Designing a business application for workflows in Cloud Computing

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Software Engineering

by

Ravneet Kaur Chawla

(800931017)

Under the supervision of

Mrs.Anju Bala

(Assistant Professor)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2011

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Designing a business application for workflows in Cloud Computing"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs.Anju Bala,* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Ravneet Kaur Chawla)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(**Mrs.Anju Bala**)
Assistant Professor
Computer Science and Engg.Deptt.
Thapar University
Patiala

Countersigned by

(Dr. Maninder Singh)
Associate Professor and Head
Computer Science and Engg.Deptt.
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Designing a business application for workflows in Cloud Computing"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs.Anju Bala,* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Ravneet

(Ravneet Kaur Chawla)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Anju Bala

(Mrs.Anju Bala)
Assistant Professor
Computer Science and Engg.Deptt.
Thapar University
Patiala

Countersigned by

(Dr. Maninder Singh)
Associate Professor and Head
Computer Science and Engg.Deptt.
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

Many people have shared their time and expertise to help me accomplish my goal. First, I would like to sincerely thank my guide, Mrs.Anju Bala, Assistant Professor, Thapar University, Patiala for her constant support and guidance. Her instant responses to my countless inquiries have been invaluable and motivational. It was a great opportunity to work under her supervision.

I would also like to thank Dr.Inderveer Chana, Assistant Professor, Thapar University for her invaluable suggestions during the period of my work.

Finally I wish to thank my family and friends for their immense love and encouragement throughout my life without which it would not have been possible to complete this work.

Last but not the least I would like to thank God who have always been with me in my good and bad times.

Ravneet Kaur Chawla
(800931017)

Abstract

Cloud computing is a new benchmark towards enterprise application development that can effectively facilitate the execution of workflows in business process management system. The workflow technology can manage the business processes efficiently satisfying the requirements of modern enterprises. A description of a business process is required in the form of a workflow specification to deploy the process as a workflow-based enterprise application. To run a workflow application, workflow management software is required which includes a process editor to support the design of workflow models. Workflows in context of software development are used to rapidly develop the software application improving the software quality.

The part of the thesis work involves the design of a workflow for an online banking application in an open-source workflow system, YAWL (Yet another Workflow Language).With its built-in verification facility, it analyzes and validates the workflow specification according to the YAWL schema. The work also involves how the same workflow specification can be designed in the process perspective of a visual modelling development environment, Orangescape Studio that itself runs as a cloud application. The application with its four-tiered architecture is built on REST-style service interface. The application built can be deployed on some of the major cloud infrastructures offered by Google, Microsoft and IBM.

Table of Contents

List of Figures

List of Tables

CHAPTER-1

INTRODUCTION

This chapter introduces a description of the work presented in the thesis. It gives a brief introduction of Cloud computing, concept behind workflows, workflow management systems and overview of work done.

1.1 Background

The underlying concept of cloud computing dates back to the mainframe days of 1960's when the idea of utility computing was coined by MIT computer scientist John McCarthy. He wrote that "computation may someday be organized as a public utility". Utility computing became a sort of business for companies such as IBM. IBM saw the potential for enormous profit to be made in this type of business and started providing computing services [1]. Figure 1.1 shows the evolution The Cloud from Clustering.

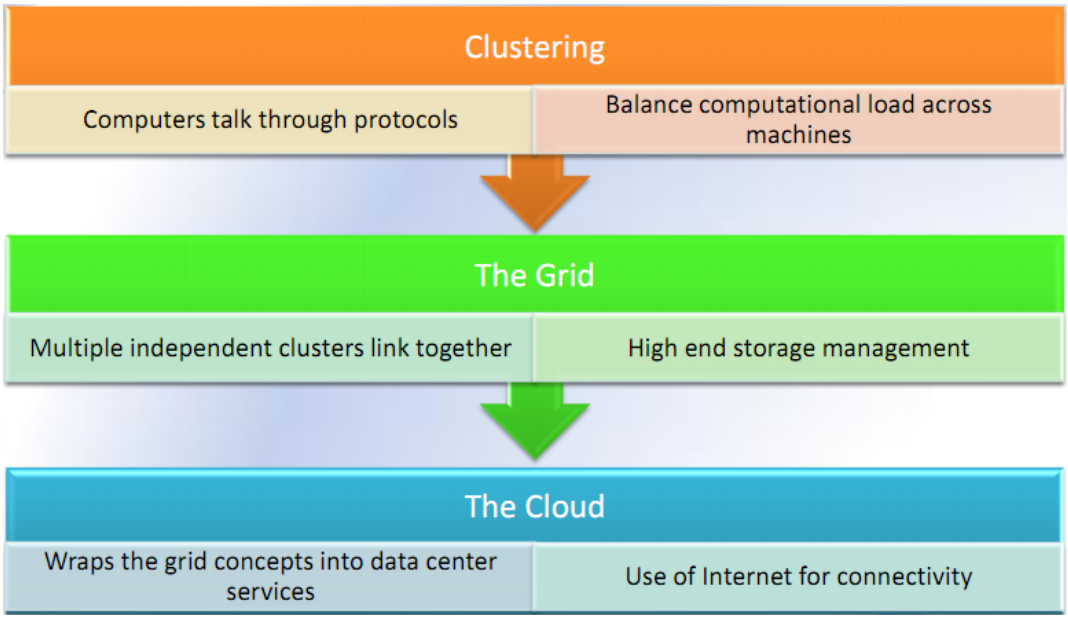

Figure 1.1: Evolution of Cloud Computing from Clustering [1]

In 1980s, the term "workflow" was first used in its modern form in the software industry by FileNet founders Ted Smith and Ed Miller. The company called its business process automation software "WorkFlo".

In 1995, the publishing industry studied how traditional publishing processes could be re-engineered .The term “electronic workflow” was used to describe the publishing process, from online delivery of digital manuscripts to the posting of content on the web for online access [2].

1.2 Concept of Cloud Computing

Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased[3].

According to Gartner, Cloud computing is a style of computing where service is provided across the Internet using different models and layers of abstraction [4]. Cloud Computing refers to the applications delivered as services [5] to the mass, ranging from the end-users hosting their personal documents on the Internet to enterprises outsourcing their entire IT infrastructure to external data centers. Clouds aim to power the next generation data centers by designing them as a network of virtual services (hardware, database, user-interface, application logic) so that users can access and deploy applications from anywhere in the world [6].

Cloud computing typically involves provisioning of dynamically scalable and often virtualized resources [1].Typical cloud computing providers deliver common business applications online accessible 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. The key characteristic of cloud computing is that the processing and the related data is not in a specified, known or static place [1].

Cloud computing is a new paradigm for distributed computing that delivers infrastructure, platform, and software (application) as subscription-based services in a pay-as-you-go model to consumers. When a Cloud is made available in a pay-as-you-go manner to the general public, it becomes a Public Cloud and the service being sold is called as Utility Computing. Current examples of public Utility Computing include Amazon Web Services, Google AppEngine, and Microsoft Azure.

The term “Private Cloud” refers to internal data centers of a business or organization which are not made available to the general public. Thus, Cloud computing is the sum of SaaS and Utility Computing [3].

1.3 Introduction to Workflow

The concept of a business process has been defined by Davenport and Short (1990) as "a set of logically related tasks performed to achieve a defined business outcome". The outcome of a business process can often be described more explicitly as the product, which is created by the process. A business process that delivers services is often referred to as a workflow, service or administrative process [7]. A business process is generally visualized using a workflow as a sequence of activities. A workflow applied to a business process brings the details of that process into focus [8].Workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules. A workflow enables the structuring of applications in a directed acyclic graph form [9], where each node represents the constituent task and edges represent inter-task dependencies of the applications [10].A single workflow generally consists of a set of tasks each of which may communicate with (depends on or is depended upon) another task in the workflow. Multiple workflows have multiple instances of workflow. A final and empirical interpretation is to consider these business processes as workflows that are supported by Workflow Management Systems [7].

1.4 Workflow Management

Workflow management is a fast evolving technology which is increasingly being adopted by businesses. Its primary characteristic is the automation of processes involving combinations of human and machine-based activities, particularly those involving interaction with IT applications and tools. Although it’s most prevalent use is within the office environment in staff intensive operations such as insurance, banking, legal and general administration, etc, it is also applicable to some classes of industrial and manufacturing applications [11].

However, any system that incorporates knowledge about how the business process is executed logistically can be used for a workflow system. Today, Enterprise Resource Planning (ERP) and Customer Relation Management (CRM) systems are incorporating more and more workflow functionality.

A workflow system does not execute any tasks of the business process itself. It focuses on the logistics of the work. In this sense, it is relevant to distinguish between the generic software with which business processes can be managed i.e. WfMS (Workflow Management Systems) and a system that is used to manage a specific business process i.e. a workflow system. Clearly, WfMS can be used to build workflow systems [12].

The main purpose of a workflow management system (WfMS) is to support the definition, execution, registration and control of business processes. It allows organizations to define and control the various activities associated with a business process. Each time an essential piece of work has been completed during a business process execution, the WfMS determines how the business process execution is to be continued by delivering the next piece of work to one or more resources that are capable of executing it. The goal of workflow management is to make sure that the proper activities are executed by the right service at the right time [12].

1.5 Organisation of Thesis

The chapters in this thesis are organized as follows –

Chapter 2 – This chapter describes in detail the literature survey, concept of cloud computing and workflow management system.

Chapter 3 – This chapter describes the problem statement of the thesis. It describes the gap analysis and the need for implementing workflows in cloud environment.

Chapter 4 – This chapter describes the solution of problem, design of solution and implementation of solution.

Chapter 5 – This chapter gives the Experimental results i.e. Snapshots of the deployed application.

