

Visualizing and Assessing a Compositional Approach to Service-Oriented Business Process Design Using Unified Modelling Language (UML)

Yusuf, Lateef Oladimeji

Department of Computer Science

University of Agriculture, Abeokuta, Ogun State, Nigeria

E-mail: truevisionconsulting@yahoo.com

Olusegun Folorunso

Department of Computer Science

University of Agriculture, Abeokuta, Ogun State, Nigeria

E-mail: folorunsolusegun@yahoo.com

Akinwale, Adio Taofeek.

Department of Computer Science

University of Agriculture Abeokuta, Ogun State, Nigeria

E-mail: atakinwale@yahoo.com

Adejumobi, A. I.

Department of Electrical and Electronics Engineering

University of Agriculture Abeokuta, Ogun State, Nigeria

E-mail: engradejumobi@yahoo.com

Received: March 14, 2011

Accepted: April 5, 2011

doi:10.5539/cis.v4n3p43

Abstract

In the context of Service-Oriented Architecture (SOA), complex systems can be realized through the visualization of business driven processes. The automation of Service Supported Systems (SSS) is the future integral part of core SOA which provides preprocessed information and solution suggestions for the Cloud Computing Users (CCU). CCU requires compact and fast decision supporting displays and user interface in order to handle the increasing work load. This requires intelligent, intuitive and robust preprocessing system as a backbone for automation lifecycle management. Complex business management processes often entail complex environmental decision-making procedures. This process can be greatly enhanced if it is based on an exploratory-envisioning system such as Information Exploration and Visualization Environment. Current scientific research has taken advantage of e-science to enhance distributed simulation, analysis and visualization. Many of these infrastructures use one or more collaborative software paradigms like Grid Computing, High Level Architecture (HLA) and Service Oriented Architecture (SOA), which together provide an optimal environment for heterogeneous and distant, real-time collaboration. While significant progress has been made using these collaborative platforms, often there is no particular software suite that fulfils all requirements for an entire organization or case study. End-user must cope manually with a collection of tools and its exporting/importing capabilities to obtain the output needed for a particular purpose. We presents how service oriented architecture can be utilized in automation services support system using RCD framework as underlying composition platform. The introduced framework combines rapid analysis development and intelligent process state visualization for CCU and discusses the challenges met in building reliable cloud computing services for web services. Unified Modeling Language (UML) is used as a specification technique for the system analysis

and design process which is the only way to visualize one's design and check it against requirements before developers start to code.

Keywords: Cloud computing, Service, Visualization, Service Oriented Architecture, Unified Modeling Language, Software Architecture

1. Introduction

An application that follows the Service Oriented Architecture paradigm (MacKenzie, 2006) is an assembly of services that realizes business processes. Business processes are designed by business specialists and typically involve many services that are composed in a variety of ways. The need to extend a SOA application with new business features (to follow market trends) arises often in practice. In the technological context of Web Services, business processes can be implemented as orchestrations of services (Peltz, 2003). Existing tools and formalisms related to business processes are essentially technologically-driven. They use a design in the large approach and do not intrinsically provide language constructions and frameworks to support the introduction of new features into existing processes. The traditional software systems are built upon platforms and Programming Languages which are tailor-made for a particular purpose, not easily extended to support a wider sharing of resources and collaborative work. Recently quite numbers of works have been done on visualization, but most of these works have not actually considered the possibilities of Service-oriented Applications and the advantages these can bring to the IT business environment. Past Literatures treat SOA application designs with mere functional demands and requirements, it is believed that how things appear and feel is an important factor to the actual end users. The traditional approach to information visualization tools uses visualization as the first and last step of a business process; it fails to take advantage within the intermediate process. Service can be introduced to take care of the intermediate process. Services as an abstraction of functionality can enable the visualization of a system that has well-defined processes with relative ease. This can lead to aspirations for achieving greater complexity with the Service-Oriented Architecture paradigm. SOA is enabled through an interconnected set of services, each accessible through standard interfaces and messaging, it offers functional abstractions that are extensible, loosely-coupled and reusable. These characteristics drive the vision of a flexible and distributed infrastructure that supports on-demand business needs. For Example, Using web services abstract process workflows enables the orchestration and interactions among several distributed services over the internet using specifications such as the Business process Execution Language (BPEL).

Cloud computing is an Internet-based computing environment, whereby shared resources tools related to its information are provided to computers and other devices on demand through services. Cloud computing is a natural evolution of the widespread adoption of virtualization, Service-Oriented architecture and utility computing. Details are abstracted from users, who engage the tools for their day today design through the web and the technology infrastructure "in the cloud", that supports them (Danielson, 2008). It describes a new supplement, consumption, and delivery model for IT services based on the Internet, and it typically involves over-the-Internet provision of dynamically scalable and often virtualized resources (Gartner, 2010 and Grumen, 2008). This frequently takes the form of web-based tools or applications that users can access and use through a web browser as if it was a program installed locally on their own computer. Typical cloud computing providers deliver common business applications relating to online that are accessed from another Web service or software like a Web browser, while the software and data are stored on servers. Most cloud computing infrastructures consist of services delivered through common centers and built on servers. Clouds often appear as single points of access for consumers' computing needs. Commercial tools offerings through the web will generally expected to meet quality of service (QoS) requirements of customers, and typically include service level agreements (SLAs) (Buyya et al., 2008).

Recently, there has been a growing tendency to adopt UML (Unified Modeling Language) for different modeling needs and domains. UML diagramming practices have been applied for specification techniques, designing and modeling various information systems so as to improve technical accuracy and understanding in requirements related with this information system (Kanwalvir and Himanshu, 2011). UML offers vocabulary and rules for communication and focus on conceptual and physical representations of a system. It uses an object oriented approach to model systems which unifies data and functions (methods) into software components called objects. Various diagrams are used to show objects and their relationships as well as objects and their responsibilities (behaviors). It is a standard for object-oriented modeling notations endorsed by the Object Management Group (OMG, 1999), an industrial consortium on object technologies. UML has become a standard after combining and taking advantage of a number of Object-Oriented design methodologies (Kobryn, 1999) and is currently posed as a modeling language instead of a design process. We applied a subset of UML diagrams for modeling RCD Beam information system.

2. Literature Review

2.1 Software Architecture

To understand how SOA works, it is imperative to understand the concept of software architecture, this will assist to construct a functional architectural design for a characteristic case study. Software architecture has emerged as an important sub-discipline of software engineering, particularly in the realm of large system development. While there is no universal definition of software architecture, there is no shortage of them, either. The following are a few of the most cited ones:

- Bass, Clements, and Kazman, 1998: *The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. By "externally visible" properties, we are referring to those assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on* (Bass et al., 1998).
- Garlan and Perry, 1995: *The structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time* (Garlan and Perry, 1995).
- Garlan and Shaw, 1993: *Beyond the algorithms and data structures of the computation; designing and specifying the overall system structure emerges as a new kind of problem. Structural issues include gross organization and global control structure; protocols for communication, synchronization, and data access; assignment of functionality to design elements; physical distribution; composition of design elements; scaling and performance; and selection among design alternatives* (Garlan and Shaw, 1993).
- Perry and Wolf, 1992: *A set of design elements that have a particular form* (Perry and Wolf, 1992).

Earlier Architectural styles that can be found in Shaw and Garlan (1996) and Buschmann et al. (1996) are; Layered, Pipes and filters, Model-View-Controller, Presentation-Abstraction-Controller, Reflective, Micro Kernel, Blackboard and Broker architectural framework.

