Service Oriented Application in Agent Based Virtual Knowledge Community

Yusuf, Lateef Oladimeji
Department of Computer Science
University of Agriculture, Abeokuta, Ogun State, Nigeria
Tel: 234-803-403-7098 E-mail: truevisionconsulting@yahoo.com

Olusegun Folorunso
Department of Computer Science
University of Agriculture, Abeokuta, Ogun State, Nigeria
Tel: 234-803-564-0707 E-mail: folorunsolusegun@yahoo.com

Akinwale, Adio Taofeek
Department of Computer Science
University of Agriculture Abeokuta, Ogun State, Nigeria
Tel: 234-806-286-7612 E-mail: aatakinwale@yahoo.com

Adejumobi, I. A.
Department of Electrical and Electronics Engineering
University of Agriculture Abeokuta, Ogun State, Nigeria
Tel: 234-703-321-5455 E-mail: engradejumobi@yahoo.com

Abstract

With the availability of the Internet, virtual communities are proliferating at an unprecedented rate. In-depth understanding of virtual community dynamics can help us to address critical organizational and information systems issues such as communities-of-practice, virtual collaboration, and knowledge management. The biggest challenge in fostering a virtual community is the supply of knowledge through services and the willingness to share knowledge with other members. This paper integrates the Service Oriented Architecture and Agent Based Theory to construct a Reinforced Concrete design (RCD) model for Reinforced Concrete Analysts and Designers in virtual communities. The aim of this paper is to elucidate the sharing of knowledge in virtual community from the perspective of RCD serviceability service. Results confirm that RC Designers will share knowledge; if their perceive benefit exceed the cost of their sharing behaviour. This study is useful for the developers of Service oriented Applications for virtual communities to insight into knowledge sharing in cyberspace for RCD.

Keywords: RCD, RCD Beam, AutoCAD, WCF, VKC, SOA, Design interfaces, Visualization, CoP

1. Introduction

Virtual Knowledge Communities (VKC) is presently popular on the internet; it is a medium through which the access and sharing of knowledge and information among communities of similar interest groups are made possible. Agent's technologies are presently being deployed to facilitate the success of VKC, which is a virtual place where knowledge agents can meet, communicate and interact among themselves. Recently, quite a number of works have been done on agent-based knowledge communities but most of these works have not actually considered the possibilities of Service Oriented Applications and the advantage this can bring to the communities. We therefore address the issue of Service Oriented Application problems in the sharing of knowledge in agent-based virtual knowledge communities using RCD Serviceability Service as a case study.

A virtual community is a means for like-minded individuals to pursue common goals. For RC Designers, It is a way to access and share knowledge and RCD information among participants of RCD community without physical or hardware constraints. The concept of a community of interest can be supported in a virtual community using Service Oriented Application to expose services; this enables the RC Designers share their knowledge.

RCD Serviceability is a service oriented application which serve as agent and assist RC Designer to perform RCD tasks that is related to checking for deflection, minimum and maximum area of steel in RC and, possibly relating visual alert to users on possible next line of action by maintaining persistent state and communicating with its owners, other agents or its environment in general (Zoran and Zoran, 2000, Yusuf et al., 2010). Virtual knowledge community is a virtual place where agents like RCD Serviceability can meet, communicate and interact with other services (Maret and Calmet, 2009). It is possible for agents to be sharing some vital information concerning RC Design without reinventing the wheel and helping RCD users within the community to achieve a safe and economic design without much ado. However, to make the system to be rigid against any vulnerability of an intruder such that agents are free to exchange any information without any fear of attack over the internet has become a subject of much concern. Most RCD community users today cannot freely share knowledge with their counterparts due to insecure system.

2. Literature Review

2.1 Service Oriented Application

Service-Oriented Architecture (SOA) is currently gaining momentum, its adopters (both business and IT executives) is increasing in a tremendous manner (Qusay, 2009). 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. According to Mike et al., (2008) SOA is an architectural style for building enterprise solutions based on services. More specifically, SOA is concerned with the independent construction of business-aligned services that can be combined into meaningful, higher-level business processes and solutions within the context of the enterprise. SOA system can reduce development costs, result in higher quality of the design of the systems, and consequently yield higher reliability (Mike et al., 2008). In this work, we propose SOA as a new approach to building RCD Beam that allows RCD businesses to leverage existing assets and easily enable the inevitable changes required to support the RCD business in a virtual community. One of the most important aspects of SOA is that it is a *business*, a *technological* as well as methodological approach (Judith et al., 2007). SOA enables businesses to make business decisions *supported* by technology instead of making business decisions *determined* by or *constrained* by technology. And with SOA, the folks in RCD community finally get to say “yes” more often than they say “no.” One of the biggest deals in the SOA world is the idea that things are not thrown away, the best of software assets used every day is packaged in a way that allow for use, reuse and keep on reusing it securely in the knowledge that future changes will be simple, straightforward, safe, and fast. This makes system less complicated and less expensive to maintain. Mike et al. (2008) described SOA as the careful balance and blending of the big picture and the immediate requirements to the practical application of theory to meet a set of goals in the present and in the future.

2.2 Mobile Agents

Although there is no universal agreement on the precise definition of the term “agent,” definitions tend to agree on more points than they disagree. Some modellers consider any type of independent component (software, model, individual, etc.) to be an agent (Bonabeau 2001); an independent component’s behaviour can range from primitive reactive decision rules to complex adaptive artificial intelligence (AI). Others insist that a component’s behaviour must be adaptive in order for it to be considered an agent; the agent label is reserved for components that can in some sense learn from their environments and change their behaviours in response. Casti (1997) argues that agents should contain both base-level rules for behaviour as well as a higher-level set of “rules to change the rules.” The base level rules provide responses to the environment while the “rules to change the rules” provide adaptation. Jennings (2000) provides a computer science view of agency emphasizing the essential characteristic of *autonomous* behaviour. The fundamental feature of an agent is the capability of the component to make independent decisions. This requires agents to be active rather than purely passive.

Why is agent-based modeling becoming so widespread? The answer is because we live in an increasingly complex world. First, the systems that we need to analyze and model are becoming more complex in terms of their interdependencies. Traditional modeling tools are no longer as applicable as they once were. An example application area is the Service Oriented Application Serviceability Service for Reinforced Concrete Design Application using www as a medium of interaction as described in Section 3. Second, some systems have always been too difficult to use on different platforms; modeling RCD serviceability service on any platform is visible because the assumption made by Foloruso et al., 2010 in there paper "SOA-RTDBS: A service oriented architecture (SOA) supporting real time database systems" is also evident for service oriented application in RCD. Third, data are becoming organized into databases at finer levels of granularity. Micro-data can now support micro simulations. And fourth, but most importantly, computational power is advancing rapidly. We can now compute large-scale micro-simulation models that would not have been plausible just a couple of years ago.

According to Kuo-Huang et al., (2007); a mobile agent is a kind of software program that can migrate from one host to another in a heterogeneous network. Also known as travelling agents, these programs will shuttle their being, code and state among resources. They are network nomads that act as personal representative, working autonomously through networks. They are able to visit network nodes directly using available computing power and are not limited by platform. The technology has become an alternative approach for the design and implementation of distributed systems to the traditional Client/Server architecture. Mobile agents can migrate from one system to another during their execution and communicate amongst one another, clone, merge and co-ordinate their computations. Mobile agents are autonomous agents in the sense that they control their relocation behaviour in pursuit of the goals with which they are tasked. Main fields of application for mobile agents are information retrieval on the www, distributed database access, parallel processing, automation of electronic marketplaces and others. Mobile agent frameworks are currently rare, due to the high level of trust required to accept a foreign agent into one's data server. However with the advances in SOA technology, mobile agent systems are expected to become more popular in the future.

