information Centre

DIT, 41 Lwr Dominick St, Dublin 1, Ireland.

ABSTRACT

Key challenges nowadays facing industry are increased competition, increased complexity and wider market reach. Consequently, the information infrastructure has become vital to enterprise competitiveness, though the diversity of enterprise databases is still a barrier to industrial exploitation of the opportunities offered by this infrastructure. Modern enterprise information systems must interoperate in Inter/Intranets and with the WEB with full interactivity, reliability and security, and have to be flexible enough in order to quickly adapt to today's changing business environment. This paper investigates the requirements and specification of a standardised open infrastructure, dedicated to the WEB interconnection and integrating in a flexible way the enterprise business model through the emerging concept of Business Objects. This work is undertaken in the context of the WONDA¹ project, which aims at developing an open framework relying on standards and providing an enabler for business objects and electronic payment.

Keywords: distributed client/server architectures, communication middleware, business objects, WEB oriented open and standardised industrial business objects frameworks.

1. INTRODUCTION AND MACRO ISSUES

The new challenges enterprises are facing today can be derived from the following political, economic, social and technological drivers:

- *Political*: the global world-wide deregulation has lowered and in some cases eliminated barriers to entry in many industries and regions, and market incumbents can no longer rely on these barriers to entry in order to protect their market position.
- *Economic*: a modern financial infrastructure has enabled potential entrants to raise funds in order to compete in almost any sector, and again market incumbents are facing increased competition. Enabled by deregulation and free trade, firms can now achieve return on investment for technological and business innovations through selling to wider markets. Therefore industry is facing more competition, and more complexity through newer products (technological innovation) and newer ways of selling them (business innovation). Also the fact that multinational companies are benefiting from economies of scale forces other firms to expand their market horizons to achieve such economies or to focus and differentiate more leading to increased business complexity.
- *Social*: the global village, as brought by global telecommunication and media reach, and the empowerment of the citizen, as brought by consumer rights, education and access to

¹ EP 25.741 WONDA: *WORLD wide eNterprise Data interoperABILITY*. URL: <http://www.bild.ie/wonda>



information, has led to increased customer sophistication, leading in turn to more exact user requirements (product customisation and business complexity) and unblinkered users who are now willing to try new products from new firms selling them in new ways.

- *Technological*: information and communication technologies are merging, not just to provide new tools, but also new drivers and in many cases to change the structure (competitive forces) of many industries. Previously, technological innovation and transfer has been hampered by the need to achieve return on investment in a limited marketplace. Now because of free trade, the market place and potential for return on investment has been expanded. Therefore more technological innovation, competition and business complexity is being brought to the market. Finally, in the past, industry leaders have controlled innovation so as to control the structure of the industry in which they operate (thereby maintaining the status quo). This behaviour may be changing as leaders are less confident of controlling the industry structure and so are concentrating on leading the change: this may be focused on the IT industry but is overflowing to other industries too.

These issues lead to:

- Increased Competition,
- Increased Change and
- Increased Complexity.

2. USER NEEDS AND TECHNOLOGY GAP

Virtual Enterprises

Facing an increasing complexity of product development along with an intensifying market competition, the Virtual Enterprise (VE – [Hard96], [Hard97]) appears nowadays as a necessity within nearly all the industrial fields. The necessity for VEs can be illustrated by the results of increased competition, change and complexity:

- *Best of Breed*: Firms have to provide Best of Breed solutions which means that they now have to concentrate on their own core competences and outsource for the right quality components/services for the right project. Effective outsourcing can necessitate partnering with suppliers within a VE.
- *Time to Market*: The race in time-to-market necessitates shorter development times which is enabled by re-use of outsourced components. Again VEs are vital.
- *Shorter Product Cycles*: Increased change and competition necessitates increased agility which can only be achieved by a flexible organisation. Such an organisation concentrates on its core competence and re-invents its offerings by re-configuring its VE per project.

Information is Power

Information is power and lack of information is lack of power over one's firm's destiny and deliverability. With increased competition, change and complexity it is increasingly important to control project: *Time*, *Cost* and *Quality*. Increased complexity requires more control in product development comprising: *Functionality*, *Aesthetics*, *Performance*, and *Reliability*. In VEs, firms need the ability to transact business processes seamlessly and to capture, access and assess the state of the business/project.