2.2 Unified Modelling Language

UML is a complete language for capturing knowledge (semantics) about a subject and expressing knowledge (syntax) regarding the subject for the purpose of communication. It applies to modeling and systems. Modeling involves a focus on understanding a subject (system) and being able to communicate this knowledge. It is the result of unifying the information systems and technology industry's best engineering practices (principals, techniques, methods and tools). It is used for both database and software modeling. UML attempts to combine the best of the best from: Data Modeling concepts (Entity Relationship Diagrams), Business Modeling (work flow), Object Modeling and Component Modeling. UML is defined as: "a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software intensive system". Software architecture is an area of software engineering directed at developing large, complex applications in a manner that reduces development costs, increases the quality and facilitates evolution (Shaw and Garlan, 1996). A central and critical problem software architect's face is how to efficiently design and analyze software architecture to meet non-functional requirements. UML offers vocabulary and rules for communication and focus on conceptual and physical representations of a system. The various structural things in UML are Class, Interface, Collaboration, Use-case, behavioral things comprise of Interaction, State machine, grouping things comprising of packages and notes. The artifacts included in standard UML consist of: Use case diagram, Class diagram, Collaboration diagram, Sequence diagram, State diagram, Activity diagram, Component diagram and Deployment diagram (OMG, 1999). Unified Modeling Language (UML) is used as a specification technique for the system analysis and design process involved in the software development life cycle. All the modules of the information system have been developed using Visual Basic .Net with AutoCAD interface at the backend. The web components are hosted on Apache web server using Visual Web .Net Developer. Any type of application, running on any type and combination of hardware, operating system, programming language, and network can be modeled in UML. It's Profiles (that is, subsets of UML tailored for specific purposes) help to model Transactional, Real-time, and Fault-Tolerant systems in a natural way. UML is effective for modeling large, complex software systems. It is simple to learn for most developers, but provides advanced features for expert analysts, designers and architects. It can specify systems in an implementation-independent manner. Structural modeling specifies a skeleton that can be refined and extended with additional structure and behavior. Use case modeling specifies the functional requirements of system in an object-oriented manner. Existing source code can be analyzed and can be reverse-engineered into a set of UML diagrams. UML is currently used for applications other than drawing

designs in the fields of Forward engineering, Reverse engineering, Roundtrip engineering and Model-Driven Architecture (MDA). A number of tools on the market generate Test and Verification Suites from UML models. The implementation of UML for our case study (RCD Beam) is well elucidated under methodology.

2.3 Service Oriented Architecture

SOA represents a new paradigm that reflects a leap transition in both computing and software industries (Tsai et al., 2006). It has emerged after decades of using distributed computing technologies to add a new element to software stack. One of the main factors that led Microsoft to develop .Net Framework with great support to XML web services and WCF was to support SOA. IT executives believe that SOA will enable them to alleviate many of problems related to heterogeneity, interoperability and market ever-changing needs by allowing them to leverage existing IT investments in a more efficient form to be able to fulfil organizational goals effectively. SOA is a computing paradigm that utilizes services as fundamental elements for developing systems. The history of SOA goes back to a concept known as software-as-a-service (SaaS) which first appeared with Application Service Provider (ASP) software model (Geopfert and Whalen, 2002). Simply, ASP is a “third party entity that deploys, hosts and manages access to a packaged application and delivers software-based services and solutions to customers across WAN from a central data centre” (Geopfert and Whalen, 2002), (Krafzig et al., 2004). So, this ASP is responsible for managing, updating, maintaining, and supporting hosted applications as well as underlying infrastructures. These hosted applications are delivered over network on subscription or rental basis. Unfortunately, ASP model has suffered from several inherent limitations such as inability to provide complete customizable applications that resulted in a new generation of monolithic and tight-coupled architectures. These limitations have allowed the SOA paradigm to emerge to offer delivery of complex business process in form of network addressable components known as services that could be accessed and reused everywhere by everyone on condition that access permissions are granted to requestors. This model enabled end-to-end integration between different systems or even organizations with the ability to construct new applications and business processes on the fly to meet new and unexpected business needs. The benefits of SOA includes loose coupling (Loosely-coupled software means routines modules, programs are called by an application and executed as needed), Location transparency through the use of URL, More reusability for various consumers on the same application, higher productivity for project executors by leveraging legacy components wrapping them into a form that might be used by modern solutions, greater interoperability, higher agility by making systems easier to be built and modified, better alignment between IT and Business Execs because SOA defines both business requirements and software functions as services.

Francesco (2009) provides a formal account of the Service Oriented Computing (SOC) paradigm and its related technologies. His three main contributions are: The introduction of Calculus for Orchestration of Web Services (COWS), Development of methods and tools to analyse COWS terms, Descriptive power of COWS. His proposed future work was explored by providing a rigorous methodological foundation for specification and validation of SOC application using RC Design Business Application as a case study, service interoperability between RC Design Business and AutoCAD interface was ensured, the interaction of RC Beam Design services obeyed http protocol. UML4SOA was defined not to COWS but to RCD Beam Business activity. Security protocol was also ensured using WS-Security embedded in WCF

Maram (2008) explored Web service-oriented architecture for an adaptive e-learning framework known as WHURLE 2.0. The limited reusability and interoperability provided by the adaptive systems is a limitation to the work. Though, the thesis attempted to address these limitations at an architectural level. The research community agree that the next step of integrating Adaptive and Intelligent Web-based Educational Systems (AIES) will be achieved by defining learning frameworks or platforms that allow the creation of intelligent systems. Those platforms and frameworks should focus on being domain independent, extensible, interoperable and reusable technically and semantically.

Olaf (2009) created a decision-centric architecture design method for enterprise application development and integration projects employing SOA as their architectural style. The work was limited in provision of metamodel extensions, provision of more comprehensive tool support for the framework steps, integration with other methods and tools, and a broader use in professional services firms and software product documentation. A usability and user-experience improvement from user-centric point of view was also lacking.

Folorunso et al., 2010 described how SOA can support RTDBMS. The approach was highly theoretical; no actual implementation was carried out for a specific real-time database problem. Their approach was adopted to implement SOA for RCD. Serviceability Limit State for RCD was exposed as a service through RCD table advisor. A Service-Oriented Architecture for RC design is an information technology approach or strategy for RC in which

RC design application tool make use of (perhaps more accurately and in a synchronized manner) rely on data-based services available in a network such as the www. Implementing a Service-Oriented Architecture can involve developing applications like RC design that use services, making RC design table advisor application tools available as services so that other RC applications can use those services.

2.4 Cloud Computing

Cloud computing is an Internet-based computing environment, whereby shared resources tools related to applications and its information are provided to computers and other devices on demand for users. Cloud computing is a natural evolution of the widespread adoption of virtualization, Service-oriented architecture and utility computing. Details are abstracted from Designers, who engage the tools for their day today design through the web and the technology infrastructure "in the cloud", that supports them (Danielson, 2008). Consumption is usually billed on a utility (resources consumed, like electricity) or subscription (time-based, like a newspaper) basis with little or no upfront cost. Other benefits of this approach are low barriers to entry, shared infrastructure and costs, low management overhead, and immediate access to a broad range of applications. In general, users can terminate the contract at any time (thereby avoiding return on investment risk and uncertainty), and the services are often covered by service level agreements (SLAs) with financial penalties (Clint, 2008 and Frank, 2008). Sharing "perishable and intangible" computing power among multiple tenants can improve utilization rates, as servers are not unnecessarily left idle (which can reduce costs significantly while increasing the speed of application development). A side-effect of this approach is that overall computer usage rises dramatically, as customers do not have to engineer for peak load limits (Carey, 2008). In addition, "increased high-speed bandwidth" makes it possible to receive the same. The cloud is becoming increasingly associated with Small and Medium Enterprises (SMEs) as in many cases they cannot justify or afford the large capital expenditure of traditional IT. SMEs also typically have less existing infrastructure, less bureaucracy, more flexibility, and smaller capital budgets for purchasing in-house technology. Similarly, SMEs in emerging markets are typically unburdened by established legacy infrastructures, thus reducing the complexity of deploying cloud solutions. According to Nicholas Carr, the strategic importance of information technology is diminishing as it becomes standardized and less expensive. He argues that the cloud computing paradigm shift is similar to the displacement of frozen water trade by electricity generators early in the 20th century (Nicholas, 2008). Although firms might be able to save on upfront capital expenditures, they might not save much and might actually pay more for operating expenses. In situations where the capital expense would be relatively small, or where the organization has more flexibility in their capital budget than their operating budget, the cloud model might not make great fiscal sense. Other factors having an impact on the scale of potential cost savings include the efficiency of a company's data center as compared to the cloud vendor's, the company's existing operating costs, the level of adoption of cloud computing, and the type of functionality being hosted in the cloud (Paul, 2010 and Balakrishna, 2009). The underlying concept of cloud computing dates back to the 1960s when John McCarthy opined that "computation may someday be organized as a public utility", almost all the modern-day characteristics of cloud computing (elastic provision, provided as a utility, online, illusion of infinite supply), the comparison to the electricity industry and the use of public, private, government and community forms was thoroughly explored in Douglas Parkhill's 1966 book, *The Challenge of the Computer Utility*. The actual term "cloud" borrows from telephony in that telecommunications companies, who until the 1990s primarily offered dedicated point-to-point data circuits, began offering Virtual Private Network (VPN) services with comparable quality of service but at a much lower cost. By switching traffic to balance utilization as they saw fit, they were able to utilize their overall network bandwidth more effectively. The cloud symbol was used to denote the demarcation point between that which was the responsibility of the provider from that of the user. Cloud computing extends this boundary to cover servers as well as the network infrastructure (Mark, 1993). The first scholarly use of the term "cloud computing" was in a 1997 lecture by Ramnath Chellappa. Amazon played a key role in the development of cloud computing by modernizing their data centers (Jeff, 2006 and Carl, 2010). In 2007, Google, IBM and a number of universities embarked on a large scale cloud computing research project (Google and IBM, 2008). In early 2008, Eucalyptus became the first open source AWS API compatible platform for deploying private clouds. By mid-2008, Gartner saw an opportunity for cloud computing "to shape the relationship among consumers of IT services, those who use IT services and those who sell them" (Amy, 2008) and observed that "organizations are switching from company-owned hardware and software assets to per-use service-based models" so that the "projected shift to cloud computing ... will result in dramatic growth in IT products in some areas and significant reductions in other areas." (Gartner, 2008) In March 2010, Microsoft's CEO, Steve Ballmer, made his strongest statement of betting the company's future in the cloud by proclaiming, "For the cloud, we're all in" and further stating, "About 75 percent of our folks are doing entirely cloud based or entirely cloud inspired; a year from now that will be 90 percent." (Steve, 2010) Microsoft has also offered details on cloud services for government agencies (Microsoft, 2010).