2.3 Virtual Knowledge Community

The concept of a community of practice or a community of interest can be supported in a virtual community in order to bring the appropriate parties together and to share their knowledge (Maret and Calmet, 2009). The advantage of this is that the members of a community centered on one specific topic or practice will only be presented with knowledge from domains they are, or at least are relatively likely to be, interested in.

The concept of knowledge cluster is used to represent a piece of knowledge from the agent's repository. Agents share and exchange knowledge clusters. A community consists of a domain of interest (a knowledge cluster), a leader (an agent), a policy and an unspecified number of member agents. The leader has created this community to achieve a goal (corresponding to the domain of interest). Each community is associated to a single policy which defines the community and which is up to the community leader. For instance, depending on the policy, a message buffer stores for a given duration or under given rules exchanged messages within this community. Quite a number of researchers that have worked in the VKC and agent based environments (Boella et al. 2006; Portillo-Rodrguez et al., 2007; Endsuleit, 2007) have all pointed out the necessity for better solutions to intrusions and other security problems associated with this research area.

The term community of practice (CoP) was coined by Lave and Wenger (1991) to describe an activity system that includes individuals who are united in action and in the meaning that action has for them and for the larger collective. Communities of practice are not formal structures, such as departments or project teams. Instead, they are informal entities, which exist in the minds of their members, and are glued together by the connections the members have with each other, and their specific shared problems or areas of interest. Wenger (1998) asserts that the generation of knowledge in communities of practice occurs when people participate in problem solving and share the knowledge necessary to solve the problems. Researchers have observed that creating and supporting communities of practice is a strong alternative to building teams (Nirenberg, 1994/1995), especially in the context of new product development and other knowledge work (Stewart, 1997).

Among the chief reasons why communities of practice are efficient tools for knowledge generation and sharing is the fact that most of a firm's competitive advantage is embedded in the intangible, tacit knowledge of its people and that competencies do not exist apart from the people who develop them (Dougerty, 1995). It was observed that tacit knowledge is embedded in the stories people tell (Horvath, 1999) and not only new knowledge but also skills are discursively produced and disseminated in conversations and networking activities (Araujo, 1998; Brown and Duguid, 1991; Weick, 1996). Therefore, one of the ways to help people share and internalize tacit knowledge is to allow them to talk about their experiences, and to exchange their knowledge while working on specific problems. Since opportunities for face-to-face interactions are rather limited in today's globally dispersed multinational companies, virtual communities of practice that are supported by internet technologies are among few viable alternatives to live conversations and knowledge exchange.

The successful functioning of a knowledge-sharing community of practice is impossible without an active participation of a substantial part (ideally, all) of its members. Dixon (2000) argues that the community of practice model allows organizations to overcome barriers to sharing information that conventional technological-based KM systems often encounter. For example, people, people who are reluctant to contribute when asked to write something up for a database are willing to share information when asked informally by their colleagues (Dixon, 2000). Members' contributions to virtual CoPs are not limited to posting lengthy and well thought through knowledge entries. For a community to be truly vibrant there should be an active participation of members in other knowledge-exchange activities: engaging in live chats, Q&A sessions, providing asynchronous feedback on previous postings, etc. (Hayes and Watsham, 2000).

Research shows that there are numerous reasons individuals could have for sharing their knowledge with other members of a CoP online, ranging from self-esteem boosting to altruistic and conformist considerations (McLure and Faraj, 2000). Furthermore, Osterloh and Frey's (2000) research on intrinsic motivation for knowledge sharing suggests that intrinsic motives are much more powerful enablers of such sharing than are extrinsic (e.g., monetary or administrative) stimuli.

However, posting of knowledge entries and other active contributions by some members of community represents only one side of the equation: the supply of new knowledge. For a community to be vibrant there should also be an active participation on the demand side: numerous members should be visiting the CoP web site, using online search tools or posting questions when they search for advice or information (Cross et al., 2001). Therefore, the second requirements (willingness to share knowledge and willingness to use a CoP as a source of knowledge) apply to any community of practice, be it face-to-face or virtual. The study reported here deals with virtual online communities of practice and, therefore, it is necessary to add one more requirement: for a virtual community to be successful, its members need to be comfortable with participating in a computer-mediated, Internet-based community of practice, which involves very little face-to-face communication.

2.4 Reinforced Concrete Design

Much of the theory of RC structures is fundamental, detailed design invariably conform to the requirements of a code of practice. We adopt Eurocode No. 2 Design of Concrete Structures, 1992. Reinforced concrete is a composite material made from concrete and steel bars. It is a strong, durable building material that can be cast into any shape, or size ranging from simple beams spanning a few metres in domestic housing, to massive structures. Its utility and versatility are achieved by combining the best features of concrete and steel (Bill et al., 2009). The aims of design are to achieve an acceptable probability that the structure will perform satisfactorily during its intended life. With an appropriate degree of safety, the structure should sustain all the loads and deformations of normal construction and use, have adequate durability, and resistance to the effects of misuse and fire. Calculations alone do not produce safe, serviceability and durable structures. Equally important are the suitability of the materials, quality control and supervision of workmanship during construction. To produce a structure which is economical to construct, maintain and service throughout its design life, the engineer has a responsibility to ensure that any structure is designed and constructed in such a way that:

- With acceptable probability, the structure will remain fit for the use for which it is required, keeping in mind the intended life of the structure and its cost.
- With appropriate degree of reliability. The structure will sustain all loads and external influences likely to occur during construction and subsequent occupation. It should also have adequate durability to keep maintenance costs to minimum.

The structure should also be design in such a way that it will not be damaged by events like explosions, impact or accidents to an extent disproportionate to the original cause. These requirements can be achieved by making a suitable choice of materials, paying proper attention to design and detailing, and specifying control procedures for all stages of design and construction. Limit states are defined as states beyond which the structure no longer satisfies the performance requirements of the design and are classified as Ultimate Limit States and Serviceability Limit States. Ultimate Limit States are associated with collapse or other forms of structural damage likely to endanger life. Serviceability Limit States are associated with poor performance of the structure which, even though not life-threatening, must be avoided. Limit State Design admits that there is an inherent variability in load, materials and methods of design and construction which makes it practically impossible to achieve complete safety against all possible shortcomings.

Much of the groundwork for this work was laid by earlier work by Yusuf et al., (2009). They visualized a simple beam by automatically generating reinforcement properties for the purpose of beam detailing. Yusuf et al., (2010) observed that user's perception was suppressed by not allowing them to use their intuition to make their choice from the steel table. They consequently modified the SSRCBS tool to incorporate visRCD Table Advisor. They noted that visRCD Beam interface was created as input visualization environment while AutoCAD interface was enhanced and borrowed as output visualization environment. The intermediate visualization environment which is very important in RC design was neglected. They therefore add visRCD Beam Table Advisor as intermediate visualization environment to incorporate visualization into every step within the RCD beam process by using Information Visualization approach to check and circumvent failure at every point where steel reinforcement is required. Fady (2008) also developed program for analysis and design of beams up to three spans, upon the input of beam parameters, the program automatically fix bar sizes. This program was created using the relatively new Action script language. The limitation of Yusuf et al., (2009 and 2010) is that their tools will not design and visualize more than one span of beam while Fady's tools will analyze, design and visualize beam up to three spans pin end condition only. RCD Beams encounter in real life are usually continuous (indeterminate) of many spans usually more than one span and can probably go beyond three spans with varying end conditions. We considered both simple and continuous beams of varying end conditions with infinite number of spans. The computer memory shall be our limitation. Yusuf et al., (2009) considers only four loads (Knife edge, linearly distributed, Equilateral triangular, and Right angled triangular loads) acting on beam structures while Fady considered only two loads (Knife edge and linearly distributed loads). We believe more loads need to be considered. Example of such loads is; couple or moment load, the loads considered by Yusuf et al., (2009) spread over the entire span; we believe that loads can be on any location within the span and should not necessarily spread over the entire span. We shall elaborate on this further under methodology. Our major contribution here will be the integration of a host of techniques to create a novel application that is both usable and useful in any RCD domain using SOA approach. A service-oriented architecture for RC design is an information technology approach or strategy for RC in which RC design application tools make use of (perhaps more accurately and in a synchronized manner) rely on Data-based services available in a network such as the World Wide Web. 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. Folorunso et al., 2010 described how SOA can support RTDBS, their approach was highly theoretical; no actual implementation was carried out for a specific real-time database problem. We shall adopt their approach to implement SOA for RCD. Serviceability limit state for RCD will be exposed as a service through RCD table advisor.