Chapter 6 – This chapter describes the conclusion and future work that can be carried based on the work presented in this thesis.

CHAPTER-2

LITERATURE REVIEW

This chapter describes in detail the literature survey, concept of cloud computing and workflow management systems.

2.1 Cloud Computing

With the promotion of the world's leading companies, cloud computing is attracting more and more attention for providing a flexible, on demand computing infrastructure for a number of applications [13]. The name cloud computing is inspired by the cloud symbol representing Internet. Hence, Cloud computing refers to the practice of moving computing to the Internet.

Cloud computing is Internet-based computing, where shared resources, software and information are provided on-demand, like a public utility. Software applications, processing power and data storage are being accessed online. A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned based on service-level agreements established through negotiation between the service provider and consumers [14].

UC Berkeley defines cloud computing to be "the sum of SaaS and Utility Computing". In their view cloud computing has all the advantages of SaaS, including ease of installation and maintenance, access from anywhere and centralized control over versioning [3].Berkeley's researchers moreover limit their definition with several additional assumptions – a) the ability to "pay-as-you-go" as the necessary billing model and b) that users "store their data safely in the infrastructure". Clouds are a large pool of easily usable and accessible virtualised resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing for optimum resource utilisation. Cloud offers many potential advantages such as lower cost, device and location independence, user friendliness [4] which can also benefit workflow systems built on top of a cloud.

Citrix president Mark Templeton said, "Optimization of the user experience will happen in the data centre, at the edge of the network and in internet cloud, allowing IT to deliver any application to any user with the best performance, security and cost savings possible [15].The opinions differ, but the wording in almost all explanations hovers around the keywords scalability, on-demand, pay-as-you-go, self-configuration, self maintenance and Software as a Service.

Companies like Google, Yahoo, Amazon, IBM and Microsoft have built Internet consumer services like search, social networking, Web e-mail and online commerce based on cloud computing [16]. A simple example of cloud computing is Yahoo mail or Gmail. A user does not need software or a server to use them. All a user would need is just an internet connection and one can start sending emails. The server and email management software is all on the cloud (internet) and is totally managed by the cloud service provider like Yahoo, Google, etc.

2.1.1 Features of Cloud Computing

Cloud computing is a technology where dynamically scalable, device-independent and task-centric computing resources are obtained over the Internet on a pay-per-usage basis [17].

- **Cloud computing is dynamically scalable**

Cloud computing is dynamically scalable as users can draw as many or as few computing resources from the cloud as they require at any particular moment. One of the major suppliers of dynamically scalable cloud computing resources is Amazon. Cloud computing developments like Amazon EC2 allow users to purchase any capacity of computing power on an hourly basis. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing users to quickly scale capacity, both up and down, as computing requirements change [17].

- **Cloud computing is device-independent**

Cloud computing resources can be accessed from any computing device on the Internet. The computing device can be a traditional desktop or laptop, a netbook, tablet, smartphone, e-book reader, surface computer, etc [17].

- **Cloud computing is task-centric**

Cloud computing is task-centric because the usage model is based entirely around what users want to achieve, rather than any particular software, hardware or network infrastructure. Users do not have to purchase or install anything before using a cloud computing resource. Also, they do not have to maintain or pay for anything during periods in which no resources are being used [17].

- **Cloud computing has no fixed costs**

In Cloud computing, all costs are on a per-usage or variable basis. Software applications purchased from the cloud incur only variable costs. In case of Amazon EC2, processing power can already be purchased from the cloud on an hourly basis [17].

- **Cloud computing is service-oriented**

The cloud is a natural home for service-oriented applications, which need a way to easily scale as services get incorporated into other applications [18]. Cloud computing is utility-based. Users only pay for the services they use, either by subscription or transaction-based models. Large amounts of computing resources can be provisioned and made available for new applications within minutes instead of days or weeks. Developers can gain access to these resources through a portal and put them to use immediately.

- **Cloud computing is SLA-driven**

Clouds are managed dynamically based on service-level agreements that define policies like delivery parameters, costs and other factors [18].

2.1.2 Architecture of Cloud

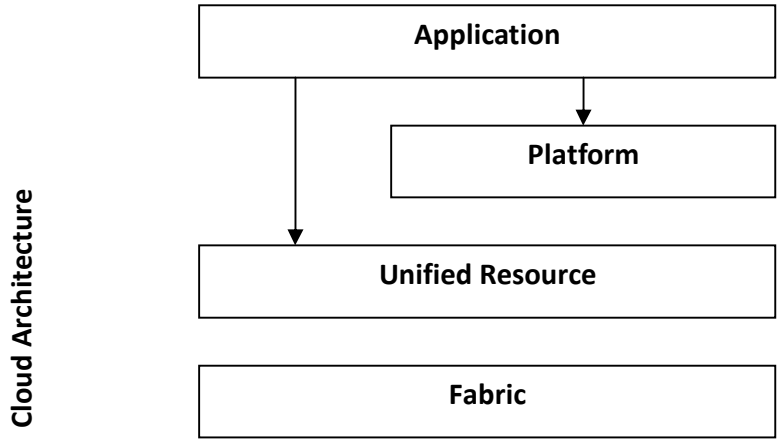

Figure 2.1: Cloud Architecture [19]

The unified resource layer contains resources that have been abstracted/encapsulated (usually by virtualisation) so that they can be exposed to upper layer and end users as integrated resources, for instance, a virtual computer/cluster, a logical file system, a database system, etc. The platform layer adds on a collection of specialised tools, middleware and services on top of the unified resources to provide a development and/or deployment platform, for instance, a Web hosting environment, a scheduling service, etc. Finally, the application layer contains the applications that would run on the cloud [19].

2.1.3 Cloud Computing Stack

The cloud computing stack consists of the following layers:

- **Cloud Infrastructure Layer**

The cloud infrastructure layer provides the core middleware capabilities on a pay-as-you-go basis [20]. The capability is to deliver computing power, network and storage over the network where the consumer is able to deploy and run arbitrary software, including operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications [21]. Infrastructure can scale up and down dynamically based on application resource needs [22]. Although user interface to the cloud infrastructure components varies substantially from one system to another, SOAP and REST are two examples of interface protocols used with some cloud computational resources.

- **Cloud Platform Layer**

The cloud platform layer provides a high-level integrated environment to design, build, test, deploy and update online custom applications [22]. The platform segment of cloud computing refers to products that are used to deploy applications. In the context of cloud computing, Platform as a Service (PaaS) represents an intersection between IaaS and SaaS. It is a form of SaaS that represents a platform provided to serve as the infrastructure for development and running of new applications in the cloud. PaaS offerings are typically presented as API to consumers [20]. A typical advantage is the ability to build and deploy scalable web applications without the cost and complexity of procuring server and setting them up.

Many of the companies that started out providing on demand application services have developed platform services as well. Prominent examples are Google AppEngine and SalesForce.com's business software development platform, Force.com. Google's AppEngine enables customers to build web applications on the same scalable systems which power Google applications. Other examples are Apache Hadoop and Yahoo's Pig [22].

- **Cloud Application Layer**

The cloud application layer consists of SaaS applications developed using the cloud platform services. Software as a Service (SaaS), the term coined by IDC and summarized by SIIA in 2001, initially referred to the Application Service Provider (ASP) model in general, and the shift from desktop/packaged software towards web-based, outsourced solutions in particular[23].Application services are delivered over the network on a subscription and on-demand basis. Using an On-demand service, the end-user pays the software provider a subscription fee for the service. The software is hosted directly from the software providers' servers and is accessed by the end user over the internet [20].Salesforce Customer Relationship Management (CRM) system and GoogleApps are two examples of SaaS. Microsoft's Live Mesh allows files and folders to be shared, and synchronized across multiple devices [22].

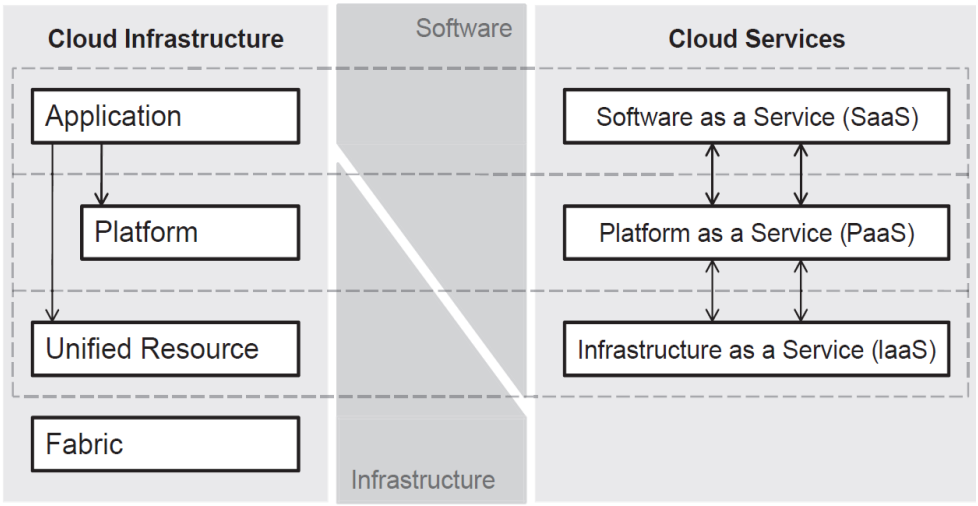

Figure 2.2: Cloud Architecture related to Cloud services [24]

Category	Characteristics	Product Type	Vendors & Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM),Google Documents, GoogleMail
PaaS	Customers are provided with a platform for developing applications hosted in the Cloud.	Programming APIs and frameworks, Deployment system.	Google AppEngine, Microsoft Azure, Manjrasoft Aneka
IaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machines management ,Storage Management	Amazon EC2 and S3, GoGrid.