The advantages derived from cloud computing is many; they include:

- Designers would be able to access their applications and data from anywhere at any time. They could access the cloud computing system using any computer linked to the Internet. Data wouldn't be confined to a hard drive on one user's computer or even a corporation's internal network.
- It could bring hardware costs down. Cloud computing systems would reduce the need for advanced hardware on the client side. Clients wouldn't need to buy the fastest computer with the most memory, because the cloud system would take care of those needs for them. Instead, inexpensive computer terminal could be bought. The terminal could include a monitor, input devices like a keyboard and mouse and just enough processing power to run the middleware necessary to connect to the cloud system. Users wouldn't need a large hard drive because all information will be stored on a remote computer.
- Firms that rely on computers have to make sure they have the right software in place to achieve goals. Cloud computing systems will give these organizations company-wide access to computer applications. The consulting firms don't have to buy a set of software or software licenses for every employee. Instead, the company could pay a metered fee to a cloud computing company.
- Servers and digital storage devices take up space. Some companies rent physical space to store servers and databases because they don't have it available on site. Cloud computing will gives these companies the option of storing data on someone else's hardware, removing the need for physical space on the front end.
- Corporations might save money on IT support. Streamlined hardware would, in theory, have fewer problems than a network of heterogeneous machines and operating systems.
- If the cloud computing system's back end is a grid computing system, then the client could take advantage of the entire network's processing power. Often, engineers and scientists and researchers work with calculations so complex that it would take years for individual computers to complete them. On a grid computing system, the client could send the calculation to the cloud for processing. The cloud system would tap into the processing power of all available computers on the back end, significantly speeding up the calculation.

While the benefits of cloud computing seems convincing, there are many potential problems:

There are a few standard hacker tricks that could give cloud computing company major headaches. One of those is called key logging. A key logging program records keystrokes. If a hacker manages successfully to load a key logging program on a victim's computer, he or she can study the keystrokes to discover user names and passwords. Of course, if the user's computer is just a streamlined terminal, it might be impossible to install the program in the first place. Perhaps the biggest concerns about cloud computing are security and privacy. The idea of handing over important data to another company worries some people. Corporate executives might hesitate to take advantage of a cloud computing system because they can't keep their company's information under lock and key. The counterargument to this position is that the companies offering cloud computing services live and die by their reputations. It benefits these companies to have reliable security measures in place. Otherwise, the service would lose all its clients. It's in their interest to employ the most advanced techniques to protect their clients' data. Privacy is another matter. If a client can log in from any location to access data and applications, it's possible the client's privacy could be compromised. Cloud computing companies will need to find ways to protect client privacy. One way is to use authentication techniques such as user names and passwords. Another is to employ an authorization format; each user can access only the data and applications relevant to his or her job. Some questions regarding cloud computing are more philosophical. Does the user or company subscribing to the cloud computing service own the data? Does the cloud computing system, which provides the actual storage space, own it? Is it possible for a cloud computing company to deny a client access to that client's data? Several companies, law firms and universities are debating these and other questions about the nature of cloud computing. How will cloud computing affect other industries? There's a growing concern in the IT industry about how cloud computing could impact the business of computer maintenance and repair. If companies switch to using streamlined computer systems, they'll have fewer IT needs. Some industry experts believe that the need for IT jobs will migrate to the back end of the cloud computing system. Another area of research in the computer science community is autonomic computing. An autonomic computing system is self-managing, which means the system monitors itself and takes measures to prevent or repair problems. Currently, autonomic computing is mostly theoretical. But, if autonomic computing becomes a reality, it could eliminate the need for many IT maintenance jobs. Cloud computing could turn home computers into simple terminal interfaces. In some ways, this is a step backward. Early computers included hardwired user terminals. Each terminal had a computer monitor and keyboard, but they only served as an interface

to the main computer. There was no way to store information locally on a terminal. Cloud computing is the first and the precise step in making computer ubiquitous.

2.5 Visualization

Information Visualization is a process of transforming information into a visual form enabling the viewer to observe, browse, make sense, and understand the information. It typically employs computers to process the information and computer screens to view it using methods of interactive graphics, imaging, and visual design. It relies on the visual system to perceive and process the information. Information Visualization serves as external cognition aids that assist people's memory and thinking and combine illustrative representations of data with interactive user interfaces which allows people to examine data from many different perspectives (Stuart, 1999). Guidelines for designing information visualizations are available from writers such as Few (Few, 2006, Few, 2009) and Tufte (Tufte, 1983, Tufte, 1990). Some of these guidelines overlap with guidelines from graphic design, including the need to present information clearly, precisely, and without extraneous or distracting clutter. Good visualizations use graphics to organize information, highlight important information, allow for visual comparisons, and reveal patterns, trends, and outliers in the data. Visualization guidelines are also derived from principles of human perception, and urge the designer to be aware of the perceptual properties which can affect the design. Few, (2006) provide a good overview of these principles. The process of data visualization includes four basic stages combined in a number of feedback loops; the collection and storage of data, the pre-processing designed to transform the data into something we can understand, the display hardware and the graphics algorithms that produce an image on the screen and the human perceptual and cognitive system (the perceiver). Visualization techniques can be classified based on; The task at hand, the structure of the underlying data set, the dimension of the display, whether the focus is geometric or symbolic, whether the stimulus is 2D or 3D, or whether the display is static or dynamic. Visual representations help for various functions such as; to address emotions, to illustrate relations, to discover trends, patterns, or outliers, to get and keep the attention of recipients, to support remembrance and recall, to present both overview and detail, to facilitate learning, to coordinate individuals, to motivate people and to establish a mutual story, to energize people and initiate actions, to present data in various forms with differing interactions, to provide a qualitative overview of large and complex data sets, summarize data, and identify regions of interest and appropriate parameters for more focused quantitative analysis, and to harnesses the perceptual capabilities of the human visual system, etc.

3. Methodology

New things are intimidating and are causing resistance (Jager & Lokman, 1999). Creating service oriented architecture takes thought, patience, planning, and time. It is a journey, and depending on the size and scope of components, it may be a journey of years or even a decade. However, an Intelligent Visualization framework using service-oriented architecture can be developed to alleviate these problems. For the analysis and visualization system, we propose four main layers which can be easily distributed over the network (Figure 1). The layers considered are: Data acquisition, Management/Orchestration which will host the mathematical analysis, feature extraction and decision making, the Visualization layer, and the Communication layer. These Layers will be interconnected using secure web services which enables the distribution of all layers to different parts of the automation services support systems. Visualization Layer Service Composed of modules that offer the end visualization outcome, which depends on performance / quality of detail, required to visualize the same data provided by the next layer. This layer includes:

- *Web Client Services*: These are third party software that range from the common Internet browser to more sophisticated readers of web 2.0 content like mobile phones (e.g., Microsoft™ Virtual Earth, Google™ Maps mobile API).
- *Virtual Collaborative Environment Services*: Game Engine with real-time, immersive environment for virtual collaboration
- *High Definition Rendering Services*: Third party tools for particular visualization needs (e.g., AutoCAD 2010, Visual Nature Studio™ 3, 3D Nature, 2008).