3. System Design

Creating service oriented architecture for RC design community 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. But we can start seeing returns on our RC design in SOA community investment very quickly, without having to rewrite all our software. Figure 1 is a simple software architecture related to RC design

This is how it works:

The Browser is a program located on a user's device (PC, laptop, PDA, or cell phone) through which members of RCD community accesses the RC design applications using a Web site. Many members of the community can access the application at the same time; so many browsers will typically link to the Web server. The primary job of the browser is to display information and accept input from the community user.

The Web Server manages when and how the many Web pages are sent to the browsers of the users who access the RC design application. (Web servers may do other things as well, but we are concentrating on its primary service.)

The Reinforced Concrete Design (RCD) Application carries out the business process that is being executed, which in this case means carrying out the necessary steps to collect, output and visualize RC design data at the member's request, if possible. This component embodies the components of RC design's business practices for interacting with customers. They are RC design Beams, RC design Slabs, RC design Columns, RC design Foundations, RC design Retaining Walls/Abutments/ RC design Culverts etc.

The Database Server is computer software that reads data from a database in a real-time manner (Folorunso et al., 2008) and sends the data to where it is needed within the RC design environment.

The Database is where the definitions of the RC design business data and the RC design data itself are stored. Information passes from the browser to the Web server to the RCD application services, which decides what to do next. The processing application might pass RCD data to the database server to write to disk, or it may request data from the database, or it may simply send information back to the browser through the Web server. What the RCD application service does depends upon the information and commands passed to it by the community user via the browser.

The diagram in Figure 1 can also be referred to as a RCD *community business service* which means in simple terms, the wrappings up of everything we have to do to make a particular business within the community function properly.

In Figure 2, a credit-checking and RCD Table Advisor components is added to the RCD business diagram. Credit-checking service is called on when community user make request to use the RCD services while RCD Table Advisor service is called when it is necessary to pick reinforcement and do serviceability checking for any RCD components; these must be done in a real-time manner as not to frustrate the community user. In the figure, we don't show or even care about how the credit checking is done. For the sake of simplicity, it is assumed that the credit-checking software component is a database run by an external company and simply provides a service in a real time manner. The company using this credit-checking software is confident that the service conducts a credit check in the right way. We shall subsequently show how the RCD Table Advisor will be run as a service under implementation.

The RCD application simply requests the credit-checking service and passes along the necessary information (a person's name and password). The credit-checking component consults its information sources, does some calculations, and passes back a credit rating. The credit checking component may connect with many computers, consult many different data sources, and use a very sophisticated algorithm to calculate the credit rating, but this is of no concern to the RCD-processing application so far the information is received in a timely manner. The same scenario is applicable to the RCD Table Advisor. As far as the RCD application is concerned, credit checking and RCD Table Advisor are just *black boxes.* Also, we need to emphasize that the credit-checking component *does only credit checking*. While RCD table Advisor only provide reinforcement steels and do serviceability checks. They don't offer a wide range of services. They have precisely narrow defined scope — that is, they do "just one thing" — that they can be used and reused as building blocks. SOA's use and reuse of components makes it easier to build new applications as well as change existing applications. Using well-proven, tested components makes testing new applications more efficient.

4. System Implementation

Windows Communication Foundation (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.

Three major steps are involved while creating and consuming the WCF services for RCD Beam. These are:

- Create the RCD Beam Serviceability Services. (Creating)
- Binding an address to the service and host the RCD Beam Serviceability Service. (Hosting)
- Consuming the RCD Beam Serviceability Service. (Consuming)

Step 1: Creating the RCD Beam Serviceability Service

In WCF, RCD Serviceability such as check for minimum steel, check for maximum steel and check for deflection are exposed services which are exposed as contracts. Contract is a neutral way of describing the RCD Serviceability service. Mainly we have four types of contract.

- Service Contract

  This contract describes all the available operations that client can perform on the RCD Beam Serviceability service. .Net uses "System.ServiceModel" Namespace to work with WCF services. We used ***ServiceContract*** attribute to define the RCD Beam Service contract. We apply this attribute on interface. We also use ***OperationContract attribute to indicate explicitly which method is used to expose part of WCF contract.*** In summary,

  [ServiceContract] applies at the class or interface level.

  [OperationContract] applies at the method level.

- Data Contract

  The Data contract defines our data types that passed in and out to the RCD Beam Serviceability service.

[DataContract] attributeis used at the custom data type definition level, i.e., at class or structure level

[DataMember] attribute is used to fields, properties, and events.

- Fault Contract

  This contract describes the error raised by the services.

  [FaultContract(<type of Exception/Fault>>)] attribute is used for defining the fault contracts.

- Message Contracts

  This contract provides the direct control over the SOAP message structure.

  [MessageContract] attribute is used to define a type as a Message type.

  [MessageHeader] attribute is used to those members of the type we want to make into SOAP headers

  [MessageBodyMember] attribute is used to those members we want to make into parts of the SOAP body of the message.

Sample Program for Service Creation for RCD Beam Serviceability Service Contract

```
Imports System
Imports System.ServiceModel
Imports System.ServiceModel.Description
Imports System.Web
<ServiceContract()>
Public Interface IService1
    <OperationContract()>
    Function MsgAlertMinSteel(ByVal AreaProvided As Single, ByVal BeamWidth As Single, ByVal
BeamHeight As Single, _
ByVal BarSizeProvided As Single, ByVal Links As Single, ByVal ConcreteCover As Integer, _
ByVal ConcreteGrade As Integer, ByVal SteelGrade As Integer) As String
    <OperationContract()>
    Function MsgAlertMaxSteel(ByVal AreaProvided As Single, ByVal BeamWidth As Single, ByVal
BeamHeight As Single) As String
    <OperationContract()>
    Function MsgAlertDeflection(ByVal SpanLength As Single, ByVal Moment As Single, ByVal
SpanEffectiveRatio As Integer, ByVal BeamWidth As Integer, ByVal BeamHeight As Single, ByVal
BarSizeProvided As Integer, ByVal Links As Integer, _
ByVal ConcreteCover As Integer, ByVal SteelGrade As Integer, ByVal AreaCalculated As Single, ByVal
AreaProvided As Single) As String
End Interface
```