Table 2.1: Cloud computing services classification [6]

Table 2.1 shows the classification of Cloud computing services based on its characteristics, Product Type and Vendors. Figure 2.2 shows the relationship between Cloud Infrastructure and Cloud Services (SaaS, PaaS and IaaS).

2.1.4 Classification of Clouds

Clouds can be classified according to who the owner of the Cloud data centre is.

Clouds can be classified as:

- **Private Clouds**

The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise [21].

- **Community Clouds**

The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise [21].

- **Public Clouds**

The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services [21].

- **Hybrid Clouds**

The cloud infrastructure is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability [21].

2.1.5 Cloud Computing Platforms

As the computing industry shifts toward providing Platform as a Service (PaaS) and Software as a Service (SaaS) for consumers and enterprises to access on demand regardless of time and location, there will be an increase in the number of cloud platforms available.

- **Amazon Elastic Compute Cloud (EC2)**

Amazon EC2 provides a virtual computing environment that enables a user to run applications. The user can either create a new Amazon Machine Image (AMI) containing the applications, libraries, data and associated configuration settings or select from a library of globally available AMIs. The user then uploads the created or selected AMIs to Amazon Simple Storage Service [14].

- **Google App Engine**

Google AppEngine allows a user to run web applications written using the Python programming language. It also supports Application Programming Interfaces (APIs) for the datastore, Google Accounts, URL fetch, image manipulation and email services. Currently, Google AppEngine is free to use with up to 500MB of storage and about 5 million page views per month [14]. AppEngine is purely a PaaS that supports running restricted versions of Java and Python code with a 30 sec timeout and read-only file system capabilities. Application execution can only be invoked via HTTP requests. Google accounts are required for setting up the applications and could also be used for programming user modules for user authentication.

- **Microsoft Azure**

Microsoft Azure Services is a platform that supports general purpose computing by enabling applications written in a .NET language (C#, VB.NET, J#) to run in the cloud environment managed by the underlying Azure operating system. To support this, Azure has a comprehensive collection of proprietary development tools and protocols which consists of Live Services, Microsoft .NET Services, Microsoft SQL Services, Microsoft SharePoint Services and Microsoft Dynamics CRM Services [14]. The user has access to some functions that could be integrated in their application code, but the user has no control on the underlying Windows Azure OS. Windows Azure acting as the managing OS provide automatic load balancing, geo-replication, application life cycle management and many additional features such as SQL Services, .NET Services, Share Point Services [23].

- **Aneka**

Aneka is a .NET-based service-oriented resource management platform. It is designed to support multiple application models, persistence, security solutions and communication protocols such that the preferred selection can be changed at anytime without affecting an existing Aneka ecosystem. To create an Aneka Cloud, the service provider only needs to start an instance of the configurable Aneka container hosting required services on each selected desktop computer. The purpose of the Aneka container is to initialize services and acts as a single point for interaction with the rest of the Aneka Cloud. Aneka provides SLA support such that the user can specify QoS requirements such as deadline (maximum time period which the application needs to be completed in) and budget (maximum cost that the user is willing to pay for meeting the deadline) [14].

- **Eucalyptus**

Eucalyptus is an open source software framework that implements an IaaS environment. It can deploy private or public clouds and gives users the ability to run and control entire virtual machine instances deployed across a variety of physical resources. The Eucalyptus API is compatible to Amazon EC2 and hence makes it possible to control both Amazon and Eucalyptus instances with the same tools. Its main objectives are to provide a platform for testing applications before they are

moved to Amazon's infrastructure, as well as to manage and control large collections of distributed resources [24]. Eucalyptus, an open source project now supports standardized integration of Amazon's API into Linux distributions for the purpose of building clouds.

- **OpenNebula**

Opposed to Eucalyptus, and currently the only software in the market that describes itself as hybrid cloud toolkit is OpenNebula. OpenNebula transparently integrates external resources in the cloud. OpenNebula also integrates drivers for Amazon EC2 that can be easily extended to support other cloud providers [24].

2.1.6 Associated Technologies

In addition to the above mentioned concepts, theoretical background is also given on Virtualization and Service-oriented Architecture (SOA).

- **Virtualization**

Virtualization has already become a major trend in the IT industry, resulting in large enterprises reaping substantial benefits. Popek and Goldberg proposed in 1974 a now often cited definition of a Virtual Machine (VM) which is an “efficient, isolated duplicate of the real machine” [23]. Virtualization in general implies many tasks related to the abstraction of computer resources. Those may include: OS/Platform virtualization, specific hardware resource virtualization, software application virtualization, virtual application appliance. VMware President and CEO Diane Greene say that the evolution of virtualization began with users deploying virtual machines for testing and development and then easing into server consolidations for production environments. The third phase was resource aggregation, with entire data centers being virtualized, followed by automation of all aggregated workloads [25]. Infrastructure virtualization enables providers such as Amazon to offer virtual hardware for use in data intensive workflow applications.

Platform-as-a-Service (PaaS) clouds expose a higher-level development and runtime environment for building and deploying workflow applications on Cloud infrastructures. Further up in the cloud stack are Software-as-a-Service providers who offer end-users with standardized software solutions that could be integrated into existing workflows.

- **Service-oriented Architecture**

A service comprises of a software component accessible to the users over a network [26]. Service-oriented architecture defines a way to design software to be delivered as a service that communicates using messages over well-defined interfaces. Taking into consideration the present development scenario, most of the applications are built as web services on the principles of SOA paradigm [27]. They are flexible enough to replace the existing services with the new sub-systems as well as allowing the new services to be created by composing other services. A SOA can be implemented through the use of any service based technology, such as SOAP (Simple Object Access Protocol) or REST. For most of the applications to be deployed on cloud, performance plays a key role. Also, the rapid application development and shorter time-to-market has become important for most of the service users.

The REST has the ability to map basic create, update and delete operations to the HTTP (Hypertext Transfer Protocol) commands such as PUT, GET, DELETE and POST. Considering these factors, REST-style web services fits well for most of the applications delivered as services on cloud. In other words, REST is suitable for applications hosted as web service since the basic REST operations are supported by any browser [27].

2.2 Workflow (Business Process)

A business process is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what. Figure 2.3 shows the model for a business process.

A business process:

- Has a Goal
- Has specific inputs
- Has specific outputs
- Uses resources
- Has a number of activities that are performed in some order

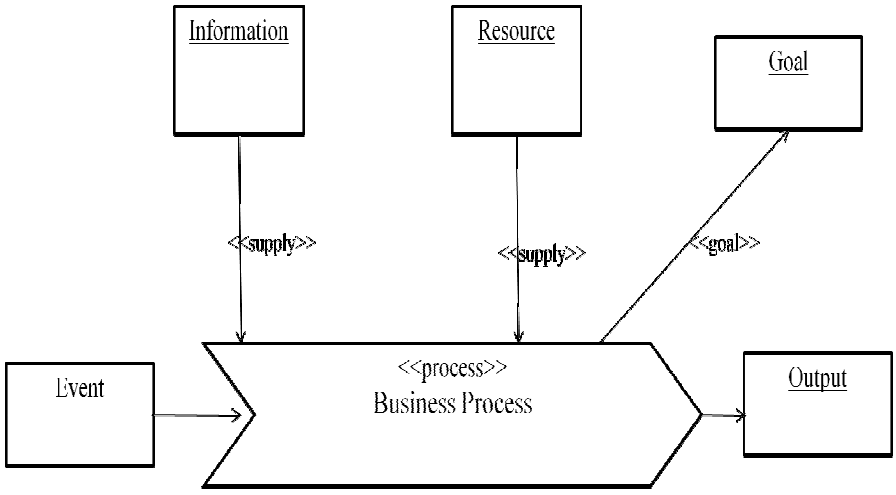

Figure 2.3: The Business Process Model [28]

A business process is generally visualized using a workflow as a sequence of activities. A workflow applied to a business process brings the details of that process into focus. A workflow models a process as consisting of a series of steps that simplifies the complexity of execution and management of applications [29]. Workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of procedural rules.

2.2.1 Workflow Representation

A workflow enables the structuring of applications in a directed acyclic graph form [9], where each node represents the constituent task and edges represent inter-task dependencies such as input and output files, of the applications [10]. Figure 2.4 represents the workflow as a directed acyclic graph. Tasks in a graph are arranged in a hierarchy according to their dependencies. Each task has an outgoing edge connecting to all other tasks that are dependent on it.

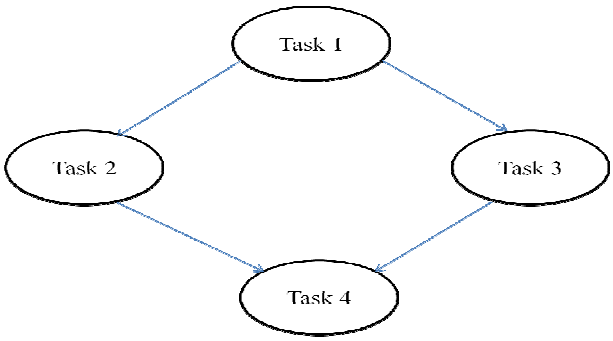

Figure 2.4: DAG Representation of a workflow [30]

A DAG can be divided into levels based on dependencies. Tasks at level 1 are not dependent on any other tasks in the workflow. Each subsequent level is dependent on tasks from the level above it and any previous levels. For example, level 2 is dependent on level 1; level 3 is dependent on levels 1 and 2, and so on. Tasks are executed in an order starting from level 1[30].