The Technology Gap

Current technology solutions have one or more of the following characteristics:

- ❑ *Homogeneity*: Lack of support for different systems.
- ❑ *High Entry Level*: Expensive to buy into.
- ❑ *Lack of Scalability*: Limited growth path.
- ❑ *Application Centric*: Need to organise the enterprise around the application.
- ❑ *Fixed Infrastructure*: Need for Leased Lines between partners restricting location independence and requiring long term relationships.
- ❑ *Lack of Support for Business Processes*: Limited security and transactional support.

The drive towards VEs is however characterised by partners who are smaller, more numerous, changeable, distributed, and with heterogeneous data and applications. The construction industry is quite a good example of such a situation. Like many other engineering sectors, the end product of construction is a value-added arrangement of standard component parts², designed and constructed by non co-located teams of separate firms who come together for a specific project. The design team comprises up to 7 disciplines: property developers, project managers, civil engineers, architects, surveyors, building services engineers, and contractors, all of them consulting regulatory, best practice, pricing, aesthetic and proprietary product information, and design services. A typical example is the need of construction industry firms to control the logistics and transactions in dealing with the suppliers, which is particularly important in the construction industry where very many suppliers are involved in any one project and they vary greatly from one supplier to another.

To enable VEs by providing the ability to transact business processes seamlessly and to capture, access and assess the state of the business/project, industry needs a combination of:

- ❑ *Low Entry Level*,
- ❑ *Scalability*,
- ❑ *Open Infrastructure & Location independent access*,
- ❑ *Enterprise Information*: Seamless capturing of the state of business from distributed legacy data,
- ❑ *Support for Business Processes*,
- ❑ *Security and Transactional Support*.

The WONDA project aims to satisfy these needs.

² But a major difference with a lot of other engineered products (like planes, cars, ships, etc.) is that almost each building is a unique prototype, and the once off product around which the factory (the site) is built.

3. THE WONDA PROJECT VISION

The WONDA vision and product objective

WONDA aims to develop an open and secure framework for business objects and electronic payment. In this context business objects (BOs) are software components which encapsulate business rules and procedures and which can run anywhere on the network. They aim at providing customers with secure and sophisticated access to diverse electronic content and software components. BOs are defined as components in the information system representing the enterprise model and promise to be the building blocks of information systems meeting end user's requirements critical to the success of the enterprise.

WONDA will use the following technical characteristics to meet the needs of VEs:

	Low Entry Level	Scalability	Security & Transactional support	Open Infrastructure	Support for Business Processes	Enterprise Information
Business Objects		*			*	*
Distribution Middleware			*	*		*
Web Infrastructure	*	*		*		
Modular Security	*	*	*			
Database Federation		*				*
Standards	*	*		*		

Openness will be WONDA's differentiating feature especially in terms of:

- ❑ extendibility (both functional and schema oriented),
- ❑ standards (which in turn enable database and platform independence),
- ❑ scalability (both architectural and performance).

Extensibility will be enabled by schema federation and configurability. Standards will be aligned to W3C and OMG. An open, modular and configurable framework will enable scalability. Especially, the architectural hub of WONDA will be the *WebMapper*, being the enabler for BOs and electronic payment. It will enable customer configurability for ODMG as well as STEP/SDAI ([Fowl95], [ISO95]) based solutions and for security solutions, providing an open framework for Web Business Objects such as construction software and financial knowledge. These features will be achieved through a symbiosis of:

- ❑ federated database technology,
- ❑ generic mapping technology between the data from the database federation to the Internet environment,
- ❑ security components,
- ❑ domain specific environments offering the WEB-interface tailored to the end-user requirements.

Product definition: the main components of WONDA

WONDA will specify and develop a three-layered architecture based on a set of existing and emerging *de jure* or industrial standards:

- ❑ federated database level in order to search and access information within various heterogeneous databases throughout a unified interface (OMG/ODMG'93, STEP/SDAI).
- ❑ WebMapper, which is a schema mapping layer enabling electronic knowledge bases to be seamlessly interconnected, and thus dealing with availability and interoperability issues. This mapper will ensure the liaison between the federated database level and any Internet/Intranet application in a model, language and application independent way ;
- ❑ third level defining "plugs-in" for domain specific applications, for intuitive access to WEB information by end users using their usual applications.