Here “IService1” is a service exposed by using the ServiceContract attribute. The service exposes three Functions (Method), “MsgAlertMinSteel”, “MsgAlertMaxSteel” and “MsgAlertDeflection” through OperationContract.

```
Imports System
Imports System.ServiceModel
Imports System.ServiceModel.Description
Imports System.Web

Public Class Service1
    Implements IService1
    Public Function MsgAlertMinSteel(ByVal AreaProvided As Single, ByVal BeamWidth As Single, ByVal
BeamHeight As Single, _
ByVal BarSizeProvided As Single, ByVal Links As Single, ByVal ConcreteCover As Integer, _
                        ByVal ConcreteGrade As Integer, ByVal SteelGrade As Integer) As String
Implements IService1.MsgAlertMinSteel
        Dim BeamDepth As Single
        Dim MinSteel As Single
        Dim fctm As Single
        Dim fctm26 As Single
        On Error GoTo userError
        BeamDepth = BeamHeight - BarSizeProvided / 2 - Links - ConcreteCover
        MinSteel = (100 * AreaProvided) / (BeamWidth * BeamDepth)
        fctm = 0.3 * ConcreteGrade ^ (2 / 3)
        fctm26 = Math.Round((26 * (fctm / SteelGrade)), 2)
        If ConcreteGrade > 50 Then
            MsgBox("Grade of Concrete should no be more than 50N/mm^2")
            MsgAlertMinSteel = "Grade of Concrete should no be more than 50N/mm^2"
            Return MsgAlertMinSteel
            Exit Function
        Else
            fctm26 = fctm26
        End If
        If MinSteel >= fctm26 And MinSteel >= 0.13 Then
            MsgAlertMinSteel = "Minimum Area of Steel is satisfied" & vbCrLf & "Minimum Percent of
Steel calculated is " & Math.Round(MinSteel, 2) & "%" _
          & vbCrLf & "Minimum recommended is 0.13% or 26x(fctm/fyk%) = " & fctm26 & "%"
        Else
            MsgAlertMinSteel = "Minimum Area of Steel is not satisfied" & vbCrLf & "Minimum Percent of
Steel calculated is " & Math.Round(MinSteel, 2) & "%" _
            & vbCrLf & "Minimum recommended is 0.13% or 26x(fctm/fyk%) = " & fctm26 & "%"
        End If
        Return MsgAlertMinSteel
        Exit Function
userError:
        MsgBox("Error due to mis use or lack of understanding of RCD by the user")
    End Function
    Function MsgAlertMaxSteel(ByVal AreaProvided As Single, ByVal BeamWidth As Single, ByVal
BeamHeight As Single) As String Implements IService1.MsgAlertMaxSteel
        Dim MaxSteel As Single
        On Error GoTo userError
        MaxSteel = (100 * AreaProvided) / (BeamWidth * BeamHeight)
        If MaxSteel <= 4 Then
            MsgAlertMaxSteel = "Maximum Area of Steel is satisfied" & vbCrLf & "Maximum Percent of
steel calculated is " & Math.Round(MaxSteel, 2) & "%" _
                                        & vbCrLf & "Maximum recomended is 4.0%"
        Else
            MsgAlertMaxSteel = "Maximum Area of Steel is not satisfied" & vbCrLf & "Maximum Percent
of steel calculated is " & Math.Round(MaxSteel, 2) & "%" _
                                        & vbCrLf & "Maximum recomended is 4.0%"
        End If
        Return MsgAlertMaxSteel
        Exit Function
userError:
        MsgBox("Error due to mis use or lack of understanding of RCD by the user")
    End Function
    Function MsgAlertDeflection(ByVal SpanLength As Single, ByVal Moment As Single, ByVal
SpanEffectiveRatio As Integer, ByVal BeamWidth As Integer, ByVal BeamHeight As Single, ByVal
BarSizeProvided As Integer, ByVal Links As Integer, _
ByVal ConcreteCover As Integer, ByVal SteelGrade As Integer, ByVal AreaCalculated As Single, ByVal
AreaProvided As Single) As String Implements IService1.MsgAlertDeflection
        Dim BeamDepth As Single
        Dim ServiceStress As Single
        Dim MsgAlert As String
        Dim mm, ModificationFactorForTensionSteel, ActualDepthRequired, Fs As Single
        On Error GoTo userError
        BeamDepth = BeamHeight - BarSizeProvided / 2 - Links - ConcreteCover
        ServiceStress = 2 / 3 * SteelGrade * AreaCalculated / AreaProvided
        If ServiceStress >= 477 Then
            Fs = 477
        Else
            Fs = 477 - ServiceStress
        End If
        mm = (Moment * 10 ^ 6) / (BeamWidth * BeamDepth ^ 2)
        mm = mm + 0.9
        mm = mm * 120
        ModificationFactorForTensionSteel = 0.55 + Fs / mm
      If ModificationFactorForTensionSteel > 2 Then ModificationFactorForTensionSteel = 2
      ActualDepthRequired = (SpanLength * 1000) / (ModificationFactorForTensionSteel * _
     SpanEffectiveRatio)
        If ActualDepthRequired <= BeamDepth Then
            MsgAlertDeflection = "Deflection criteria is satisfied" & vbCrLf & _
                "Beam Depth = " & BeamDepth & vbCrLf & _
                "Actual Depth Required = " & Math.Round(ActualDepthRequired, 2)
            Else
                MsgAlertDeflection = "Deflection criteria is not met" & vbCrLf & _
                    "Beam Depth = " & BeamDepth & vbCrLf & _
                    "Actual Depth Required = " & Math.Round(ActualDepthRequired, 2)
            End If
            Return MsgAlertDeflection
            Exit Function
userError:
            MsgBox("Error due to mis use or lack of understanding of RCD by the user")
        End Function
End Class
```

Service1 is a class which implements the interface IService1. This class defines the functionality of methods exposed as service.

Step 2 Binding and Hosting

Each service has an end point. Clients communicate with this end point only. End point describes three things:

- Address
- Binding types
- Contract Name (which was defined in step 1)

Address: Every service must be associated with a unique address. Address mainly contains the following two key factors:

- Transport protocol used to communicate between the client proxy and service.

  In our case, the address is HTTP (http:// or https://)
- Location of the service which describes the targeted machine (where service is hosted) complete name or path and optionally port. Ours is localhost:54337

  Our full address is: http://localhost:54337/Service1.svc

Binding is a set of choices regarding the transport protocol. For example, the basic binding uses .net class that implements BasicHttpBinding, http/https transport, text/MTOM encoding, it is interoperable and it is used to expose a WCF service as a legacy ASM.

Every service must be hosted in a host process. Our service was hosted in Internet Information Service (IIS) Hosting. IIS manages the life cycle of host process. The limitation is that only HTTP transport schemas WCF services are hosted in IIS. IIS hosting is same as hosting the traditional web service hosting. Create a virtual directory and supply a .svc file. Figure 3 shows the WCF Test Client with ASP.NET development Server Port 54337.

Step 3: Consuming the RCD Beam Serviceability Service

With WCF, the client always communicates with the proxy only. Client never directly communicates 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.

To consume RCD Beam serviceability service using the proxy, the service must be running. In the solution explorer of the legacy program for RCD Beam, right click on "RcdBeam" solution and click on "Add Service Reference", select http://localhost:54337/Service1.svc from the address list, give a choice name (i.e., RcdServiceabilityServiceReference) to the namespace textbox and click OK button. A service reference is created under the reference folder in the RcdBeam solution. We use the following proxy class to consume the WCF service for check minimum steel Button_Click event:

```
On Error GoTo userError
            Dim Client As New RcdServiceabilityServiceReference.Service1Client()
            Me.MsgAlert.Text = Client.MsgAlertMinSteel(AreaProvided:=Me.AreaProvided.Text,
BeamWidth:=Me.TxtBeamWidth.Text, BeamHeight:=TxtBeamHeight.Text,
BarSizeProvided:=Me.BarSizeProvided.Text, Links:=TxtLinks.Text, ConcreteCover:=TxtConcreteCover.Text,
ConcreteGrade:=Me.TxtConcreteGrade.Text, SteelGrade:=Me.TxtSteelGrade.Text)
            Exit Sub
userError:
            MsgBox("Serviceability Service is not available" & vbCrLf & "Please Contact your RCD
Serviceability Provider" & vbCrLf & "Thank you")
```

Our RCD Serviceability Service also serves as a mobile agent to the RcdBeam interface. It is a network nomad that acts as personal representative, working autonomously through networks. They are able to visit network nodes directly using available computing power and are not limited by platform. SOA technology in RcdBeam has become an alternative approach for the design and implementation of distributed RcdBeam systems. A mobile agent is a kind of software program that can migrate from one host to another in a heterogeneous network (Kuo-Huang et al., 2007). Also known as travelling agents, these programs will shuttle their being, code and state among resources (Ogunleye and Ogunde, 2010).