2.3 Workflow Management

Workflow management is a fast evolving technology which is increasingly being adopted by businesses. Its primary characteristic is the automation of processes involving combinations of human and machine-based activities, particularly those involving interaction with IT applications and tools. Although it's most prevalent use is within the office environment in staff intensive operations such as insurance, banking, legal and general administration, etc, it is also applicable to some classes of industrial and manufacturing applications [11].A workflow specification defines workflow activities (tasks) and their control and data dependencies. At run time, a workflow enactment engine manages the execution of the workflow by utilizing middleware.

The workflow definition (or workflow schema) defines the sequence of activities that are to be carried out. It specifies what should be done and when by the definition of activities and transitions. An activity represents something to be done: reviewing a document, publishing a document, placing an order, sending an e-mail, and so on. Transitions define the appropriate sequence of activities for a process. A user or system is generally assigned to carry out the activity, through its application.

A Workflow Management System is software providing support for the necessary services of workflow definition and process creation, workflow enactment, administration and monitoring of workflow processes. Figure 2.5 illustrates the basic characteristics of WfM systems and the relationships between these main functions [33]. A workflow management system defines, manages and executes workflows on computing resources [32]. The goal of workflow management is to make sure that the proper activities are executed by the right service at the right time [12].

Workflow scheduling is one of the key issues in the management of workflow execution. Scheduling is a process that maps and manages execution of inter-dependent tasks on distributed resources.

It introduces allocating suitable resources to workflow tasks so that the execution can be completed to satisfy objective functions specified by users. Proper scheduling can have significant impact on the performance of the system [31].

Workflow management systems are the software systems that enable workflow management. There are a variety of software systems available for planning and executing workflow applications, including: YAWL, jBPM, Intalio, ProcessMaker, BonitaSoft, GoFlow and EnhydraShark. YAWL is a good example of one of these workflow management systems.

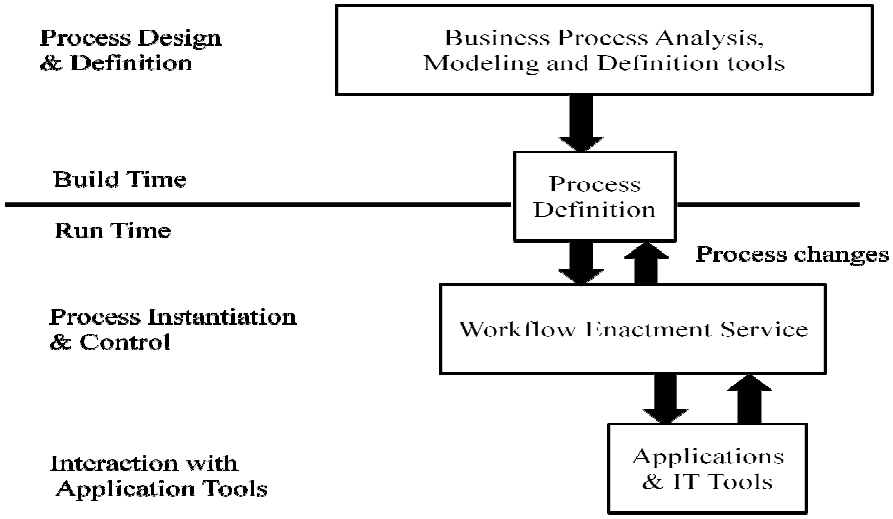

Figure 2.5: Workflow System characteristics [33]

2.4 Benefits of implementing workflows in cloud

Moving workflows to a cloud computing environment enables the utilisation of various cloud services to facilitate workflow execution. The primary benefit of moving to clouds is application scalability. Unlike grids, scalability of cloud resources allows real-time provisioning of resources to meet application requirements at runtime. The elastic nature of clouds facilitates changing of resource quantities and characteristics to vary at runtime, thus dynamically scaling up when there is a greater need and scaling down when the demand is low. This enables workflow management systems to readily meet quality-of-service (QoS) requirements of applications. The "user-centric" model in a cloud computing environment makes workflow execution more user-friendly thus increasing user satisfaction. The "pay as you go" business mode in a cloud can reduce the execution cost of workflows.

Cloud providers also take advantage of economies of scale, providing compute, storage and bandwidth resources at substantially lower costs. Thus utilizing public cloud services could be economical as compared to the more expensive dedicated resources [29].

Figure 2.6 illustrates how a client request for executing a business process (workflow) is satisfied by the orchestration layer by invoking two different services provided by cloud service providers [34].

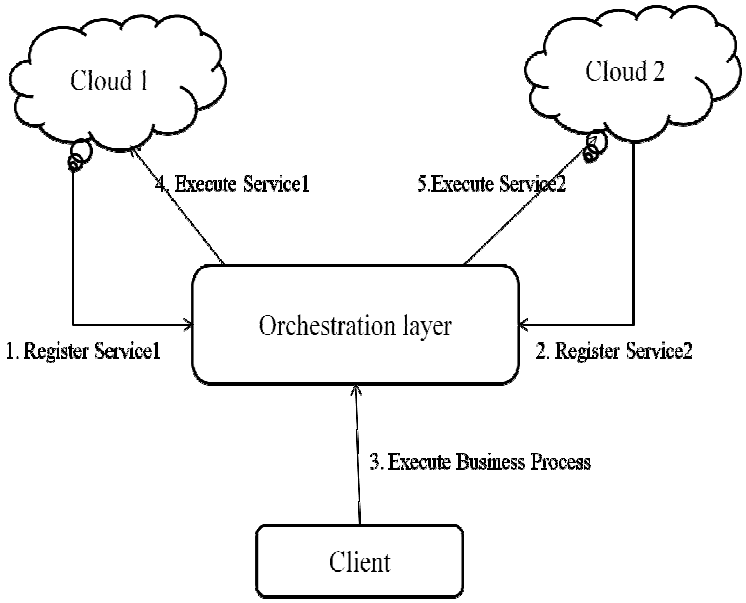

Figure 2.6: Cloud Orchestration [34]

The advantages of moving workflows to cloud can be summarized as:

- **Illusion of infinite resources**

Unlike grids, clouds give the illusion that the available computing resources are unlimited. This means that users can request and are likely to obtain sufficient resources at any given time.

- **Leases**

In grids and clusters, the user specifies the amount of time required for a computation. In clouds, the user directly allocates resources as required to schedule their computations. This model is ideal for workflows and other loosely-coupled applications because it decreases the scheduling overheads.

- **Elasticity**

Clouds allow users to acquire and release resources on-demand. This enables workflow systems to easily grow and shrink the available resource pool as the needs of the workflow changes over time [29].For these reasons there is a need to migrate workflow executions to cloud computing platforms.

2.5 Challenges of implementing workflows in cloud

The challenges with such an approach are as follows:

- **Cost**

Although in reality some cloud services say, Google searching is free, some cloud services for execution of business workflow applications are not. This means users may need to pay for what they use. Cost becomes a major concern for cloud workflow applications. Given that the attributes of both time and cost are involved, users may decide on the fly to pay slightly more to reduce execution time, or save execution cost by allowing longer execution time as long as the final deadline can be met for workflow execution [16].

- **Service level agreements**

With most cloud computing services coming from large commercial organizations, service level agreements have been an important concern to both the service providers and consumers. Due to competitions within emerging service providers, greater care is being taken in designing SLAs that seek to offer (a) better QoS guarantees to customers and (b) clear terms for compensation in the event of violation.

- **Data volumes**

Depending on the provided service, the data volumes required to be transported across cloud services is another important factor to be considered.

- **Platform support**

Depending on the service, the platform support required by the cloud application service could also be a limiting factor. Apart from the above, there could be other challenges like security, regulatory compliance, data transparency [34].

CHAPTER-3

PROBLEM STATEMENT

3.1 Gap Analysis

In a collaborative environment, different application development teams are situated at different places of the world. When working on a single application, it is difficult to maintain versions of application modules. To solve this, applications need to be deployed in cloud environment.

Many workflow applications are data-intensive and processing and managing such large amounts of data requires the use of a distributed collection of computation and storage facilities. In the current development scenario, the workflow management requires the use of database servers. The coordination between workflow scheduling and database management requires more accurate management of service usage and accurate estimation of service load. So, workflows for a business application need to be implemented in cloud environments.

Since it is difficult to integrate most of the workflow management systems with any of the cloud platforms to carry out the deployment of applications, there is a need to follow an approach that can easily integrate workflows with the cloud. There is a need to follow an approach that can easily port applications built for one cloud platform to another. Applications should be built once and deployed several times on any cloud platform as per user's requirements. Hence, the aim is to design a workflow-based application and deploy it in cloud using a visual modeling development environment.

3.2 Need for implementing workflows in cloud

Initially, workflows were being implemented in grids. Due to the reduced performance faced in grids, now there is a need to implement workflows in cloud. The recent progress in virtualization technologies and the rapid growth of Cloud computing services have opened a new paradigm in distributed computing for utilizing existing resource pools for on demand and scalable scientific computing.

The primary benefit of moving to Clouds is application scalability. Unlike Grids, scalability of Cloud resources allows real-time provisioning of resources to meet application requirements. The elastic nature of clouds facilitates dynamically scaling up when there is a greater need for additional resources and scaling down when the demand is low. This enables workflow management systems to readily meet Quality-of-Service (QoS) requirements of applications, as opposed to the traditional approach that required advance reservation of resources in global multi-user Grid environments. Cloud services like compute, storage and bandwidth resources are available at substantially lower costs. Thus utilizing public cloud services could be economical as compared to the more expensive dedicated resources. Workflow applications often require very complex execution environments that include specific operating systems, libraries, file system structures and numerous application programs and configuration files. These environments are difficult to create on grid resources. In addition, each grid site has a different configuration, which results in extra effort each time an application needs to be ported to a new site. Virtual machines allow the application developer to create a fully customized, portable execution environment configured specifically for their application.