A last specific WONDA development will be a security toolkit for data protection and confidentiality levels configuration, so that applications can be securely linked, thus enabling WONDA to be exploited for real industrial and commercial applications.

Hence, WONDA focuses on a framework for business solutions and open interchangeable software tools. It will develop an infrastructure fully dedicated to the WEB interconnection, and based on major standards. Along with uniform access to any database, the objective is to extend the openness of the platform towards WEB standards and applications via the WebMapper for information presentation and query, and at a generic level in terms of standardised models for presentation and not from a tool-oriented point of view. Thanks to WONDA, coupling of WEB data with internal corporate information memory including data managed by Intranet applications will become possible, along with the development of sector specific software for Internet servers and browsers with a plug-in³ based technology.

4. BUSINESS OBJECTS: CONCEPTS AND INTEGRATION IN FLEXIBLE ENTERPRISE INFORMATION SYSTEMS

Introduction and technology state-of-the-art

Despite notable advances, software technology still looks for improvements with respect to flexible and reusable components, for reactive and adaptable corporate information systems which have become the backbone of modern companies. Actually, the enterprise information system is oftentimes a complex and inflexible mix of old fashioned solutions. Regarding component issues along with the WEB, various enterprise object frameworks coupling multiple relational DBS and generating HTML pages with database data have been developed. An interesting example of such a framework are the WebObjects⁴, but they

³ A plug-in is defined as a small piece of dedicated software, dealing with specific kinds of (information) format and/or specific functionality. Thus, it can add several new features to a larger software through communication with this software, and extensions of programs are then naturally possible in a dynamic way using as many plug-ins as required, without the necessity of integrating all the features into a single huge program. This concept of plug-in can be implemented according to various technologies, depending the fact that the extended tools are WEB tools (WEB plug-in) or specialised tools (Intranet plug-in). A WEB plug-in is a software extension dedicated to a specific WEB browser and specialised in the management of a given type of document, most of the time identified by its format: mpeg, vml, etc. The actual main issue is that current plug-ins are developed for a given browser, and thus the end-user becomes fully dependant of this browser and its plug-ins.

⁴ Information about WebObjects can be found at the following URL: <http://www.apple.com/webobjects>

federate relational databases only, and don't provide language independence nor security. On the other hand, the WebBots⁵ components are dynamic objects evaluated when saving or browsing WEB pages: they allow the query of WEB servers, the updating of WEB pages each time content changes, the inclusion of other pages or images on a page, and adds full text-searching capability. They also grant the insertion of advanced components through the use of scripts (VB script, Javascript) or Java applets. WebBots embody valuable features, but they only deal with client side interfaces, and make transactions with WEB servers, not enterprise servers. Another more open solution are the well known JavaBeans, relying on the Java language and platform, supposed to be portable on any hardware and operating systems, but even JavaBeans focus on graphical components and environments for end-user interfaces.

Tackling the more general concept of intelligent agents, most of the today's agent-based systems are custom architecture or proprietary frameworks (with internal models, legacy script languages, connection to legacy databases). These lead to less interoperability with other applications, poor integration in large enterprise information systems, and above all, less ability to extend (e.g. by direct integration of other pre-built components not based on the same agent network) and to follow the market evolution, even if some of them are business-oriented. Thus technological state-of-the-art does not reveal today's systems or frameworks providing at the same time the following features:

- ❑ uniform access to the federation of all enterprise data;
- ❑ language neutrality;
- ❑ clear separation of data content and appearance;
- ❑ secure access to enterprise data;
- ❑ business oriented access to information;
- ❑ all models and components based on world-wide recognised and adopted standards.