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 is generated. The error is cached with message alerting the user that the service is not available.

5. Result and Discussion

The key issue for this RCD Beam prototype for community users was to emphasize the use of RCD table advisor to pick reinforcement from the steel table through the web service which also check for serviceability and monitors the messages delivered via the AutoCAD environment. The message, in fact, is that of monitoring the inputs for different type of beam end conditions, various types of loads and output for the bending moment, shear force and beam detail diagrams including explanatory text labels.

However, RCD community users have to learn and master a challenging set of skills in RC analysis and designs to be able to enter valid input in other to produce content that delivers right messages in the way it is intended and tampering with that content would require a skill in the manipulation of drawings in AutoCAD environment. RCD table advisor expose RCD serviceability services (i.e., check for minimum and maximum steel, check for deflection) through messages from the remote server or through the internet. Also, the user might expect to get the message they subscribed to if and only if the service is available from the service provider. Figure 4 shows a message when the service is not available to the client. The message is delivered through proxy. Alteration can be made to the drawings using the modify tools in the AutoCAD environment. Examples of such alterations would be using zooming tool to view context + details. In case of detailing, we might think of enlarging parts of the image to highlight hidden features in the hope to make it better stand out on screen.

Also, it would be naive to imagine that our preferences would apply to every RCD community users. But, tampering with the message content is highly feasible or desirable and a plus to user centered design approach. The RCD properties and Beam Loading interface which was designed as a dialog boxes also help the user to pick valid input (Figure 5); user selects what he wants through the combo or list box. Jakob Nielsen states similar ideas in his book "Designing Web Usability" (Nielsen 2000). Speaking specifically about web pages he states that the emphasis is to be put on the content instead of various other aspects such as site navigation. The RcdBeam tool interface needs not to be too sophisticated, however. It should be responsive and concentrate on providing the content quickly and easily. It should also make the needed features pleasurable to use.

The main benefit of RCD Beam is the fact that it can handle any type of beam with any established known loading combinations (Figure 6). It addressed the needs of the RCD analysts as far as beam design is concerned. It is available most of the times when a community analyst needs it if not hindered by RCD serviceability service provider.

This paper is mostly interested in the consumption aspects of RcdBeam by RCD community analysts. The hypothesis is that the RCD community users will subscribe to the tool for their routine analysis and design activities. In this case consumption means the use of RcdBeam tool to design RC Beams. It can include both determinate and indeterminate beam structures. The act of using the tool is the deciding factor when deciding if something has been consumed or not.

We received good design feedback from participants suggesting how best to move towards redesign. For example, many users disliked the black background colour of the textbox for message alert. They wanted a white background with fore-colour in green indicating success while red fore-colour to indicate danger or failure. They also wanted to see all the messages sent to the message alert textbox to be appended for ease of review. Users expressed strong concerns about the desirability of entering the design moment directly as an option with 'area of steel calculated' automatically generated.

It is rare to encounter a clear-cut expression of preference, or the reverse, for a thoroughly explored innovative interface, and the outcome of an overall satisfaction questionnaire and briefing completed by participants is no exception. Responses to the questionnaire revealed no significant differences, though users preferred the new prototype tool, several areas of future work were identified. Inevitably users requested a long list of desirable features and these must be examined to see how they would affect users without jeopardizing ease of use for the novice. It was also recognized that studies must be carried out on how to incorporate our tool into hand-held devices like the newly introduced Windows Phone 7 using pens and touch-screens rather than the mice and keyboards that necessarily had to be employed in the reported studies.

What should be studied is what the user think of the method after all the other functionality is included to the controls. As the prototype was the preferred choice for the participants, further studies should be done to break down the user experience factors by altering the elements present in the interface. Such studies would hopefully clarify the relative importance of the functions and the looks that the prototype implemented.

6. Conclusions and Future work

With its own small part, the RcdBeam tool prototype for SOA in VKC has paved way of what is to come. The prototype has introduced the use of Service Oriented Application through the web service to check for serviceability in RCD. All the controls work on top of AutoCAD interface; communicate with AutoCAD through interoperability to visualize Bending Moment, Shear Force diagrams and Beam Detailing. It is extremely difficult to say what aspects affected the positive end results when it comes to user experience. But what is clear is that it is not enough to think about application design with mere functional demands and requirements; how things appear and feel is an important factor to the actual end user. It is difficult to name a formal method of creating attractive and pleasant applications from the end-user perspective. Also, the proposed hypotheses about emotional functions (transferring emotional context and creating a connection via quirks) are not validated by these tests alone but they are not shown to be in direct conflict either. Further studies would be needed to refine the assumptions and to see how general those might be.

References

Araujo, L. (1998). Knowing and learning as network, *management learning,* Vol. 29 No. 3, pp. 317-336.

Bill M, John B, Ray H. (2007). Reinforced Concrete Design to Euro code 2. Palgrave Macmillan.

Boella G., Van D., & Torre L. (2006). Security policies for sharing knowledge in virtual communities. *IEEE Transactions on Systems, Man and Cybernetics* Vol.36, N.3, 2006, pp439- 450.

Bonabeau, E. (2001). Agent-based modeling: methods and techniques for simulating human systems. In *Proc. National Academy of Sciences* 99(3): 7280-7287.

Brown, J.S., and Duguid, P. (1991). Organizational learning and communities of practice, *Organization Science,* Vol. 2 No. 1, pp. 40-57.

Casti, J. (1997). *Would-be worlds: how simulation is changing the world of science*, New York: Wiley.

Cross, R., Bogatti P., and Parker, A. (2001). Beyond answers: dimensions of the advice network, *Social Networks,* vol. 23 No. 3, pp. 215-235.

Dixon, N. (2000). *Common Knowledge: How Companies Thrive by Sharing what they know.* Havard Business School Press, Boston, MA.

Dougherty, D. (1995). Managing your core incompetence for corporate venturing. *Entrepreneurship Theory and Practice,* vol. 19 No. 3, pp. 113-135.

Endsuleit R. (2007). *Robust and Private Computations of Mobile Agent Alliances*. PhD Dissertation, University of Karlsruhe, June 2007. Retrieved from iaks-www.ira.uka.de/calmet/dissertationen/diss_endsuleit.pdf.

Fady Rostom (2008). Computer Analysis and Reinforced Concrete Design of Beams, Department of Civil Engineering, University of Nairobi, Kenya.

Folorunso O, Longe HOD, Akinwale AT. (2008). Visualizing Concurrency Control Algorithms for Real-Time Database Systems; *Data Sci. J.* (7).

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).

Hayes, N., and Walsham, G. (2000). Competing interpretations of computer supported co-operative work, *Organization,* vol. 7 No. 1, pp. 460-467.

Horvath, J.A. (1999). Tacit knowledge in the profession, in Sternberg, R., and Horvath, J. (1999), Tacit Knowledge in Professional Practice, Laurence Erlbaum, London.

Jennings, N. R. (2000). On agent-based software engineering; *Artificial Intelligence,* 117:277-296.

Judith Hurwitz, Robin Bloor, Carol Baroudi and Marcia Kaufman. (2007). *Service Oriented Architecture for Dummies;* Wiley Publishing Inc.

Kuo Huang, Yu-Fang Chung. (2007). *Effective migration for mobile computing in distributed networks.* Elsevier, February 2007.

Lave, J., and Wenger, E. (1991). *Situated Learning: Legitimate peripheral participation,* Cambridge university press, new York, NY.

Maret Pierre and Calmet Jacques. (2009). Agent-Based Knowledge Communities. *International Journal of Computer Science and Applications Technomathematics Research Foundation* Vol. 6, No. 2, pp. 1-18, 2009.