CHAPTER-4

SOLUTION OF PROBLEM

4.1 Design of Solution

This section gives an overview of the workflow-based Online banking Application and shows how it can be designed in an open-source workflow management system, YAWL.

4.1.1 Example Application

Online banking is a service where people can perform financial transactions such as account to account transfers, deposit of cash and withdrawal of cash and so on. In our demo system, the security is provided by giving a Personal Identification Number (PIN) to every customer to access his/her account. The system checks whether the PIN entered by the user is valid or not. It also takes care of the remaining balance in the customer's account. If the amount entered by the user to withdraw the cash exceeds the current balance, the system gives a warning message else it allows the user to withdraw the cash. Figure 4.1 shows a graphical overview of the functionality for an online banking application in terms of actors, their goals (represented as use cases) and dependencies between the use cases.

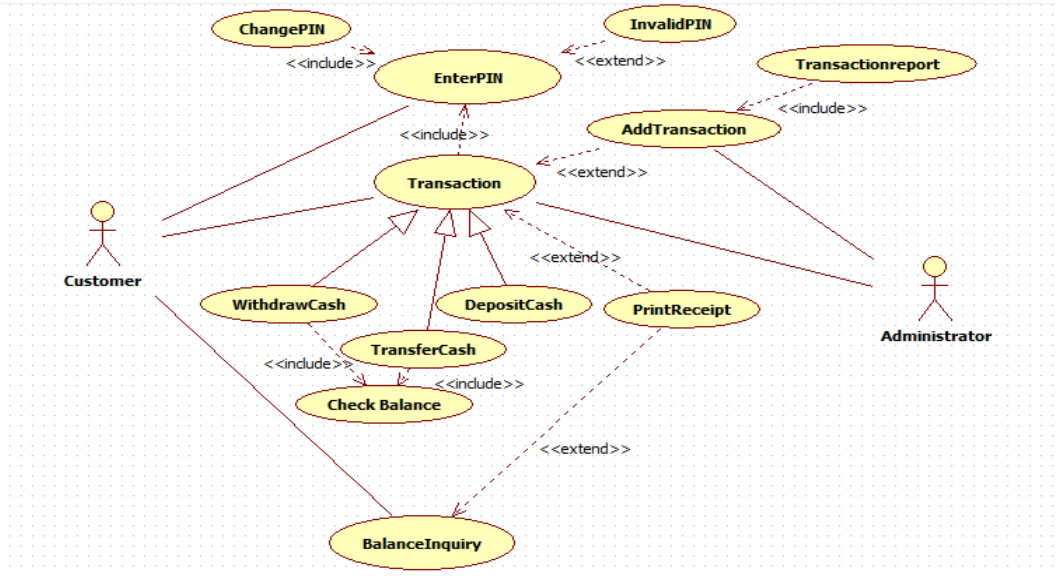

Figure 4.1: Use case diagram for Online banking application

4.2 Tools used for the design of application

This section introduces the Workflow Management System, YAWL in designing the workflow for the example application. Further, it describes a visual modelling development environment, OrangeScape Studio which is used to design and deploy the complete application on cloud platform, Google AppEngine.

4.2.1 YAWL (Yet Another Workflow Language)

YAWL (Yet another Workflow Language) started as a language focusing on the control-flow (process) perspective. Later the attention shifted to include other perspectives and a workflow management system supporting YAWL was implemented [35].

YAWL [36], an open source workflow management system is chosen as a start point for its formal definition language and extensible service-oriented architecture. YAWL is proposed as a variant of Petri Net and expresses workflow patterns more explicitly. Petri Net is a mathematical modelling language widely used in describing distributed systems and is deemed for process modelling. Compared to other process modelling languages, it has exact mathematical definition of execution semantics and hence well-formed theories in process analysis.

YAWL is composed of YAWL engine, YAWL designer, YAWL manager and YAWL services [37]. A workflow specification in YAWL is a set of process definitions which form a hierarchy. Each process definition consists of tasks (either composite or atomic) and conditions which can be interpreted as decision points. Each process definition has one unique input condition and one unique output condition.

Workflow specifications are designed using the YAWL designer and deployed into the YAWL engine which, after performing all necessary verifications and task registrations, stores these specifications in the YAWL repository. Once successfully deployed, workflow specifications can be instantiated through the YAWL engine, leading to workflow instances (cases). The engine handles the execution based on the state of a case and its specification determining which events it should offer to the environment[35].YAWL services provide the interfaces for clients to access to YAWL system [37].Figure 4.2 shows the graphical symbols used to design a workflow in YAWL workflow system.

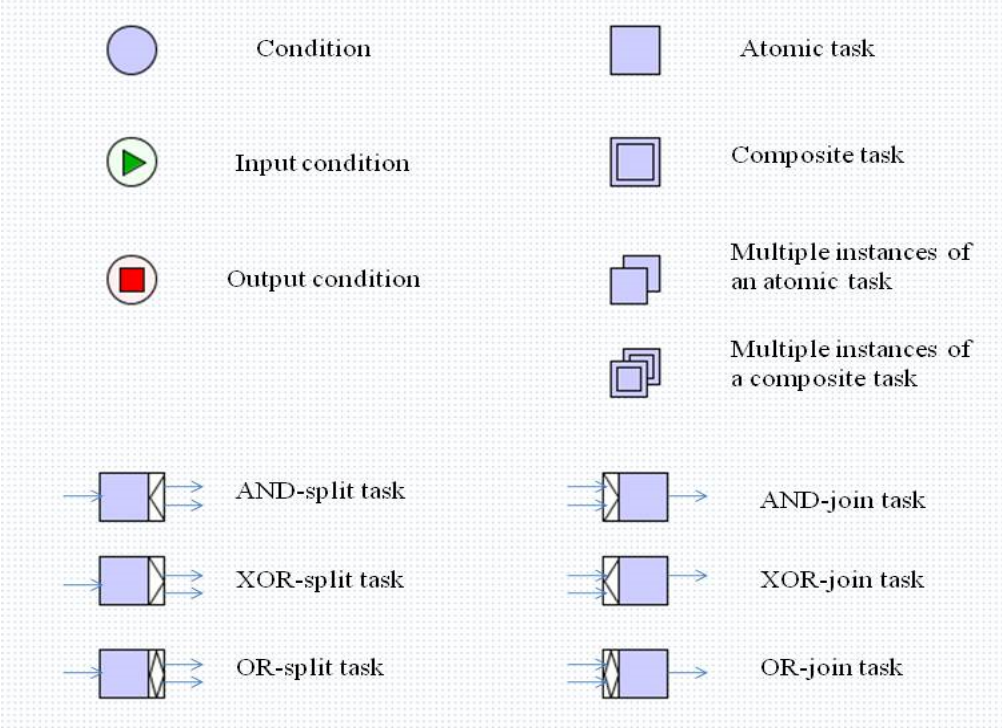

Figure 4.2: Symbols used in YAWL

Designing the workflow

The workflow for the online banking application can be designed in YAWL process editor.

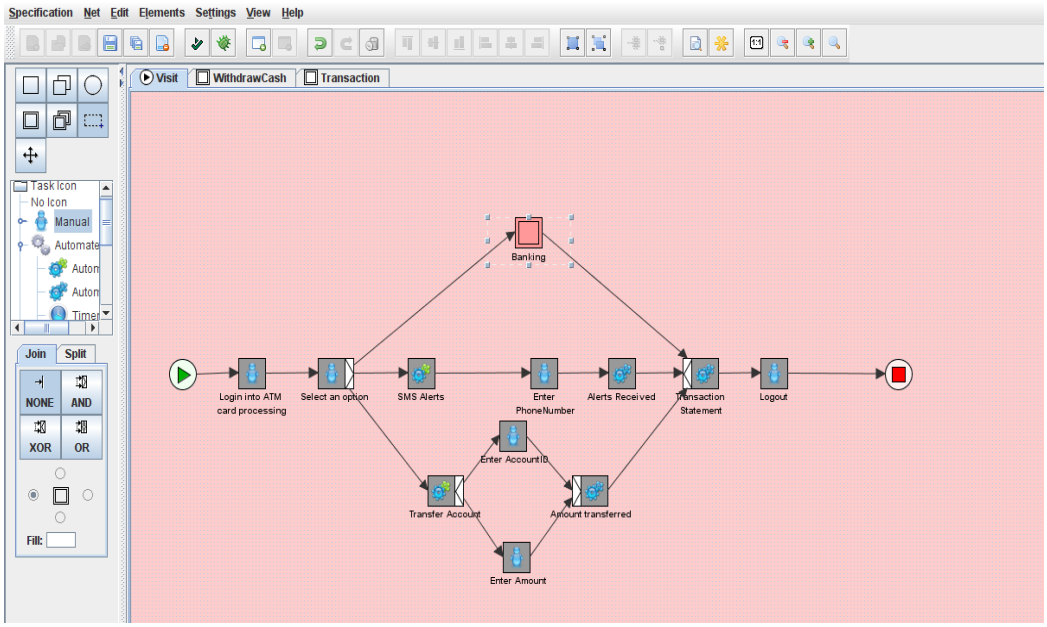

Figure 4.3: Workflow specification for login into ATM Card processing

The workflow specification is created by clicking the "New Specification" under Specification tab. The workflow specifications are also referred as Net specification. Figure 4.3-4.5 shows the workflow specifications for the example application.