A new solution coming to help with these issues expands from *OMG* with *Business Objects* and the Business Application Architecture⁶. It intends to promote a standard framework for business application residing on top of CORBA ([Mowb96], [Sieg96], [Orfa96]). BOs are the "glue" between client applications and enterprise data contained in large data stores. They are expected to facilitate communication, design and modelling between implementers and business domain experts through the sharing of the same concepts. They have to model the real world so that people focus on main characteristics and relationships among BOs. They can be associated to software components, and therefore can be assembled into *frameworks* to support high-level industrial product design and developments. When considering *distributed architectures*, which are the ultimate solution for the interoperation of business components in heterogeneous user views on product and computer implementation of the product data. Thus, while product data models correspond to a conceptual structural approach of data, the BOs (and software components) conform to a more functional and process view on information, though relying of course on data structures. BOs are concerned with the definition of methods and operations available for objects, and corresponding possible queries on these objects.

⁵ Information about WebBots can be found at the following URLs:

<http://www.internetbay.com/bots.htm>

<http://www.netjammer.com/TRAINING/HPD/engines.htm>

<http://websupport.net/fpguide/fp97webbots.html>

⁶ The concept of BO is currently under standardisation by the OMG, together with a standard framework for business applications (see the following WEB site: <http://www.tiac.net/users/jsuth/oopsla/bowp2.html>).

They equally have to manage information on objects, identifying constraints and relationships according to some process or business context. They can be related to ways of retrieving information like object browsers, query languages, keywords-based technologies, etc.

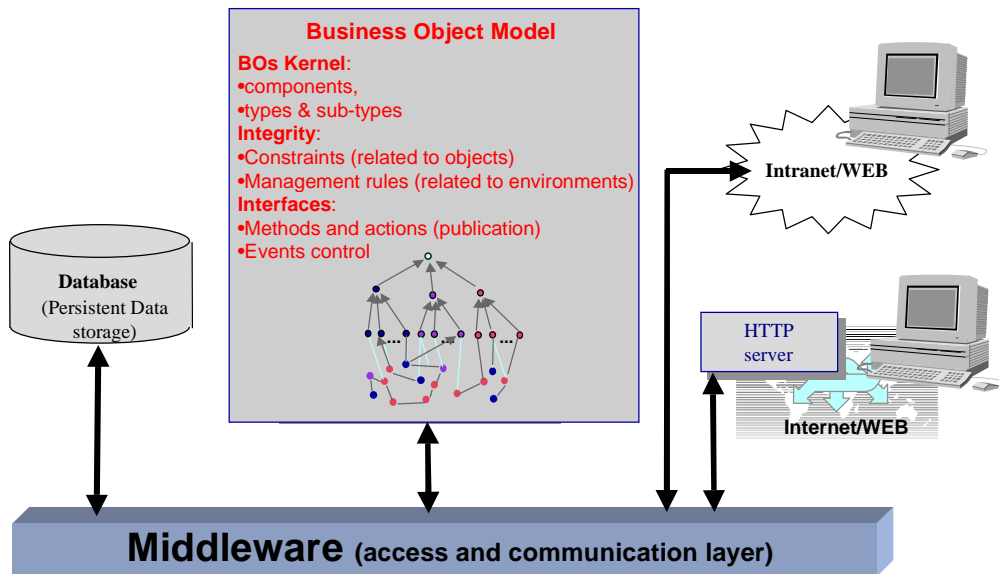
Definition of a BO

A BO is a concrete representation of an active thing in a specific business domain in the real world, e.g. it typically stands for employees, products, shipping, and so on. Thus, a BO may act as a participant in a business process and try to mimic the way things are. Conceptually, BOs add value over other representations because they give a higher level view and package the main concepts of the business model they represent. As we stated earlier, BOs stand for active things found in the real world. More precisely, a BO is uniquely identified by its name in a business domain. Based on an underlying object technology, a BO is a set of:

- Attributes, which stand for associated data elements which provide information about the BO (transient or persistent data the BO holds and which compose its internal state).
- Operations, that is its behaviour,
- Constraints, that is not only integrity constraints, but also triggers business rules it must ensure in order to fulfil the requirement expressed by the end user.
- Relationships, are the sets of objects it is in relation with. Relationships help to make components interact. Without any relationships, components would be isolated islands of knowledge. They allow bi-directional traversal between instances of different types. Moreover, they permit loose coupling of co-operative distributed components and a way to easy assembly components originated from different vendors.
- Events that externalise state changes when operations are invoked on BOs. An event model is intended to avoid tight coupling while integrating several BOs in a specific system. By doing so, changes are easier to manage.
- Business rules that focus on the semantics of the component in a specific domain. An appliance rule is plugged into a container on which the rule applies. The BO's behaviour is partly defined by this sort of sub-component. For example, invariant appliances define rules that must always hold for the component.
- States that symbolise the mutually exclusive conditions a component may be in. Moreover, some states may be reached only when pre-defined conditions yield.