McLure, M., and Faraj, S. (2000). It is what one does": Why people participate and help others in electronic communities of practice, *The Journal of Strategic information Systems.* Vol. 9 No. 2-3, pp. 55-173

Mike Rosen, Boris Lublinsky, Kevin T. Smith and Marc J. Balcer. (2008). *Applied SOA: Service Oriented Architecture and Design Strategies;* Wiley Publishing, Inc. Indianapolis, Indiana

Nielsen Jakob. (2000). *Designing Web Usability,* New Riders, cop 2000, 419 pages.

Nirenberg, J. (1994/1995). From team building to community building, *National Productivity Review,* winter, pp. 51-62.

Ogunleye, G.O., and Ogunde, A.O. (2010). Intrusion Detection in agent-Based Virtual Knowledge communities. *Computer and Information Science.* vol. 3, No. 3; www.ccsenet.org/cis

Osterloh, M., and Frey, B.S. (2000). Motivation, knowledge transfer, and organizational forms, *Organization* Science, vol. 11 No. 5, pp. 538-550.

Portillo-Rodrguez J., Aurora V., Juan P. S., Mario P. & Gabriela N. A. (2007). *Fostering Knowledge Exchange in Virtual Communities by Using Agents*. Springer Berlin / Heidelberg, (Lecture Notes in Computer Science), Volume 4715 pp32-39.

Qusay F. Hassan. (2009). Aspect of SOA: An Entry Point for Starters. Annals Computer Science Series 7th Tome 2nd Fasc. 2009.

Stewart, T. (1997). *The invisible key success, Fortune*, August 5, pp. 173-176

Tsai W.T., Xiao B., Paul R.A., and Chen Y. (2006). Consumer-Centric Service-Oriented Architecture: A new Approach, Proceeding of IEEE 2006 International Workshop on Collaborative Computing, Integration, and Assurance (WCCIA), April 2006, pp. 175-180.

Weick, K., and Westley, F. (1996). Organizational learning: reaffirming an oxymoron 11, in Clegg, S., Haroly, C., and Nord, W. (Eds), Handbook of Organization Studies, Sage, London.

Wenger, E. (1998). *Communities of Practice: Learning, Meaning and Identity,* Cambridge university press, Cambridge.

Yusuf, L.O., Folorunso, O., Akinwale, A.T., and Adejumobi, A.I. (2009). Visualizing the behaviour of reinforced concrete beam structure under various types of loadings. *Published by African Journal of Mathematics and Computer Science Research* Vol. 2(10), pp. 202-217, November, 2009.

Yusuf, L.O., Folorunso, O., Akinwale, A.T., and Adejumobi, A.I. (2010). Visualization RCD (reinforced concrete design) table advisor for decision support activity. *Published by African Journal of Mathematics and Computer Science Research* Vol. 3(7), pp. 132-142, July, 2010.

Zoran Putnik and Zoran Budimac. (2000). *Mobile agents – a new and advanced concepts*. Proceedings of the TARA 2000 Conference Novi Sad, Yugoslavia, September 6-7, 2000. VOL. 30, No. 2, 2000, 113-123.

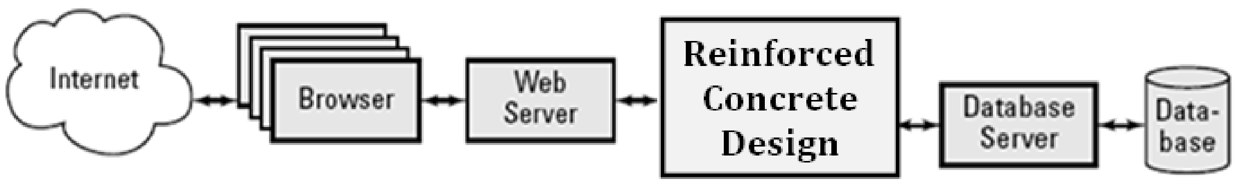


Figure 1. A simple software architecture

(Adapted from Service Oriented Architecture for Dummies by Judith et al., 2007).

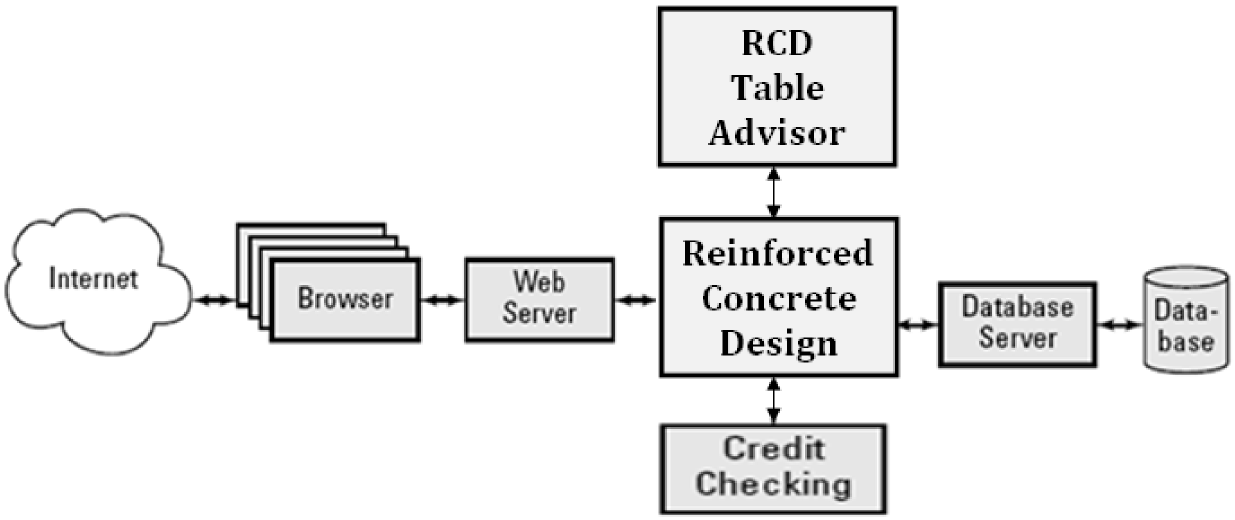
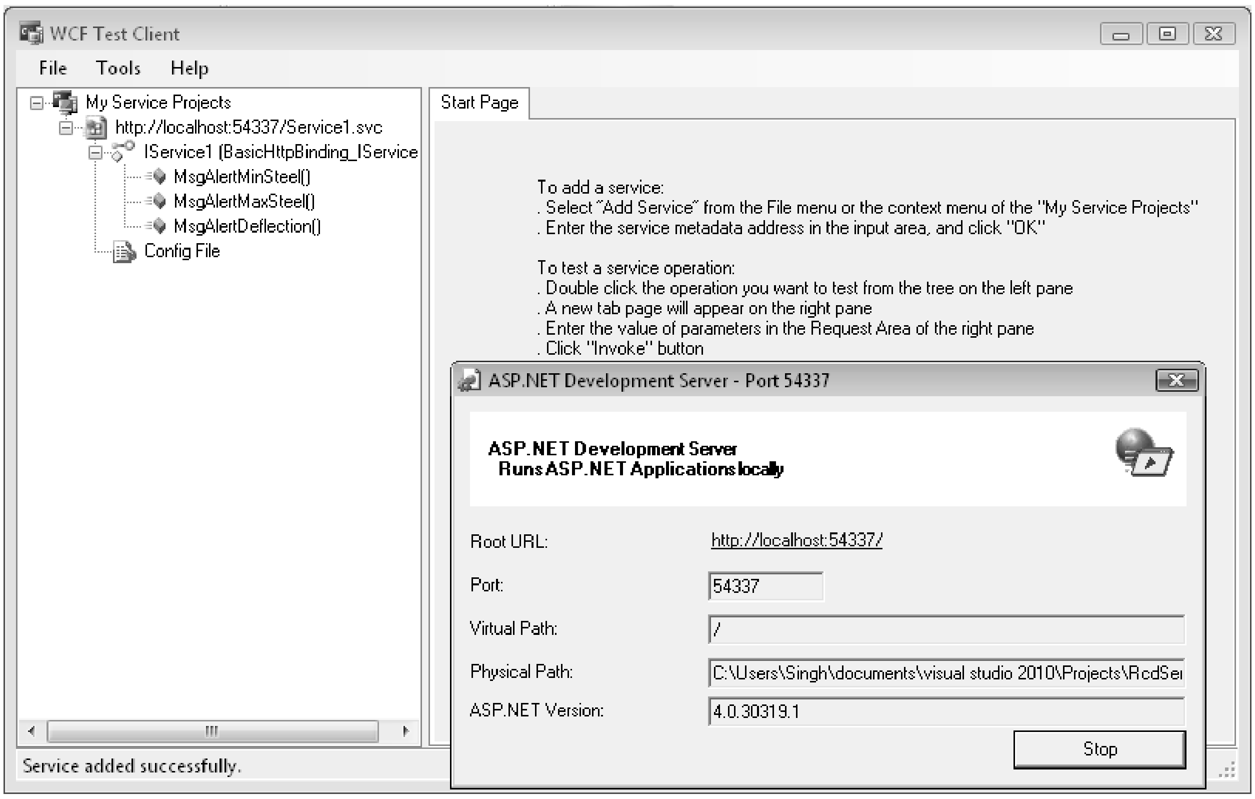