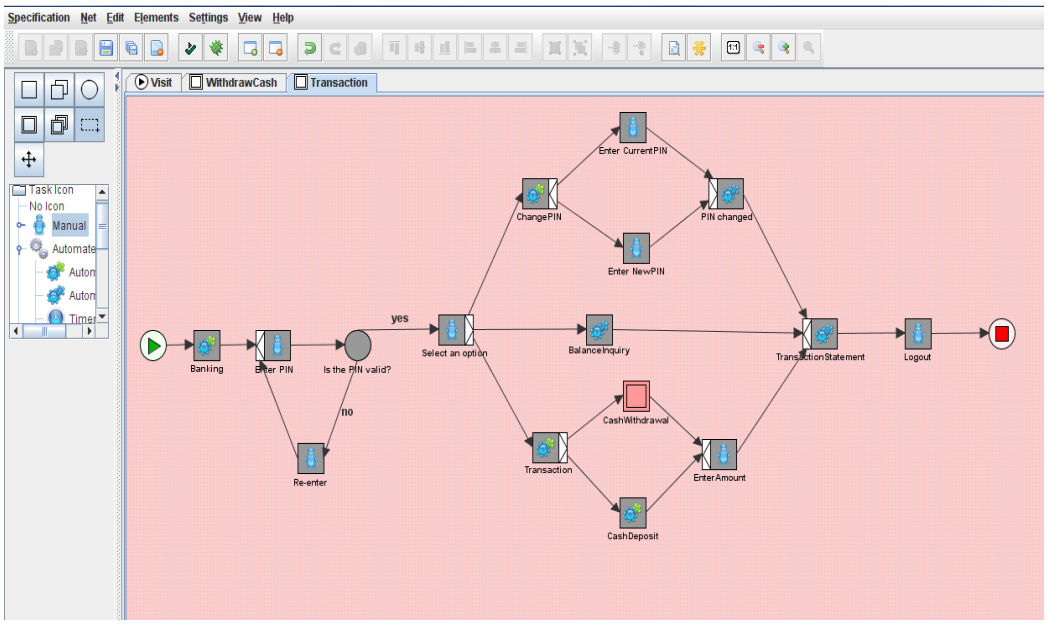

Figure 4.4: Workflow specification for Transaction processing

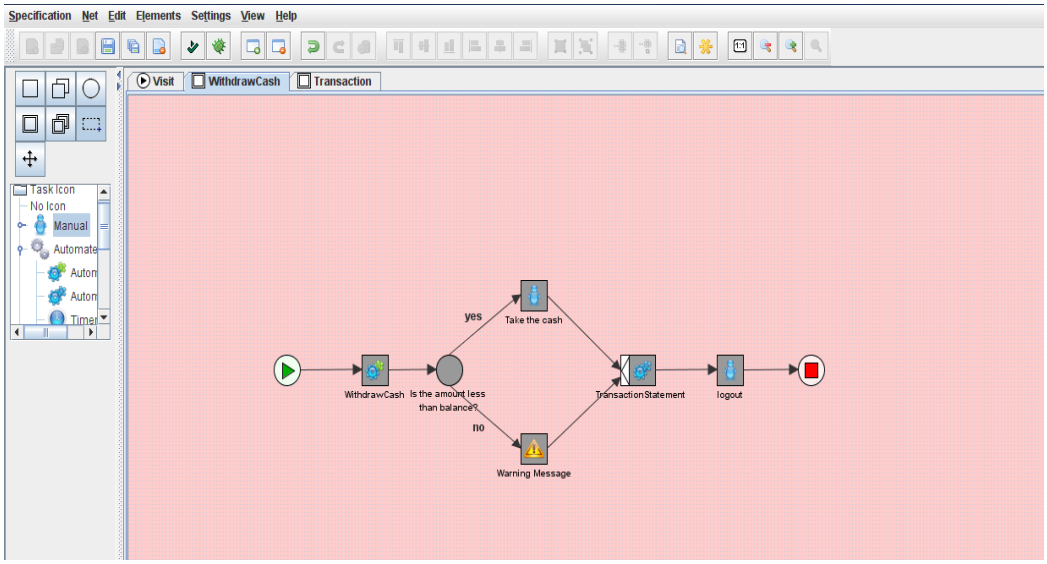

Figure 4.5: Workflow specification for Cash withdrawal

4.2.2 OrangeScape Studio

Orangescape Studio is ideal for designing process-oriented business applications. In the current scenario, the process is implemented in a visual modelling environment by creating workflows, a representation of a process. The platform is flexible and scalable enough for managing the workflow processes in different business domains. To simplify business application development, it enables the users to build data model, business processes and rules using a familiar spreadsheet like interface without writing a single line of code. The visual background provides a complete version control and test environment for the applications. The platform itself runs as a cloud application offering techno-functional experts a service to convert their ideas into business logic by expressing logical rules. Each object in the OrangeScape Virtual Machine abstracts persistence, business process, rule behaviour, procedural behaviour, authorization & Service-oriented Architecture (SOA) serialization.

System requirements

- Mozilla Firefox
- Microsoft Silverlight

Mozilla Firefox is a free and open source web browser managed by Mozilla Corporation. To display web pages, Firefox uses the Gecko layout engine, which implements most current web standards in addition to several features. The latest Firefox features include tabbed browsing, spell checking, incremental find, live bookmarking, a download manager, private browsing. Firefox runs on various operating systems including Microsoft Windows, Linux, Mac OS, FreeBSD and many other platforms.

Microsoft Silverlight is an application framework for writing and running browser plug-ins or other rich internet applications with features similar to those of Adobe Flash. The run-time environment for Silverlight is available as a plug-in for most web browsers. The current versions support multimedia, graphics and animation and give developers support for development tools. It is compatible with multiple web browsers used on Microsoft Windows and Mac OS.

The other pre-requisites required to use OrangeScape development environment are: an account on OrangeScape Trial Studio to build applications and an account on Google AppEngine to run applications. Users having Google account, GoogleApps account or Yahoo account can log into OrangeScape Trial Studio.

Creating an application

The first step in building an application using Studio is to create a project workspace for the application. Designing a process-oriented application involves: defining the data and logical rules, designing a user format/interface and establishing a process workflow for data processing. The current environment follows the bottom-up design approach for rapid application development through four perspectives-Model design, Form design, Process design and Action design. These perspectives fully support data model and rule design (business rules), drag-and-drop form design, graphical process design and visual designer for modelling integration and procedural logic.

- **Data modelling as the start point**

The application development centers on the data model. Data modelling takes care of what data needs to be stored by the application and then creating data structures accordingly. Instead of representing a technical database design, the data model organizes the data by modelling real world entities along with their attributes. The creation of a data model diagram clearly visualise the entities and their relationships.

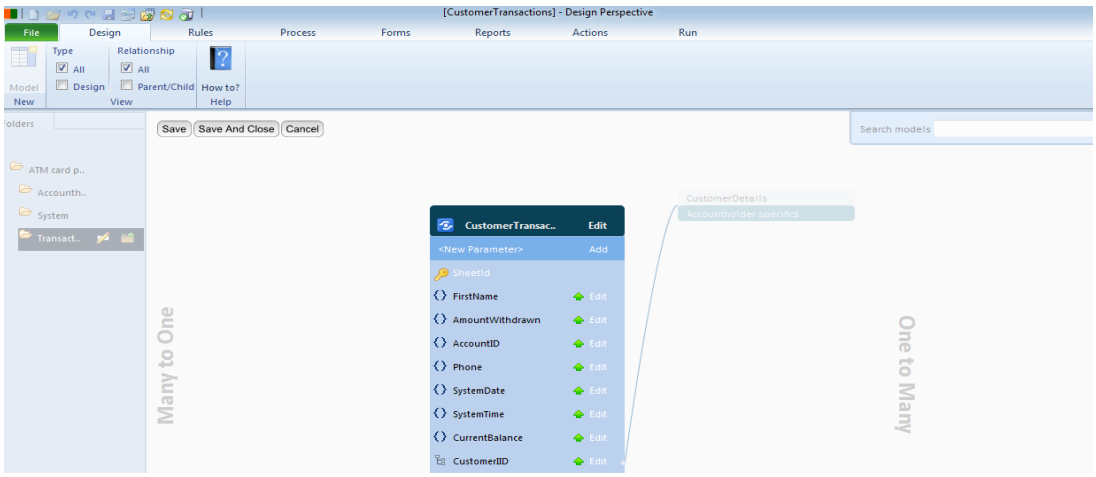

Figure 4.6: Creating a data model

Figure 4.6 shows the attributes of the entity "Customer" and the relationship between two data models namely, Customer Transactions and Customer Details for our example application.

The attributes for the Customer entity are namely, CustomerID, FirstName, Phone, CurrentBalance, AccountID and AmountWithdrawn.

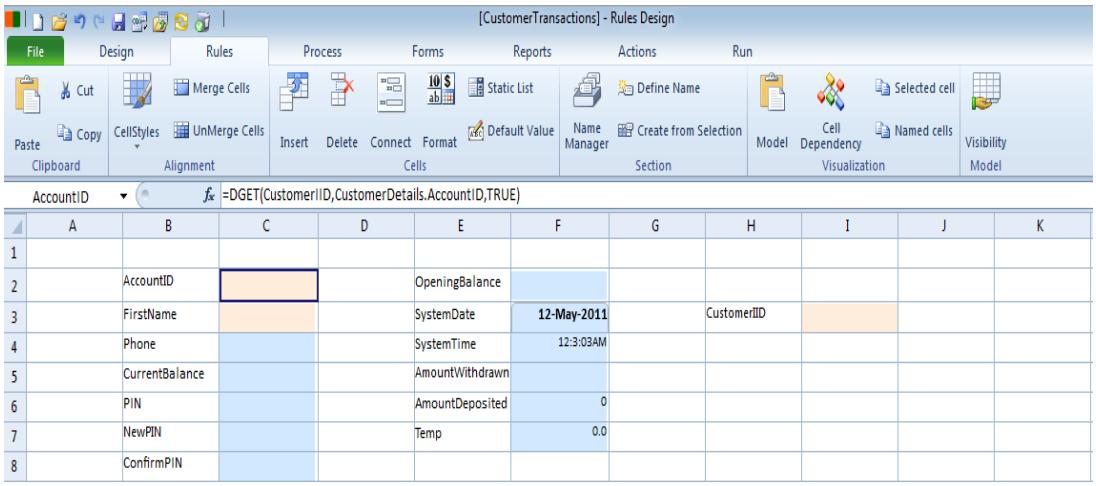

Figure 4.7: Specifying rules for data model

The spreadsheet based interface allows the developers to define the complex business rules to implement the application logic. Figure 4.7 shows the spreadsheet based interface for the example application. An effective data model ensures that a system built on its foundation does not need to be reconstructed to accommodate new business requirements. The approach follows with data modelling as a start point and progresses to reach form and process design.