BOs and Three-Tiered infrastructures

In order to take full advantage of them, BOs are certainly to be considered in the context of a full distributed architecture based on middleware technology (i.e. on a software bus in charge of communication between objects, like an ORB or a DCOM bus). Indeed, BOs must support distribution to be easily integrated in client/server architectures, and they promote the nowadays well-known 3-tier (and even n-tier) architecture. They aim at being distributed on remote sites for specific use and even, while at the same time being accessible in a seamless and transparent way. They constitute the fundamental bricks of the "business-centred" middle tier of today's emerging architectures.



The *data storage and server* level (DBMS) allow access to basic data within DBMS through gateways (native gateways for direct connection to legacy systems, co-operative gateways using the network interface of the accessed database, or procedural gateways for access to the database through inside procedures or TP monitors), or through a middleware layer in a distributed environment (CORBA/ORB, DCOM, etc.).

The *client applications* level is mainly concerned with GUI⁷. WEB de facto standards (HTML, VRML, etc.), offering standardised interfaces to clients and dealing with Intranet as well as Extra/Internet users. Depending on the fact that client computers are PC or NC⁸ ones, or powerful workstations, specific treatments can be localised on the client side.

The *Middleware* level has to manage generic services such as information search, object or server identification, communication between objects through messages transport, marshalling and unmarshalling, etc., at the level of the distribution layer. Eventually, global rules (not related to business objects, but rather to their inter-working in the context of the full enterprise model) can be handled at that level too, like rules defining actions when a given program execution has failed, actions to undertake in some specific context defined through global variables, etc.

Eventually, the *Central tier (business)* level contains the embedded business logic, with one (or several) BOs model(s). From a client point of view, this business model expresses the business domain (without technology consideration), and provides the end-user with some unified view on data hiding the underlying data sources. From the architecture viewpoint, it implements the business rules and processes, dealing with complex operations on the business model such as object mappings or links, views filtering, model conversion/transformation, fine-grained security issues, possible client interface rules (if based on presentation standards like in the WEB technology) and so on. A flexible and evolving architecture must integrate models and rules so as to control treatments on engineering knowledge, both at the level of managing and massaging the information and at offering various views and interfaces to client applications and end-users. This role is indeed devoted to the central tier component,

⁷ Graphical User Interface.

⁸ Network Centric.

acting as a specific applications server for fundamental services encapsulating objects and views with their procedures and rules in BOs as a modelling of the business enterprise.

Existing implemented model for BOs

The emerging technologies presented hereinafter can be considered as implemented solutions of a model and application architecture for BOs. Nevertheless, it is worth noticing that these developments have been undertaken regardless of the specific current standardisation efforts of the OMG with respect to Business Objects Facility and Business Application Architecture. They are proprietary SUN (*Enterprise JavaBeans - EJB*) and Microsoft (*Active X components*) solutions to deal with business oriented framework and components. As based on Java, the Enterprise JavaBeans, however, can be considered, at least at the moment, as the most open framework. Moreover, BOs are supposed to rely on distributing computing: the most currently achieved specification is the OMG CORBA⁹ [OMG95a], but even other distributed-object systems, like OLE/DCOM for instance, are candidates to support BOs. In any case, the integration of an ORB is a requirement for the real deployment of a BOs application infrastructure, allowing any application on the network to deal with BOs.