Management / Orchestration layer services architecture depends on process services that link and sequence services according to existing and potentially new visualization requirements. These automated services further delegate specialized functions such as management, security, batch processing and similar features. This layer includes:

- *Workflow Manager*: Responsible for managing sequence of operations/processes to achieve a specific organizational goal (like rendering a sequence of images), orchestrating the interaction of both human and machine actors that may intervene in the process.
- *RTI Manager*: Responsible for managing the Real Time Infrastructure that implements the rules. It enforces the standards that any engaging simulation should adhere to. Consequently, it will coordinate data feed/exchange and operations between simulations federates running on the framework's execution platform.
- *Render Manager*: Responsible for scheduling batch or simultaneous rendering tasks.
- *Grid Middleware Manager*: Responsible for enabling grid technology, thus sharing resources across multiple machines, while masking this implementation to the other layers which only interact with a single "virtual" entity.

Data Layer Service Data sources can be composited to feed spatial and non-spatial information requirements that the orchestration layer needs to fulfil its lifecycle, thus abstracting the need for a particular data source, whether this source is a AutoCAD Database, a RSS live feed or other machine available sources such as anonymous ftp repositories.

Communication Service Layer encapsulate information using Web Services protocols (Web Service Description Language - WSDL, Simple Object Access Protocol -SOAP, and Universal Description Discovery and Integration - UDDI), data is transferred from all layers through Wrappers/Interfaces that are implemented by standard contracts on each module.

The analysis, feature extraction and decision making was built using Visual Basic .NET studio which simplifies system development process. The visualization on the personal computer (PC), mobile device, etc., was implemented keeping in mind the portability requirements and networks. The visualization framework support global condition monitoring services and is designed to ease the work of the decreasing number of system experts on the move. The implementation of the visualization system will require good commitment on behalf of the automation process personnel and/or expert organizations. The knowledge needed for building the reasoning layer cannot be done in any other way than in close contact with the automation experts.

We implement UML for RCD beam using yEd Graph Editor Tool (Figure 2a and 2b). The tool is a freeware downloadable at <http://www.yWorks.com>. The UML for RCD shows different types of beams for analysis which inherits their properties from RCD properties interface. There are eight loads possibilities on each beam span which can be combined in sixty-four ways on a single span. The RCD properties interface and Load Beam interface worked together as a dialogue boxes, they are always activated simultaneously to get user's input. The manoeuvring for drawing the bending moment, shear diagram and detailing of the beam using AutoCAD (2010) Active X control is done through the RCD Engine class which is revealed in AutoCAD Interface. The RCD Engine class also activate RCD Table advisor for design and detailing purpose.

All continuous beams implements infinite number of span, the memory of the host machine on which the tool will be put into operation will be our limitation to the quantity of span that can be implemented. Eight loads conditions considered are Point load, Distributed linear load, Couple load, equilateral-triangular load, right-angled-triangular load. The loads are on a particular location along the span or spread entirely over the span. The load beam interface and RCD properties interface are link into RCD Engine class which contains objects that facilitate the analysis, design and visualization of our RCD. For example, the `getBeamData()` object collects inputs from the RCD properties and Load Beam dialog boxes for further analysis, it also ensure that all data collected have valid inputs. The `showDialog()` object for steel table also gets data from RCD table advisor which is exposed as a service. The `designMainReinforcement()` object communicates with AutoCAD interface for the visualization of the output data.

4. Implementation

Encapsulating RCD Beam business process information using Web-Services protocols was achieved with Window Communication Foundation (WCF) embedded in Microsoft Visual Web Developer Express 2010; WCF is the integration of Web Service Description Language (WSDL), Simple Object Protocol (SOAP) and Universal Description Discovery Integration (UDDI). Data is transferred from all layers through Wrappers/Interfaces that are implemented by standard contracts on each module. As a proof-of-concept in leveraging our architecture in Figure 1 with reality, we mapped and integrated our platform of high definition rendering using the commercial software AutoCAD with the Visual Basic Express 2010, Visual Web developer 2010 and other tools. A typical flow of information through this architecture is shown in Figure 3.

Visualization RCD Service Oriented Architecture was implemented using Visual Studio .Net Express 2010 Edition on Window Vista Home Premium 64-bit service pack 2 Operating System with Intel core duo CPU at 2.00 GHz, 4 GB memory. Microsoft Visual Basic Express 2010 was used to implement the core program while Window Communication Foundation (WCF) found in Microsoft Visual Web Developer Express 2010 was used to execute the services. The service is thereafter exposed using the http protocol. AutoCAD 2010 was borrowed and enhanced through interoperability as a visualization environment for the RCD Beam; RCD Beam for AutoCAD form was created and activated simultaneously with the AutoCAD environment. The form hosts the File, View, Beam, Option and Help menus. The number of beam span to be designed is entered in the black box placed after the help menu; its default value is 1 for simply supported beam. When the Beam menu is clicked; it reveal all the known options for different types of beams encountered by RC designers as earlier planned in Figure 2 under methodology. When the mouse is hovered on each item on the list, corresponding picture illustrating what the selected option meant is visualized in the anticipated monitor at the extreme right hand side as shown in Figure 4. When any of the list options for different types of beams is clicked, two forms are loaded simultaneously. They are called “Load Beam by Checking Applicable Load Type” form and “RCD Properties” form. The first form hosts all different types of loads (Figure 5). For example when “point or Knife Edge Load is checked along with “Linearly Distributed Load Over The Entire Span” are checked, two new forms will be revealed, waiting for the user input for point/knife edge load in the first form and Linearly distributed load in the second form (Figure 6). In case the user decides to consider both live and dead load separately, the check box for “Consider Dead and Live Load” is checked, the form automatically expand waiting for the user’s input for dead and live load. The value for ultimate load (P(kN) and/or w(kN/m) automatically change as the user enter valid numeric values in the textbox for Dead and Live load. “RCD Properties” form host all the RCD beam properties needed to achieve a novel design for reinforced concrete beam (Figure 7).

Validation of basic input is ensured within the code to minimize run time error and other errors. Users are given choice of values through the combo box and list box; users are also free to input valid values of their choice. When the completion of the input is successfully achieved, “Draw Bending Moment for Beam” sub menu is triggered from the option menu. The Bending Moment generated is revealed with text labels at the critical areas within the AutoCAD environment (Figure 8). The visualization pan and zooming tool in AutoCAD environment is employed to focus and view details especially the negative moments, positive moments and text details. To design for the steel reinforcement, “Design Reinforcement for Beam” sub menu is triggered from the option menu which invokes the steel table dialog box. The Value for the bending moment, Area of steel calculated, k-value, Lever arm are automatically revealed to guide the user in using his intuition and judgement to pick the right bar size from the steel table. The dynamic query approach we have adopted empowers users to perform far more complex searches by using visual search strategies. The enthusiasm RCD analysts have for dynamic queries emanates from the sense of control they gain. They quickly perceive patterns in data, fly through data by clicking buttons and rapidly generate new queries based on what they discover through incidental learning. Interactive visRCD Table visualization is a process made up of a number of interlocking feedback loops that fall into three broad classes. At the lowest level is the data manipulation loop, through which objects are selected and moved using the basic skills of eye-hand coordination. At an intermediate level is an exploration and navigation loop, through which an analyst finds his or her way in a large visual RCD data space. But exploration of RCD data process can be generalized to more abstract searching operations. The visRCD Steel Table dialog box (Figure 9) is a web service which implements services for RCD Beam using Windows Communication Foundation (WCF). WCF is Microsoft’s unified programming model for building service-oriented applications. It enables developers to build secure, reliable, transacted solutions that interoperate with applications in different platforms. With WCF, the client will always communicate with the proxy only. The RCD client will not be allowed to directly communicate with the services, even though the service communicates with the proxy; proxy forwards the call to the service. Proxy exposes the same functionalities as Service exposed. The limitation of the implementation is that the service must be available; otherwise the proxy will not be able to access the server, error that crashes the system will be generated. The error will be cached with message alerting the user that the service is not available. When ‘Design for Main Bar’ sub menu is activated from option menu, the visRCD Table Advisor is triggered as a dialog box showing the Design Moment, Area of Steel Calculated, k_value and lever arm, z value. This guides the user to make the best of choice decision when picking reinforcement from the steel table. For example, when the button with 943 is clicked, area of Steel provided immediately appears in the text box as 943 while text box in front of Number of steel label reveal 3 and text box in front of Bar size provided label divulge 20. The Black Monitor at the extreme right hand side of the visRCD Table Advisor Interface quickly alerts the user that is on the right path since Area provided is greater than Area calculated. The alert is revealed in blue colour otherwise, it will be revealed in Red colour. Without leaving the interface and provided

the service is available (Figure 8), we can check for Minimum and Maximum Area of Steel and deflection by clicking the appropriate button, and then the revelation is monitored in the black box. Once this is okay, 'Accept' button is triggered while the data is transferred back to the main program for further use. Thereafter, the interface becomes invisible.