- **Designing user interface screens**

Once the data modelling completes, the next step identifies, defines and designs the user interface elements. The user interface wire-frame links directly to the data model and workflow state. Form designer is a grid based page builder for defining forms and form elements such as display fields and input fields. It includes a point-and-click selector to validate the field elements, specify permissions and actions. Permissions are expressed as static or dynamic based on the expressions. The actions defined can be mapped to forms through buttons. Depending on the workflow state, interface design can be customized such as including parameters from other models that are related to the current model as well as enabling or disabling form field elements.

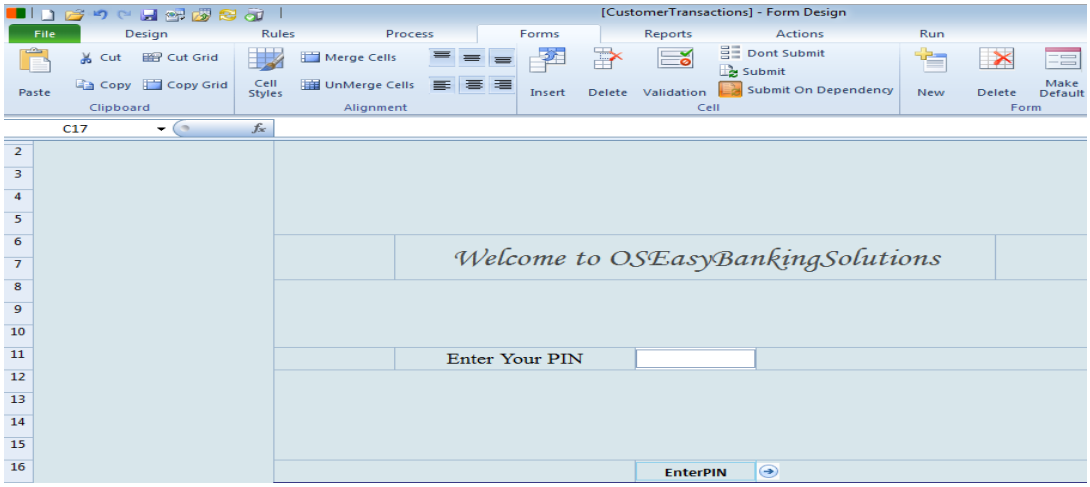

Figure 4.8: Form design for entering the Customer's PIN

Figure 4.8-4.12 shows the design for the forms displayed as user interface screens when the complete application is deployed on Google AppEngine. The user who accesses the application enters his/her PIN number. After the PIN number is validated by the system, the user is asked to select an option: to change the PIN number, to inquiry account balance or to perform transactions. The user can withdraw as much amount of cash as he/she wants provided the amount entered is less than the remaining balance in the account. Similarly, the user can perform other transactions like to deposit the cash or transfer the cash from one account to another.