Enterprise JavaBeans

Java (libraries and first Java oriented IDEs¹⁰) initially provided only the opportunity to client applications for dynamic WEB user interfaces, with sometimes some ready-to-use simple components (client-side JavaBeans). The JVMs¹¹, allowing execution of Java applets, were not designed to support enterprise applications servers, especially because of a lack for essential support of transactions. But the on-going definition of the EJP (*Enterprise Java Platform*) an EJB has the objective to standardise the services and API required for object oriented distributed applications based on Java, i.e. transactional services, objects lifecycle, synchronous or asynchronous objects interoperability, security aspects, and so on. In a similar way JavaBeans describe an API for reusable graphical object components (but not adapted to large enterprise systems of today), the *EJB*¹² provide a component architecture for development and deployment of distributed enterprise wide objects. Applications written using EJB should be deployed on all Java-enabled server systems.

The EJB model aims at offering distributed scalable, transactional and reusable components, and any *Java* enabled platform may run these components, provided they use a corresponding *EJB* component enabler. EJB extend the Java model on the server side, and aim at accommodating components for large transaction-oriented applications, tying legacy data to Intra/Internet clients, under the control of the business logic coded in the EJB on a middle tier. It is worth noticing that, in its third release of CORBA, due by the fourth quarter of 1998, the OMG plans to incorporate the Java Beans model in order to deliver *CORBA Beans*. This will both enable rapid development of CORBA objects and integration using graphical tools. Moreover, CORBA objects may be easily and seamlessly combined with Microsoft's Active X components. With the emergence of EJB, a plentiful market for ready-to-use JavaBeans components for client and servers can now be deeply envisaged.

⁹ It is a requirement of the CBO-BOF RFP that BOs should be CORBA-based BOs.

¹⁰ Integrated Development Environment.

¹¹ Java Virtual Machine.

¹² Indeed, the EJB extends the JavaBeans component model to the server side.

Active X components

Besides *de facto* standards as those promulgated by the OMG, Microsoft is developing more proprietary solutions for distributed architectures. Microsoft solutions have similar properties to other distributed architectures, but don't provide platform independence. Active X is an extension of OLE¹³ for the specific context of the Internet/WEB. This model allows to get benefits from a component-based approach across a broader scale of multi-user applications in a distributed component architecture, and is supposed to rely on DCOM¹⁴, a technology enabling software components to communicate directly with each other across networks, including Internet/Intranets. Active X is the Microsoft infrastructure for compound documents, communication between clients and servers of distributed objects, and the integration of objects within WEB pages for Intra/Internet applications. Together with Windows DNA, a unified approach for building distributed, scalable, multi-tier applications that can be delivered over any network, they form the first application architecture to integrate the Internet, client/server, and PC models of computing for new distributed computing solutions.

BOs in the WONDA project

The major problem today is certainly the absence of standards for manufacturing BOs, and thus no consensus with respect to the interfaces to provide for BOs. Standards like STEP or the IFC normalise product data entities at an appropriate level for data exchange and sharing, but not for interoperable product components, nor do they specify a framework supporting component interoperability. In that context, the current efforts of OMG are of primary importance, both for the standardisation of Common BOs (CBOs) and domain specific BOs as well. These standards have to focus on the interfaces (API) which define the communication layer, whilst the communication mechanisms themselves will be implemented by a tier middleware (the choice for OMG is naturally CORBA with its services [OMG95b]).

BOs will be implemented in WONDA through the WeBuild and WeBank components. Though dealing with more product-oriented considerations with WeBuild and more financial-oriented considerations with WeBank, the main idea in WONDA is to define BOs from a broad point of view, i.e. providing different end users views (like engineering or financial information views), but also realising the links between information related to engineering product data, financial data, process (workflow) oriented information, and so on. For instance, these BOs will aim at making the correlation between some engineering data (as stored, for instance, in an SDAI database) and some financial information (stored in an ODMG database), while at the same time masking the internal representation of the various stored information, and encapsulating the rules and relationships/associations associated to the BOs. Indeed, just as a building is a unique arrangement of standard products, a project is a unique arrangement of financial, logistic and technical product data, and the role of the BOs is to translate it in a Business model. Thus, BOs will realise the interoperation between engineering product data and financial product data.

¹³ Object Linking and Embedding is a set of libraries and applications for storage, data exchange and integration of elements within compound documents.

¹⁴ Distributed Component Object Model.