The interaction diagrams were presented to usability RCD experts to be initially evaluated and commented upon. The design decisions were implemented using Microsoft Visual Studio 2010 and AutoCAD 2010. A prototype was produced to test the theories in practice and to gain information about the RCD users. The implementation itself took advantage of emerging technologies such as WCF which make it possible to use Service Oriented Applications through the web services. A total of twenty-five RCD experts were recruited to perform various tasks on both Structural Analysis and Design (STAAD) Software Product and the prototype RCD applications and rate their experiences. The outcomes of the user-study were positive, with the majority of the users (twenty-two out of twenty-five) preferring the prototyped interface to the existing and fully working solution. This was considered a good result given the qualitative feedback from the users. Some of the findings emphasized the importance of the holistic experience and look-interact-and-feel over a pure set of technical features and merits.

5. Conclusion

The major contribution was the integration of a host of techniques to create a novel application that is both usable and useful in any domain using SOA approach. Cloud Computing as a new innovation to computer business technology was enhanced for RCD business activities, Civil/Structural engineering Consulting firms do not need to spend too much to acquire stand alone software that get obsolete in less than a year to do their job, they can pay for the service on the internet to get the latest innovative software at the lowest cost possible. We simulate the old ways RC designer works to persuade them into accepting the new technology which eventually reduce cost to the client. More people will now patronize expert rather than quacks for structural design of their building structures. Safety of the structures will be ensured. Incessant collapse of buildings will drastically reduce. The visualization techniques described in this paper really helps the design of a Service Oriented Process when using a compositional approach. Considering compositions as first class entities, we provide to designer a framework able to support their design process. Based on these techniques, we identify chronic fragment patterns and sketch a categorization of these entities. Moreover, when analyzing the composed result, we identify critical points where process extensions interact violently with initial behavior.

In this paper, we deliberately focus our visualization work on composition definitions. Our goal is to understand" a system designed by composition, and supports the designer during the design process. In an SOA realized orchestrations of Web Services, partnerships between services and processes is a key point for performance measurement. Many experts in government and commerce still consider the greatest barrier to adoption of cloud as; information security, availability of service and privacy. While these risks exist across the entire cloud ecosystem, every cloud customer retains responsibility for assessing and understanding the value and sensitivity of the data they may choose to move to the cloud. As the owners of that information, cloud customers also remain accountable for decisions regarding the protection of that data wherever it may be stored. The platform services and hosted applications must be secure and available. The cloud is a dynamic hosting environment in which technologies and business models continue to evolve. This continuous change is a security challenge that cloud providers must address through an effective and dynamic security program. Sophisticated malicious attempts aimed at obtaining identities or blocking access to sensitive business data is a threat that undermines the willingness of organizations to adopt cloud services. Cloud providers must prove that they have put into place and constantly evaluate the effectiveness of the technologies, controls, and processes used to mitigate such disruptions. In addition to these challenges, cloud providers must also address the myriad requirements related to delivering services globally online including those coming from governments, legal rulings, and industry standards. In short, cloud service providers need to manage information security risks in a way that engenders trust with their customers—the government organizations or businesses that do provide such services to end users, as well as directly with end users. Cloud customers, having decided to transfer some risk to a cloud provider by consuming a cloud service, should understand what their cloud provider has done and is doing to protect their information. A successful information security program should also incorporate risk-based decision-making processes into day-to-day business activities, integrates information security into core IT and business practices, ensures adequate resource allocation for the projects and programs designed to reduce risk and dedicates resources to focus on key elements of the information security program.

References

- Amy Schurr. (2008). Keep an eye on cloud computing, Amy Schurr, Network World, 2008-07-08, citing the Gartner report, Cloud Computing Confusion Leads to Opportunity. [Online] Available: <http://www.networkworld.com/newsletters/itlead/2008/070708itlead1.html>. Retrieved 2009-09-11.
- Balakrishna Narasimhan. (2009). Cloud Computing Savings – Real or Imaginary?. Appirio.com. 2009-04-16. [Online] Available: <http://www.appirio.com/blog/2009/04/cloud-computing-savings-real-or.php>. Retrieved 2010-08-22.
- Bass, L., Clements, P., and Kazman, R. (1998). *Software Architecture in Practice*. Reading, MA: Addison-Wesley Longman
- Buschmann F., Jakes C., Meunier R., Rohnert H., Stahl M. (1996). Pattern-Oriented Software Architecture – A System of Patterns, John Wiley & Sons. [Online] Available: <http://www.citeseer.nj.nec.com/context/14159/o>
- Buyya, Rajkumar; Chee Shin Yeo, Srikumar Venugopal. (2008). *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*. Department of Computer Science and Software Engineering, University of Melbourne, Australia. pp. 9. [Online] Available: http://www.gridBus.org/~raj/papers/hpcc2008_keynote_cloudcomputing.pdf. Retrieved 2008-07-31.
- Carey, P.W. (2008). Cloud Computing: The Evolution of Software-as-a-Service. [Online] Available: <http://knowledge.wpcarey.asu.edu/article.cfm?articleid=1614>. Retrieved 2010-08-22.
- Carl Brooks (2010). Amazon's early efforts at cloud computing? Partly accidental [Online] Available: <http://itknowledgeexchange.techtarget.com/cloud-computing/2010/06/17/amazons-early-efforts-at-cloud-computing-partly-accidental/>
- Clint Boulton. (2008). <http://www.eweek.com/c/a/Enterprise-Applications/Forresters-Advice-to-CFOs-Embrace-Cloud-Computing-to-Cut-Costs/>
- Danielson, Krissi (2008). Distinguishing Cloud Computing from Utility Computing. Ebizq.net. [Online] Available: http://www.ebizq.net/blogs/saasweek/2008/03/distinguishing_cloud_computing/. Retrieved 2010-08-22.
- Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly.
- Few, S. (2009). *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.
- Folorunso Olusegun, Yusuf Lateef O., and Okesola Julius O. (2010): "SOA-RTDBS: A Service Oriented Architecture (SOA) Supporting Real Time Database System". *Oriental Journal of Computer Science & Technology*; Vol. 3(1), 171-184 (2010)
- Francesco Tiezzi. (2009). Specification and Analysis of Service-Oriented Applications. PhD. Thesis in Computers Science. Dipartimento di Sistemi E Informatica Università Degli Studi Di Firenze.
- Frank Dzubeck. (2008). Five cloud computing questions. Networkworld.com. [Online] Available: <http://www.networkworld.com/columnists/2008/080508-dzubeck.html>. Retrieved 2010-08-22.
- Garlan David and Perry Dewayne. (1995). Introduction to the Special Issue on Software Architecture. *IEEE Transactions on Software Engineering* 21, 4 (April 1995).
- Garlan, D. and Shaw, M. (1993). An Introduction to Software Architecture, in *Advances in Software Engineering and Knowledge Engineering*, vol. I. River Edge, NJ: World Scientific Publishing Company, 1993.
- Gartner. (2008). Gartner Says Worldwide IT Spending On Pace to Surpass \$3.4 Trillion in 2008, Gartner, 2008-08-18. Retrieved 2009-09-11.
- Gartner. (2010). Gartner Say's Cloud Computing Will Be As Influential As E-business. Gartner.com. [Online] Available: <http://www.gartner.com/it/page.jsp?id=707508>. Retrieved 2010-08-22.
- Geopfert, J. And Whalen, M. (2002). An Evolutionary View of Software as a Service. IDC white paper; [Online] Available: www.idc.com
- Google and I.B.M. Join. (2008). in 'Cloud Computing' Research [Online] Available: http://www.nytimes.com/2007/10/08/technology/08cloud.html?_r=3&ex=1349496000&en=92627f0f65ea0d75&ei=5090&partner=rssuserland&emc=rss&oref=slogin
- Gruman, Galen. (2008). [Online] Available: What cloud computing really means. *InfoWorld*. [Online] Available: <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>. Retrieved 2009-06-02.

