

## QTECH Hybrid Systems Inc.

*An eBusiness Components, Tools  
and Solutions Company*

Toll Free: 877-748-7919  
Tel: 613-233-8939  
Fax: 613-667-3553  
Internet: [www.qtech.ca](http://www.qtech.ca)  
Email: [info@qtech.ca](mailto:info@qtech.ca)

200-38 Auriga Drive  
Ottawa, Ontario  
CANADA K2E 8A5

Contact:

**Bill A.V. Pezoulas**  
CEO & President

Direct: 613-724-9726  
Email: [bill.pezoulas@qtech.ca](mailto:bill.pezoulas@qtech.ca)

## The Reality of Application Delivery Using Componentized Software

Imagine an application development process that costs 80%-90% less than the current state of the art in object-oriented development. This application delivery process consists of selecting business domain components from a third party or corporate catalog, extending them to suit your unique business processes, and using these business objects along with standard application elements to assemble full-featured applications. The process is completed by selecting and configuring standard datastores, connections to legacy systems, networking elements, server containers and desktops required for deployment.

Until now Component Based Development (CBD) has fallen short of producing such a revolutionary leap forward in productivity. However, the new EJB 3.0 Java Persistence technology and other recent developments have reconfirmed the fundamental value of the discipline and are moving it toward realisation in the mainstream.

QBeans, an Application Development Framework created by **QTECH Hybrid Systems Inc.**, is an enabling technology that supports this vision of CBD.

## Background

IT departments seek to develop business applications that:

- solve defined business problems;
- respond to current user requirements and evolve with requirements and technologies; and
- are cost effective to create, deploy and maintaining

Object Oriented Analysis, Design and Programming evolved from the desire to improve software engineering processes to the level of other engineering processes.

The concept of reusable software components was created to encapsulate best of breed solutions into components that can be easily combined to create numerous applications. Reusable components can be created to address business and/or technical problems.

Component Based Development (CBD) is a discipline that increases the predictability of software systems delivery by using componentization and reuse to decrease technical and schedule risk.

---

## The Problem

---

In practice, most currently available reusable components solve technical problems. For example, a ComboBox control addresses a common GUI challenge and JDBC Drivers address the challenges of accessing relational databases in a uniform fashion. These types of components accelerate the development process because they encapsulate the solutions to specific technical problems. The technological characteristics of the target deployment environment determine the components chosen. Applications built with these components are therefore tightly coupled to the deployment environment.

Although technical software components do accelerate the software development process to a certain degree, they have the following limitations:

- **They limit the lifespan of the application** The value of these components is rooted in the technological problems they solve. As technology advances, these problems change and so too must their solutions. Later versions of an application often have to be re-engineered in order to replace technologies that are no longer supported or that do not support new requirements, or to take advantage of newer technologies that were not available when the project started. The benefits of these components are therefore short-lived, which diminishes their value over time.
- **They distract from business problems** The technological nature of the components keeps the development team focused on technical rather than business aspects of the application, which leads to applications that are tightly coupled to a particular deployment environment. Current technologies make it difficult and costly to maintaining a loose coupling between the technology and the business application.

CBD has yet to deliver on its promise of allowing cost-effective assembly of applications because technology is constantly changing and most componentization efforts have been applied to solving technical or technological problems. Inadequate investment has been made in componentizing business domains such as Sales, Finance, and Human Resources into business objects, services and processes. Also lacking are the “Business Domain Models” that can be used in the assembly of business applications. A notable exception to this can be found in the healthcare sector. See [www.hl7.org](http://www.hl7.org) for information on the efforts of a healthcare-focused standards development organization to standardize the Business Domain Model for their industry (more on this in a future article by the author).

Commercial Off the Shelf Software (COTS) applications have evolved in particular business domains to provide cost-effective solutions to the specific needs of a particular industry sector. These applications are usually architected as a proprietary technology solution and try to “be all things to all people” in a particular business domain, but fall far short because they end up being:

- too complex, having features that are not required in all situations,
- difficult to configure and extend, often leading to unanticipated costs

when firms configure the COTS application to meet the specific needs of the individual business,

- not quite suited to the purpose, frequently requiring the customer to mould its business to the application.