5. OVERVIEW OF THE WONDA PILOT

The WONDA pilot will simulate a contractors VE, accommodating heterogeneous information systems and the ability to transact project business processes seamlessly and to capture, access and assess the financial, technical and logistic state of the project. The pilot will federate heterogeneous project data such as CAD drawings, document archives, project plan, material plan and cashflow plan. Managers anywhere in the VE can then access financial, technical and logistic overviews of the state of the project. They will be able to transact project business such as ordering, paying, approving goods, documents or services.

BOs will map from legacy and heterogeneous data to the business model whose rules and relationships and integrity is encapsulated by the BOs. They will comprise building block business objects known as CBOs, which support generic services such as security, transaction and legacy database mapping. Client BOs will be web enabled plug-ins to the clients workbench whether that is a browser, CAD or procurement application.

The pilot will be built up in incremental steps to demonstrate low entry level and the ability to scale through intermediary steps according to the business value offered.

1. *Cashflow linked to Materials Plan & Account Query*: This enables the contractor's financial manager to plan its cashflow for materials expenditures which typically accounts of 40% of project expense
2. *E-Commerce to Suppliers*: Contractors can pay suppliers by electronic fund transfer.
3. *Materials Plan linked to Project Plan*.
4. *Project Plan linked to Estimation and Tender Documents*.
5. *Project Plan linked to Drawing and BoQ*: Enables 4D CAD where the 4th dimension is time line of project planning or expressed another way enables Project Plan visualisation. In effect just as a building is a unique arrangement of standard products, a building project is a unique arrangement of logistic, financial and technical product data. Such product data are the building blocks of a project-centric enterprise.

6. CONCLUSION

WONDA aims to meet requirements typical of the construction industry and central to the competitiveness of firms within the industry. WONDA will enable take-up of IT systems for VEs by delivering low entry level, scalability, open infrastructure and location independent access, seamless enterprise information, transparent support for business processes, security and transactions. A central enabling technology exploited by WONDA is BOs, which are defined as components in the information system representing the enterprise model and which promise to be the building blocks of information systems meeting requirements critical to the success of the enterprise. WONDA will execute a construction industry pilot which will simulate a contractors VE, accommodating heterogeneous information systems and the ability to transact project business processes seamlessly and to capture, access and assess the financial, technical and logistic state of the project.

WONDA endeavours to develop the vision of a standardised open WEB-oriented framework for information access and electronic commerce. The combination of new concepts (including BOs) and technologies based on standards (either *de jure* ones like official norms, or *de facto* ones in their large industrial use but still leading to openness), is supposed to open the door for substantially enhanced integration of value-added networks, high level application development, and distributed object middleware. This combination will radically simplify and

reduce time for the development, deployment and management of distributed applications providing corporations with a competitive advantage, especially on the Internet market.

7. REFERENCES

- [Fow195] J. Fowler. STEP for Data Management, Exchange and Sharing. Technology Appraisals 1995.
- [Hard96] Hardwick M., Spooner D. L., Rando T. and Morris K. C. : « Sharing Manufacturing Information in Virtual Enterprise », Communications of the ACM February 96 Vol 39 N°2.
- [Hard97] Hardwick M., Spooner D. L., Rando T. and Morris K. C. : « Data Protocols for the Industrial Virtual Enterprise », 1997.
- [ISO95] Industrial automation systems and integration - Product data representation and exchange Part 22. Standard Data Access Interface. 1995.
- [Mowb96] T. J. Mowbray, R. Zahavi. The Essential CORBA - System Integration Using Distributed Objects. John Wiley and Sons.
- [OMG95a] The Common Object Request Broker Architecture and Specification (CORBA) Revision 2.0. July 95. URL <http://www.omg.org>.
- [OMG95b] Object Management Group (OMG). CORBA Services: Common Objects Services Specification, Revised Edition, 95-3-31 ed, March 1995.
- [Orfa96] R. Orfali, D. Harkey and J. Edward. The Essential Distributed Objects Survival Guide. John Wiley and Sons Editor.
- [Sieg96] J. Siegel & Al. "CORBA Fundamentals and Programming," John Wiley and Sons, April 96.