- Jager, A. K. and Lokman A. H. (1999). Impacts of ICT in education. The role of the teacher and teacher training; Paper presented at the European Conference on Education Research, Lathi, Finland 22-25 September 1999. [Online] Available: <http://www.leeds.ac.uk/educol/documents/00001201.htm>
- Jeff Bezos. (2006). [Online] Available: http://www.businessweek.com/magazine/content/06_46/b4009001.htm
- Kanwalvir S.D., and Himanshu A. (2011): Modeling and Designing Land Record Information System Using Unified Modeling Language. *International Journal of Advanced Computer Science and Applications (IJACSA)*. Vol. 2, No. 2, February 2011.
- Kobryn, C. (1999). UML 2001: a standardization odyssey', *Comm. of the ACM*, Vol. 42 No. 10, October, pp. 29-37, 1999.
- Krafzig, Dirk, Banke, Karl, Slama and Durk. (2004). *Enterprise SOA, Service Oriented Architecture Best practices*. Prentice Hall, November 2004. ISBN 0-13-146575-9.
- MacKenzie, M., Laskey, K., McCabe, F., Brown, P., Metz, R. (2006). Reference Model for Service Oriented Architecture 1.0. Technical Report wd-soa-rm-cd1, OASIS (February 2006)
- Maram Meccawy. (2008). A Service-Oriented Architecture for Adaptive and Collaborative E-Learning Systems. PhD Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy.
- Mark Laubach 1993). July, 1993 meeting report from the IP over ATM working group of the IETF. [Online] Available: <http://mirror.switch.ch/ftp/doc/ietf/ipatm/atm-minutes-93jul.txt>. Retrieved 2010-08-22.
- Microsoft (2010). Government Cloud Computing: Improved Economy and Services in the Cloud. [Online] Available: http://www.microsoft.com/industry/government/guides/cloud_computing/default.aspx retrieved 26-12-2010
- Nicholas G. Carr. (2008). Nicholas Carr on 'The Big Switch' to cloud computing. *Computerworlduk.com*. [Online] Available: <http://www.computerworlduk.com/technology/internet/applications/instant-expert/index.cfm?articleid=1610>. Retrieved 2010-08-22.
- Olaf Zimmermann. (2009). An Architectural Decision Modelling Framework for Service-Oriented Architecture Design. PHD Thesis at the Institut für Architektur von Anwendungssystemen der Universität Stuttgart.
- OMG. (1999). UML Revision Task Force, OMG-Unified Modeling Language Specification, [Online] Available: <http://uml.systemhouse.mci.com/> accessed March 10th 2011.
- Paul, Fredric. (2010). 1 Midsize Organization Busts 5 Cloud Computing Myths. *Bmighty.com*. [Online] Available: <http://www.bmighty.com/services/showArticle.jhtml?articleID=211600030>. Retrieved 2010-08-22.
- Peltz, C. (2003). Web Services Orchestration and Choreography. *Computer* 36(10)
- Perry, Dewayne E. and Wolf, Alexander L. (1992). Foundations for the Study of Software Architecture. *ACM SIGSOFT Software Engineering Notes*, 17:4 (October 1992).
- Shaw, M., and Garlan, D. (1996). *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.
- Steve Ballmer (2010). speech at UW: "We're all in" for cloud computing, [Online] Available: http://seattletimes.nwsources.com/html/microsoftpri0/2011255515_steve_ballmer_speech_at_uw_were_all_in_for_cloud_c.html retrieved 26-12-2010
- Stuart, Card. (1999). Information visualization. In Stuart Card, Jock Mackinlay, & Ben Shneiderman (Eds.), *Readings in information visualization: Using vision to think* (pp. 1–34). San Francisco, CA: Morgan Kaufmann.
- Tsai, W. T., Malek, M., Chen, Y. and Bastani F. (2006). Perspectives on Service-Oriented Computing and Service-Oriented System Engineering. Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering (SOSE'06)
- Tufte Edward. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Chelshire, CT.
- Tufte, E. R. (1990). *Envisioning information*. Graphics Press Cheshire, Conn. (PO Box 430, Cheshire 06410).

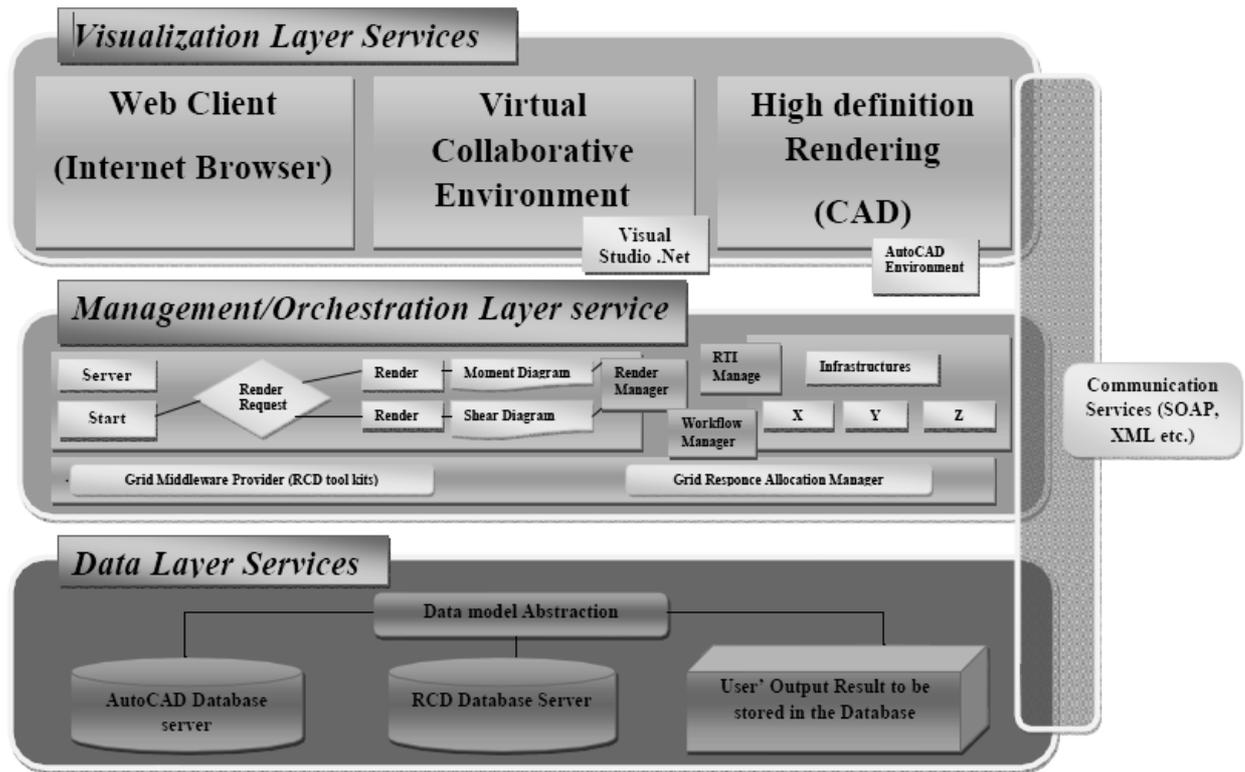


Figure 1. Architecture diagram with a modular SOA platform built upon aggregation of modules according to the services offered.