## The Requirement

---

Clearly, there is a need to focus on business domain component solutions that can be deployed into a variety of technical environments. This will eliminate the application level buy vs. build dilemma and enable the creation of a whole new generation of business applications based on standard, extensible Business Domain Models and standard technology solutions that can evolve independently in response to changing business requirements and evolving technology.

The business solution must be designed to reflect the business needs of the target user community while respecting the limitations of available technologies. Similarly, the design of the technology solution should reflect factors such as the geographical distribution of the users, security considerations, activity projections, interactivity requirements, data volumetrics, all of which are derived from the business needs.

The different emphases of the technology and business solutions must be reinforced through standards that recognize the value of maintaining these differences. Products based on such standards would enable organizations to cost-effectively create information systems that:

- Closely match the needs of their unique business processes and can evolve along with these processes,
- Make effective use of current technologies and can easily evolve to take advantage of new technologies.

## The Solution

---

Industry has long recognized the dichotomy of the business- and technology- domains and has only just started to develop standards for solutions to common technology problems, design patterns that express best practices, and finally frameworks of reusable components that emphasize the separation of the Business Domain Model from the technology solution. These efforts will result in:

- substantial software development productivity gains,
- a growing body of technology-independent business solutions,
- an alternative to both the traditional custom development and the COTS approaches to large-scale enterprise application development.

Over the past five years, the following innovations in the Java technical domain have brought together many of the elements necessary to produce a revolutionary leap forward in the CBD process.

- Java and its related technologies and open standards have been accepted as the de-facto standard for platform independent applications,
- The Java community is committed to open specifications and an open source model for many critical technologies,
- The Java EE architecture for enterprise applications includes established specifications and standards relevant to:
  - transaction servers (Enterprise Java Beans (EJB)),
  - web-based applications (Java Server Faces(JSF), Java

- 
- Server Pages (JSP) and Servlets),
  - rich client applications (Java Foundation Classes (JFC)),
  - messaging (JMS),
  - access to relational databases (JDBC)
  - Business Domain Objects can be transparently persisted using the new EJB 3.0 Persistence APIs (or its predecessor – Java Data Objects (JDO)). These technologies decouple business objects from the persistence mechanism, eliminating the last impediment to creating libraries of truly portable and technology independent business objects.

Two additional elements are currently missing from the marketplace, and present a significant opportunity for innovation:

- Rapid Application Development Frameworks for web services, browser-based and rich-client applications that incorporate the relevant standards and best of breed technology solutions.
- Extensible and technology independent business domain models that encapsulate comprehensive solutions to common business domain requirements. A truly productive CBD process will become a reality once these two elements are widely available from multiple vendors.

## **QBeans: A Rapid Application Development Framework for CBD**

---

QBeans™, developed by QTECH Hybrid Systems Inc., is an enabling technology that supports CBD by emphasizing the segregation of business domain from technology considerations.

QBeans™ is a Rapid Application Development framework for business applications that is designed to assist developers and business analysts to deliver robust, highly functional desktop rich-client applications while minimizing the Java skillset and coding / testing effort required. QBeans™ consists of:

- a set of Java Beans application components based on the Java Foundation Classes (JFC)
- persistence adapters based on the Java Data Objects (JDO) and (future) EJB 3.0 Persistence specifications
- a flexible development process for creating best of breed applications, and
- a set of design patterns that incorporate best practices.

QBeans™ enables users to easily create libraries of reusable business components that are independent of both application presentation and data persistence technologies. Fully functional business applications can be rapidly assembled using these technology-independent business domain objects and QBeans™. These applications are deployed into appropriate datastores, such as relational or object databases, and into middle tier application servers through third party JDO or EJB 3.0 persistence implementations as appropriate to the technical requirements of the environment.

The next generation QBeans™ products will extend the development productivity benefits to multi-tier applications.