Figure 2. Adding a service oriented component

(Adapted from Service Oriented Architecture for Dummies by Judith et al., 2007).



To add a service:
. Select "Add Service" from the File menu or the context menu of the "My Service Projects"
. Enter the service metadata address in the input area, and click "OK"

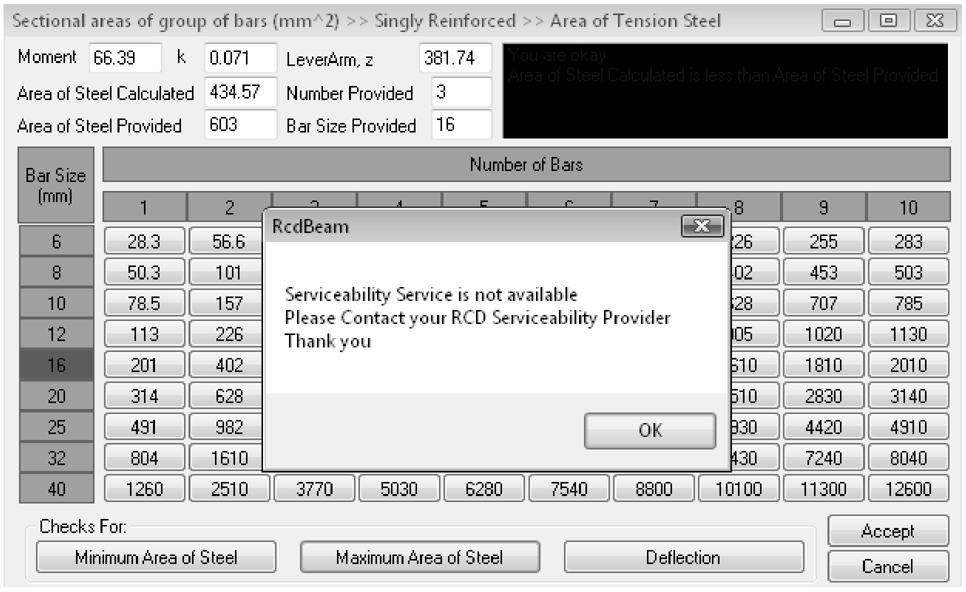
To test a service operation:
. Double click the operation you want to test from the tree on the left pane
. A new tab page will appear on the right pane
. Enter the value of parameters in the Request Area of the right pane
. Click "Invoke" button

ASP.NET Development Server
Runs ASP.NET Applications locally

Root URL:	http://localhost:54337/
Port:	54337
Virtual Path:	/
Physical Path:	C:\Users\Singh\documents\visual studio 2010\Projects\RcdSe
ASP.NET Version:	4.0.30319.1

Service added successfully.

Figure 3. WCF Test Client with ASP.NET development Server Port 54337


Sectional areas of group of bars (mm^2) >> Singly Reinforced >> Area of Tension Steel
RcdBeam
OK
Checks For:
Minimum Area of Steel
Maximum Area of Steel
Deflection
Accept
Cancel

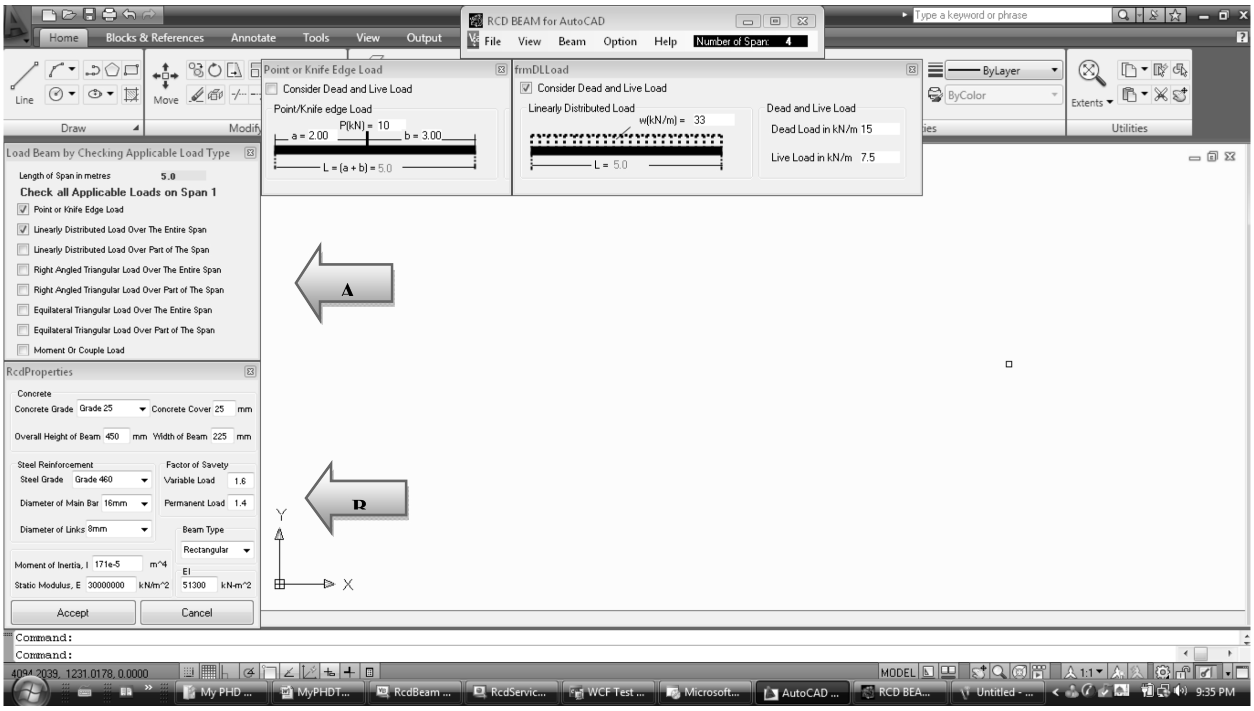
Moment	66.39	k	0.071	LeverArm, z	381.74
Area of Steel Calculated	434.57			Number Provided	3
Area of Steel Provided	603			Bar Size Provided	16

You are okay
Area of Steel Calculated is less than Area of Steel Provided

Bar Size (mm)	Number of Bars									
	1	2	3	4	5	6	7	8	9	10
6	28.3	56.6						26	255	283
8	50.3	101						02	453	503
10	78.5	157						28	707	785
12	113	226						05	1020	1130
16	201	402						10	1810	2010
20	314	628						10	2830	3140
25	491	982						30	4420	4910
32	804	1610						430	7240	8040
40	1260	2510	3770	5030	6280	7540	8800	10100	11300	12600