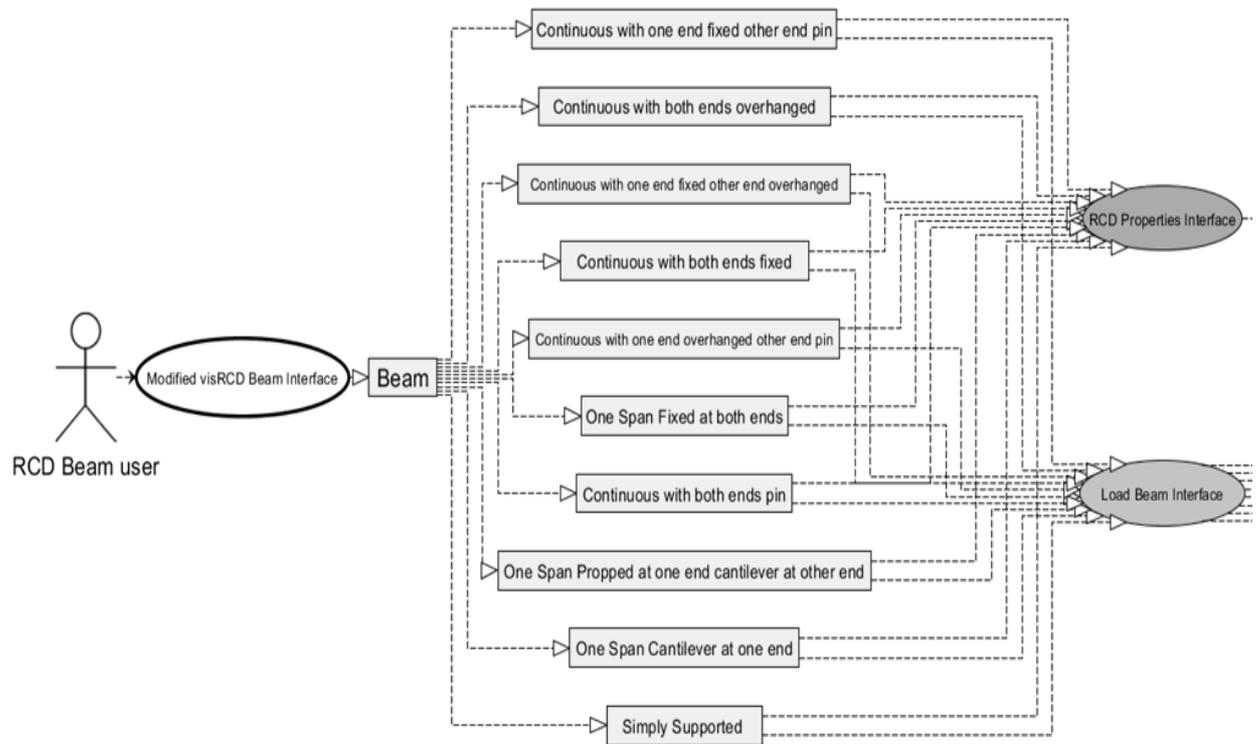


Figure 2a. UML Diagram for RCD Beam Interface showing RCD Table Advisor (continued at Figure 2b)