Serviceability Service is not available
Please Contact your RCD Serviceability Provider
Thank you

Figure 4. Message indicating that the service is not available when Maximum Area of Steel button is clicked



RCD BEAM for AutoCAD

File View Beam Option Help Number of Span: 4

Point or Knife Edge Load

Consider Dead and Live Load

Point/Knife edge Load

P(kN) = 10

a = 2.00

b = 3.00

L = (a + b) = 5.0

frmDLLoad

☑ Consider Dead and Live Load

Linearly Distributed Load

w(kN/m) = 33

L = 5.0

Dead and Live Load

Dead Load in kN/m 15

Live Load in kN/m 7.5

Load Beam by Checking Applicable Load Type

Length of Span in metres 5.0

Check all Applicable Loads on Span 1

- ☑ Point or Knife Edge Load
- ☑ Linearly Distributed Load Over The Entire Span
- Linearly Distributed Load Over Part of The Span
- Right Angled Triangular Load Over The Entire Span
- Right Angled Triangular Load Over Part of The Span
- Equilateral Triangular Load Over The Entire Span
- Equilateral Triangular Load Over Part of The Span
- Moment Or Couple Load

RcdProperties

Concrete

Concrete Grade: Grade 25 Concrete Cover: 25 mm

Overall Height of Beam: 450 mm Width of Beam: 225 mm

Steel Reinforcement

Steel Grade: Grade 460

Diameter of Main Bar: 16mm

Diameter of Links: 8mm

Factor of Savety

Variable Load: 1.6

Permanent Load: 1.4

Beam Type: Rectangular

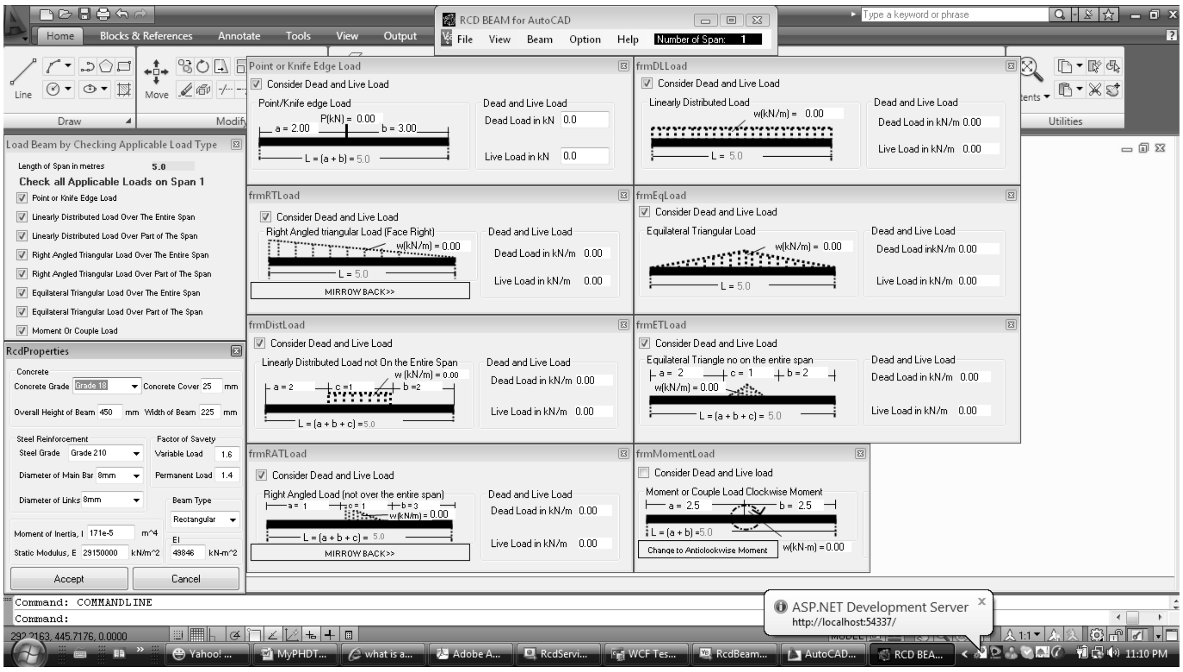
Moment of Inertia, I: 171e-5 m^4

Static Modulus, E: 30000000 kN/m^2

EI: 51300 kN-m^2

```
Command:
Command:
```

Figure 5. Beam Loading Interface (A) and RCD Properties Dialogue Boxes (B)



RCD BEAM for AutoCAD

File View Beam Option Help Number of Span: 1

Point or Knife Edge Load

☑ Consider Dead and Live Load

Point/Knife edge Load: P(kN) = 0.00, a = 2.00, b = 3.00, L = (a + b) = 5.0

Dead and Live Load: Dead Load in kN 0.0; Live Load in kN 0.0

frmDLLoad

☑ Consider Dead and Live Load

Linearly Distributed Load: w(kN/m) = 0.00, L = 5.0

Dead and Live Load: Dead Load in kN/m 0.00; Live Load in kN/m 0.00

frmRTLoad

☑ Consider Dead and Live Load

Right Angled triangular Load (Face Right): w(kN/m) = 0.00, L = 5.0

Dead and Live Load: Dead Load in kN/m 0.00; Live Load in kN/m 0.00

frmEqLoad

☑ Consider Dead and Live Load

Equilateral Triangular Load: w(kN/m) = 0.00, L = 5.0

Dead and Live Load: Dead Load inkN/m 0.00; Live Load in kN/m 0.00

frmDistLoad

☑ Consider Dead and Live Load

Linearly Distributed Load not On the Entire Span: w (kN/m) = 0.00, a = 2, c = 1, b = 2, L = (a + b + c) = 5.0

Dead and Live Load: Dead Load in kN/m 0.00; Live Load in kN/m 0.00

frmETLoad

☑ Consider Dead and Live Load

Equilateral Triangle no on the entire span: a = 2, c = 1, b = 2, w(kN/m) = 0.00, L = (a + b + c) = 5.0

Dead and Live Load: Dead Load in kN/m 0.00; Live Load in kN/m 0.00

frmRATLoad

☑ Consider Dead and Live Load

Right Angled Load (not over the entire span): a = 1, c = 1, b = 3, w(kN/m) = 0.00, L = (a + b + c) = 5.0

Dead and Live Load: Dead Load in kN/m 0.00; Live Load in kN/m 0.00

frmMomentLoad

Consider Dead and Live load

Moment or Couple Load Clockwise Moment: a = 2.5, b = 2.5, L = (a + b) = 5.0, w(kN-m) = 0.00

Load Beam by Checking Applicable Load Type

Length of Span in metres 5.0

Check all Applicable Loads on Span 1

- ☑ Point or Knife Edge Load
- ☑ Linearly Distributed Load Over The Entire Span
- ☑ Linearly Distributed Load Over Part of The Span
- ☑ Right Angled Triangular Load Over The Entire Span
- ☑ Right Angled Triangular Load Over Part of The Span
- ☑ Equilateral Triangular Load Over The Entire Span
- ☑ Equilateral Triangular Load Over Part of The Span
- ☑ Moment Or Couple Load

RcdProperties

Concrete Grade: Grade 18 Concrete Cover: 25 mm

Overall Height of Beam: 450 mm Width of Beam: 225 mm

Steel Grade: Grade 210; Diameter of Main Bar: 8mm; Diameter of Links: 8mm

Factor of Savety: Variable Load 1.6; Permanent Load 1.4

Beam Type: Rectangular

Moment of Inertia, I: 171e-5 m^4

Static Modulus, E: 29150000 kN/m^2

EI: 49846 kN-m^2

```
Command: COMMANDLINE
Command:
```

ASP.NET Development Server
http://localhost:54337/

Figure 6. RCD Beam with known (common) Load Combinations