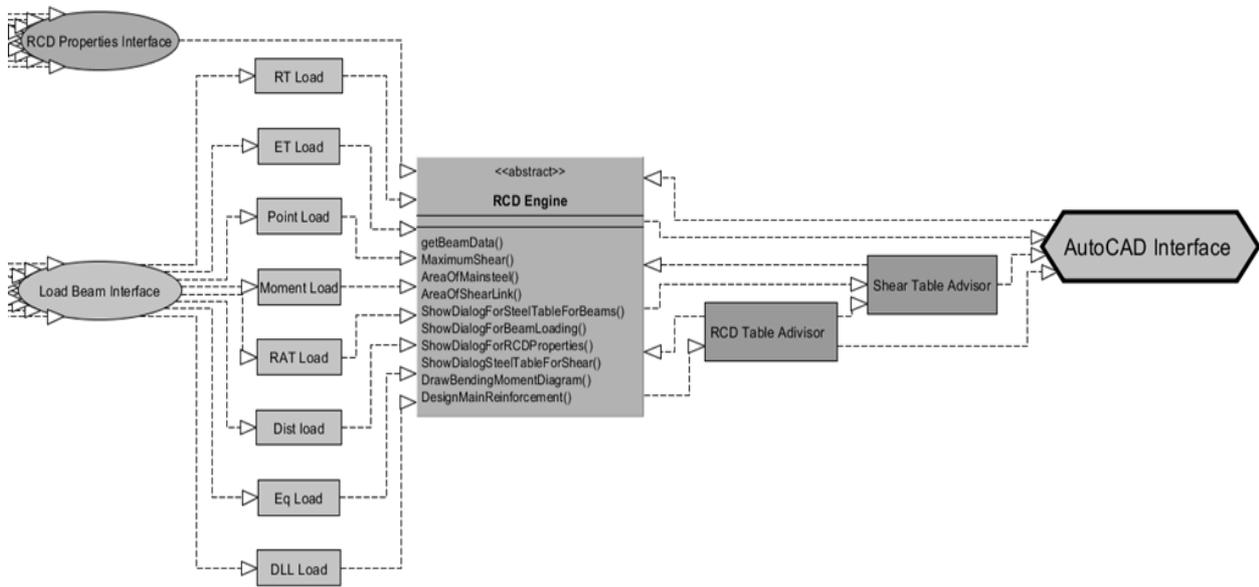


Figure 2b. UML Diagram for RCD Beam Interface showing RCD Table Advisor

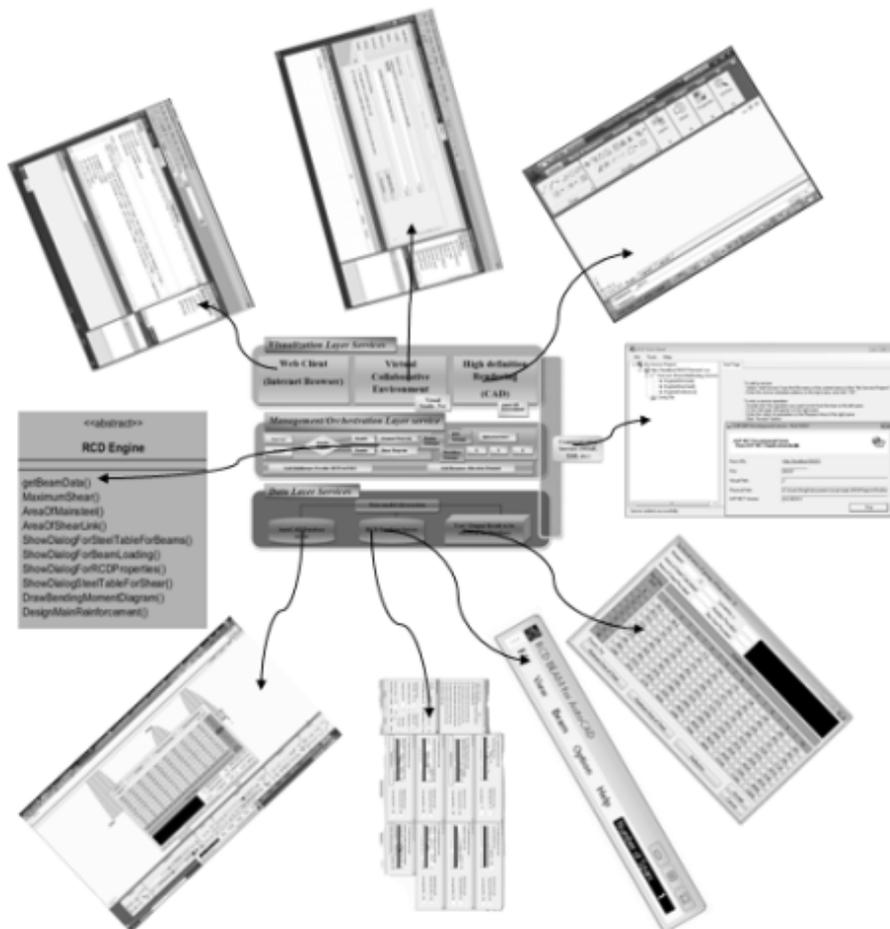


Figure 3. Framework's workflow for RCD Beam Business Process Design

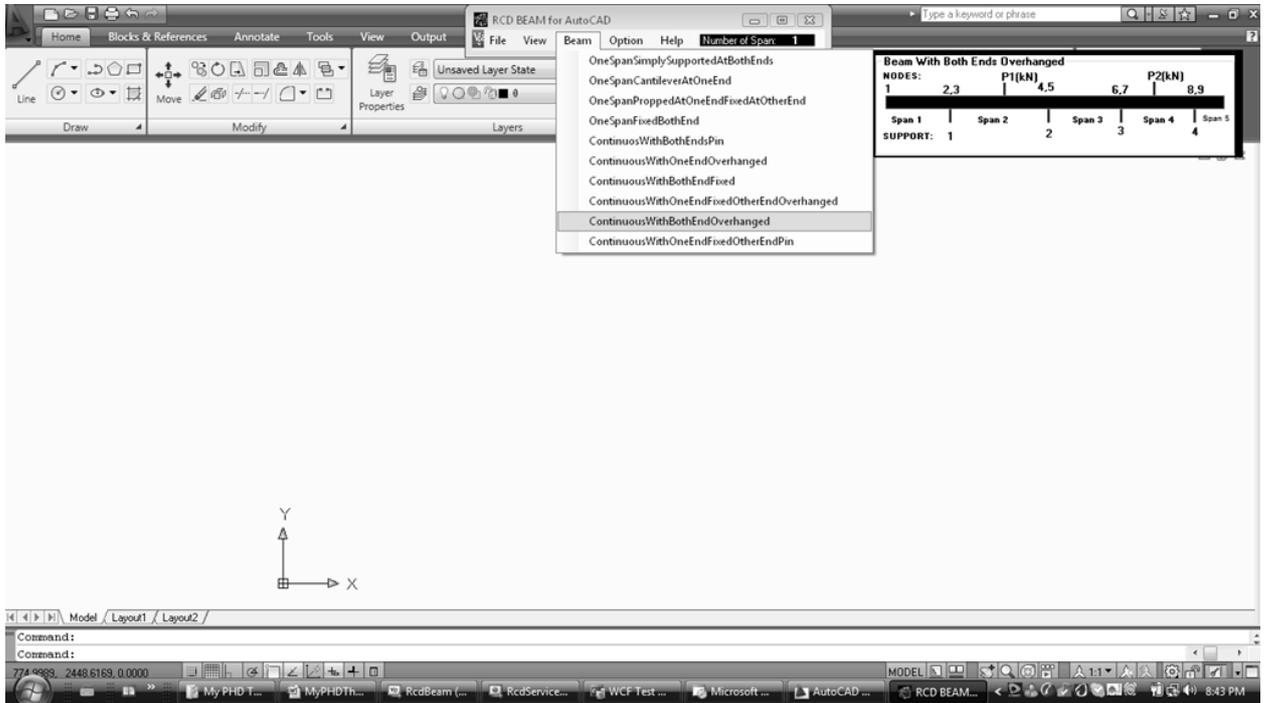


Figure 4. Animated List Control showing different types of beams with corresponding movie on the Right Hand Side



Figure 5. Link to load beams

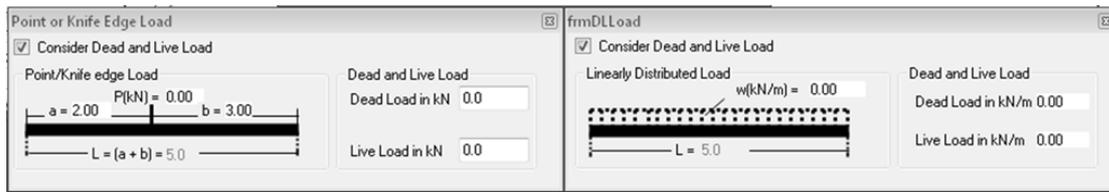


Figure 6. Knife Edge load and Linearly Distributed load

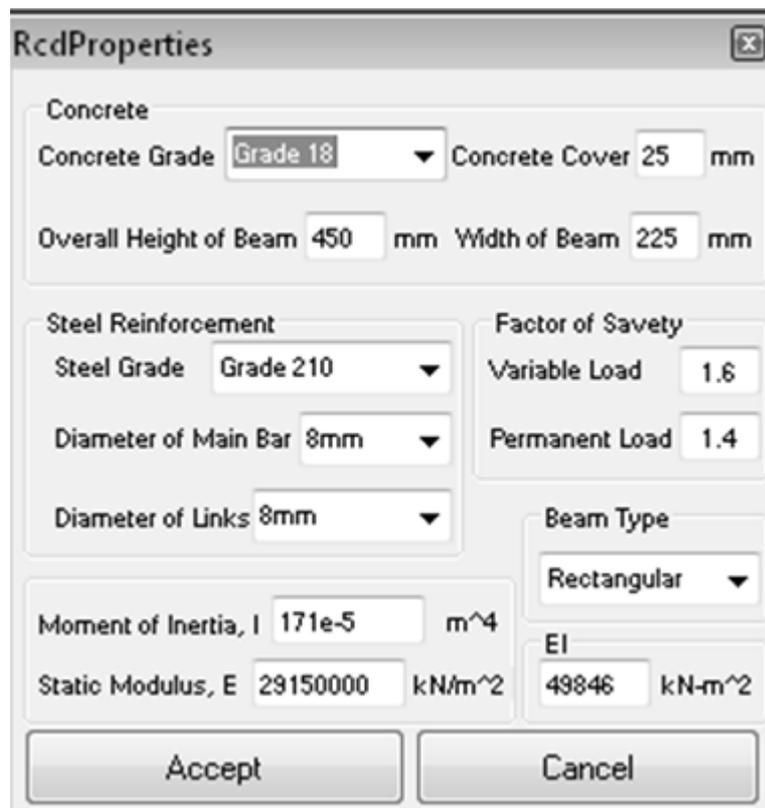


Figure 7. RCD Beam properties for Design

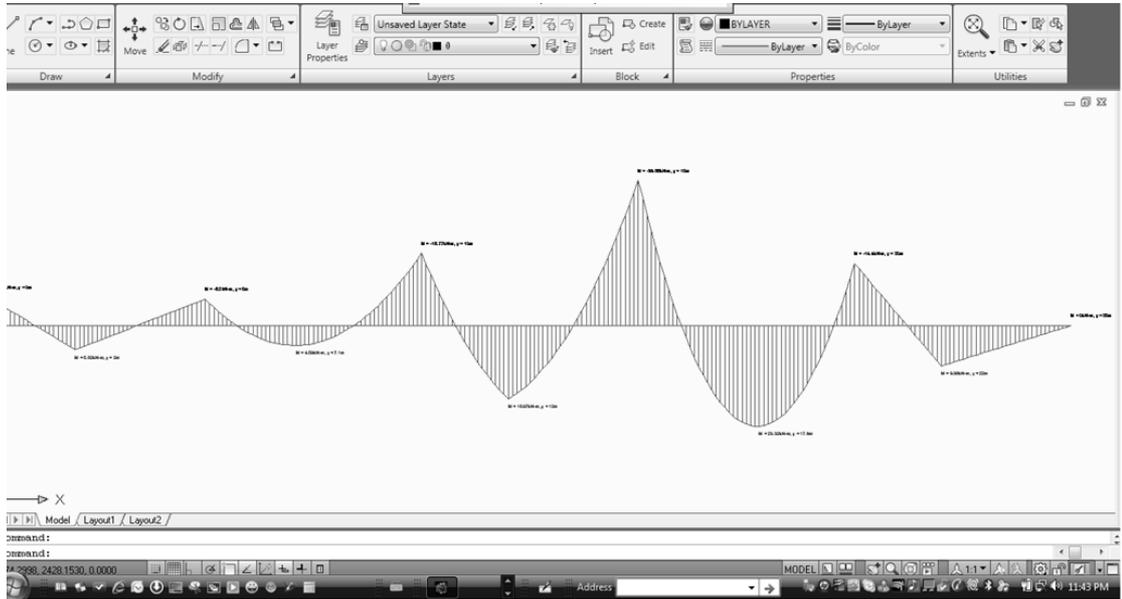


Figure 8. Bending Moment Diagram

Sectional areas of group of bars (mm²) >> Singly Reinforced >> Area of Tension Steel

Moment 57.4 k 0.083 LeverArm, z 380.12

Area of Steel Calculated 826.52 Number Provided 3

Area of Steel Provided 943 Bar Size Provided 20

Bar Size (mm)	Number of Bars									
	1	2	3	4	5	6	7	8	9	10
6	28.3	56.6	84.9	113	142	170	198	226	255	283
8	50.3	101	151	201	252	302	352	402	453	503
10	78.5	157	236	314	393	471	550	628	707	785
12	113	226	339	452	566	679	792	905	1020	1130
16	201	402	603	804	1010	1210	1410	1610	1810	2010
20	314	628	943	1260	1570	1890	2200	2510	2830	3140
25	491	982	1470	1960	2450	2950	3440	3930	4420	4910
32	804	1610	2410	3220	4020	4830	5630	6430	7240	8040
40	1260	2510	3770	5030	6280	7540	8800	10100	11300	12600

Checks For: Minimum Area of Steel Maximum Area of Steel Deflection

Accept Cancel

Figure 9. visRCD Table Advisor as Service