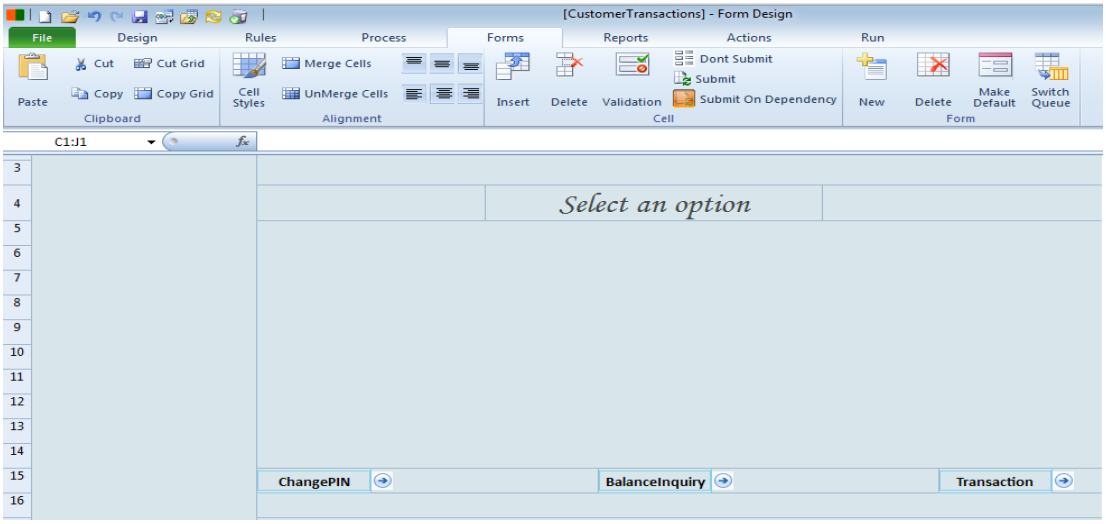

Figure 4.9: Form design for selecting an option after the PIN verification

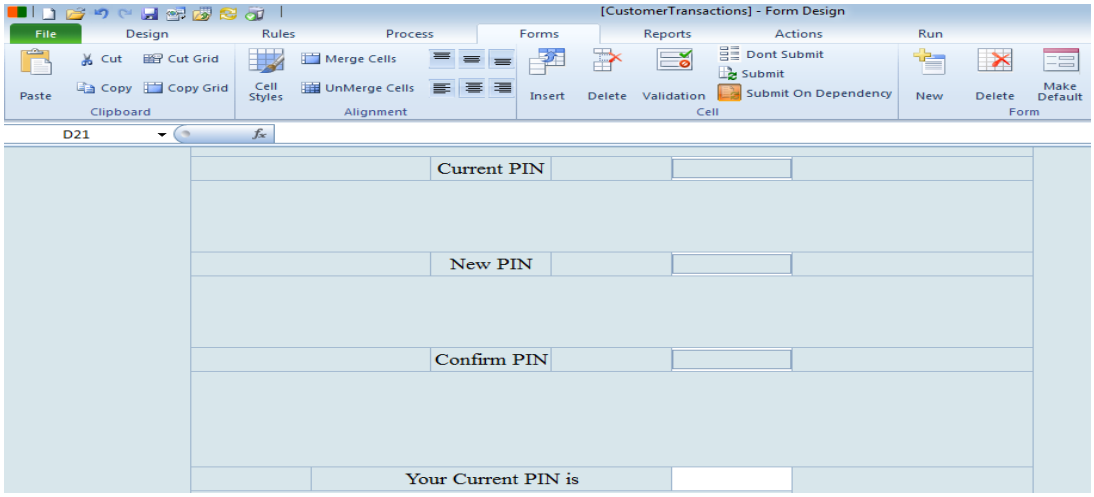

Figure 4.10: Form design for changing the PIN number

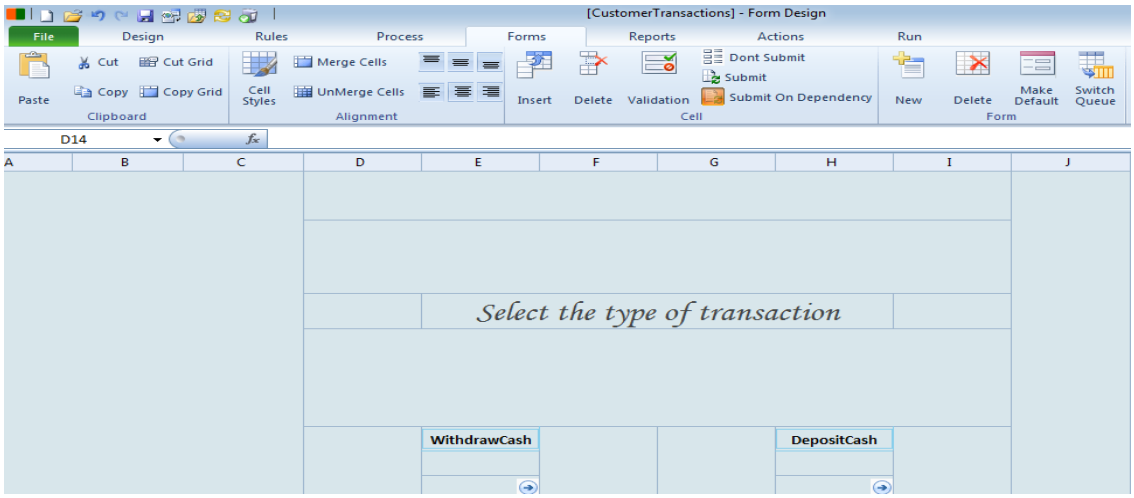

Figure 4.11: Form design for selecting the type of transaction

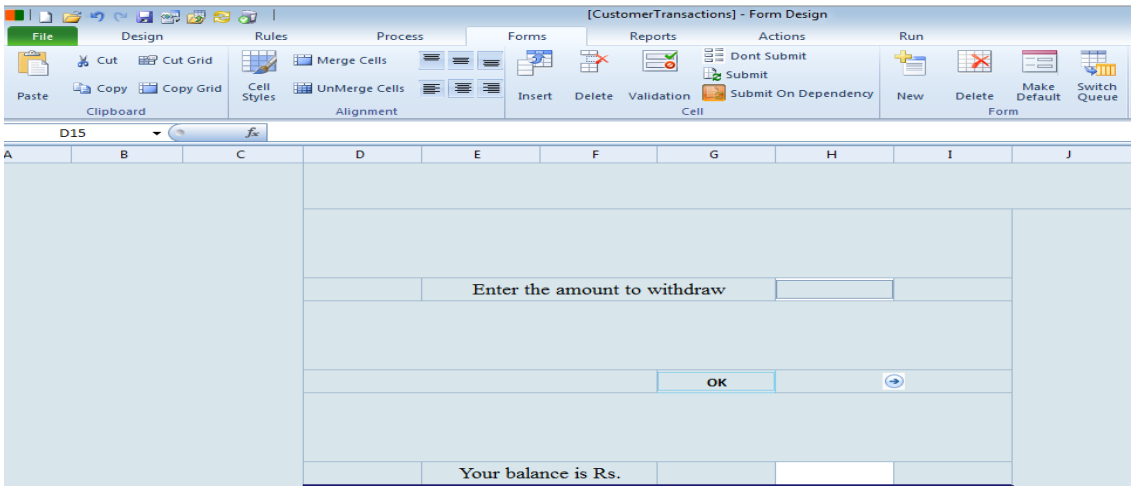

Figure 4.12: Form design to withdraw cash from Customer's account

- **Workflow modelling**

A business process can be thought of as a set of interrelated activities that produces a specific service. It is generally visualized using a workflow as a sequence of activities. A workflow applied to a business process brings the details of that process into focus. In our approach, workflows are designed using a visual modelling environment in the process design.

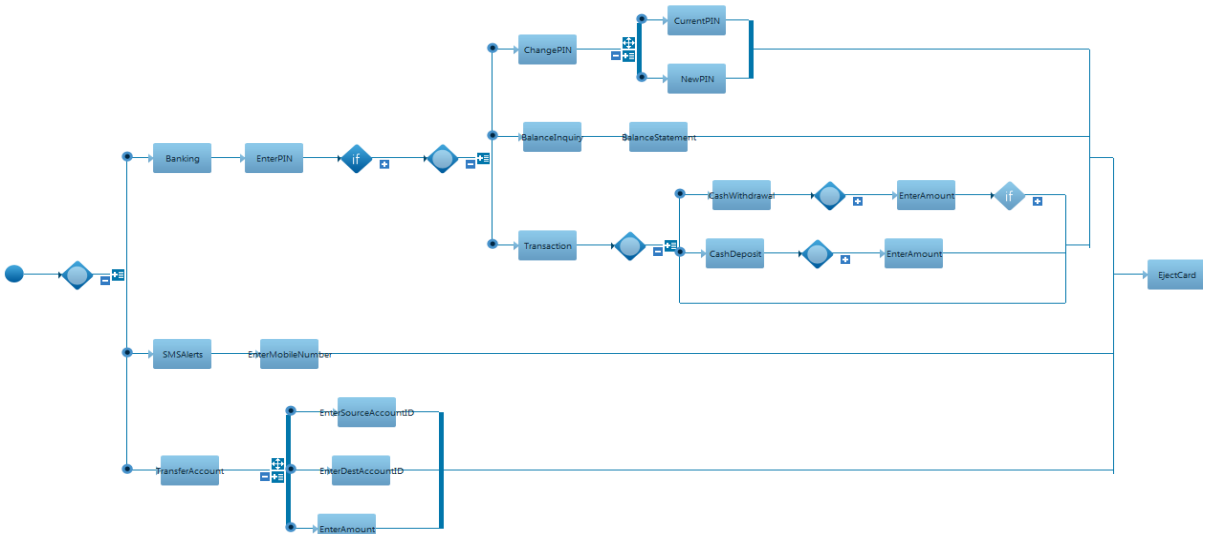

Figure 4.13: Design of workflow for the example application

The advanced workflow features includes branching (conditional or parallel), routing, association of various forms and users to activity states, linking of routing logic with the complex business rules .Figure 4.13 shows the workflow design for the example application.

- **Defining automation tasks**

Action design perspective can be visualized as a typical procedural programming structure. Action designer includes a set of predefined commands such as updating the current entity, creating a loop which breaks on a condition, searching and calling parent/child entity, sending mail, calling an external web service. The "update" command updates the current model entities, while the "submit" command submits an instance from current workflow activity to the next activity.

The commands when graphically arranged create more complex commands. The created composite command maps to a button within the process design for automated execution. Figure 4.14 shows the update and submit commands in the action design perspective of the example application. This perspective defines automation tasks (such as email notification) and integration with external systems using REST.

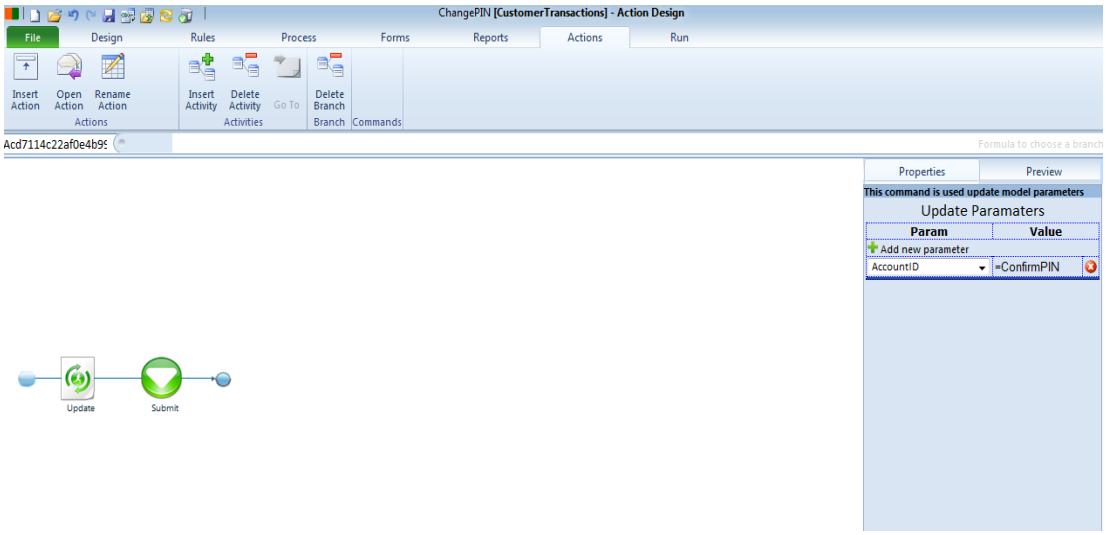

Figure 4.14: Update and Submit commands in Action design

CHAPTER-5

EXPERIMENTAL RESULTS

This section shows the analysis of the workflow specification designed in YAWL as well as the snapshots for the example application deployed on Google AppEngine cloud platform.

5.1 Analysis of the workflow design

Once the workflow is designed in workflow management system, YAWL, the next most important step is to analyse the workflow specification. At any stage, the specification can be validated and/or saved to a YAWL engine file (.yawl). By clicking the validation button on the left corner of the workflow editor, the specification can be tested against YAWL schema. If problems are detected, a table listing them will appear at the bottom of the Editor with details of any inconsistencies that would stop the specification from running in the YAWL engine. Figure 5.1 shows the specification for the example application with no validation problems

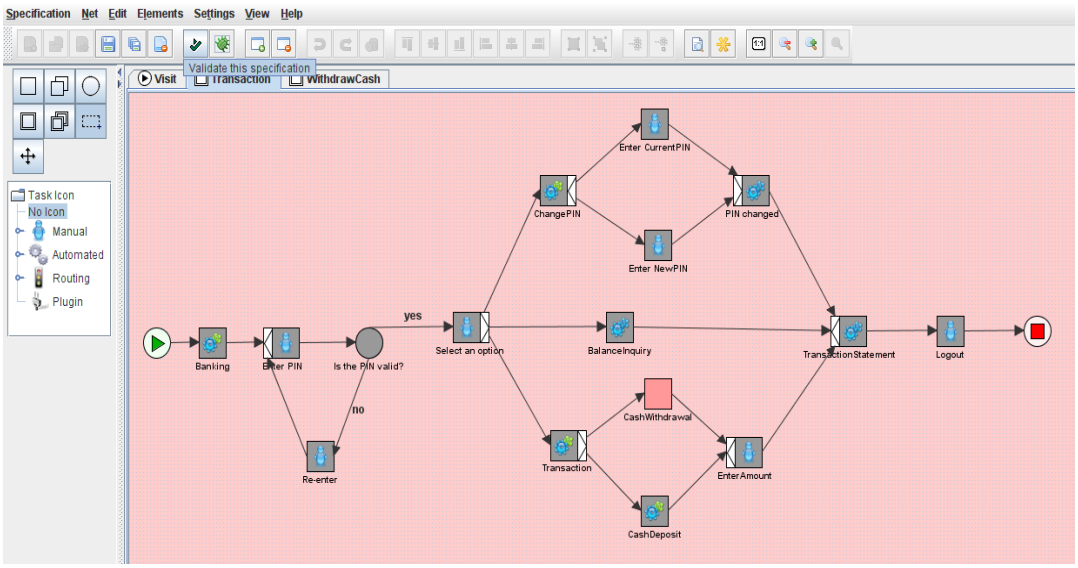

Figure 5.1: Validating the workflow specification against YAWL syntax

The analysis toolbar button on the left corner of the editor or the matching Analyse Specification menu item under the Specification menu allow workflow designers to analyse the specifications.

A number of potential problems with a workflow can be automatically spotted with analysis. Examples include spotting potential deadlock situations, unnecessary or-join decorators. Figure 5.2 shows the analysis of the workflow specification for the example application.

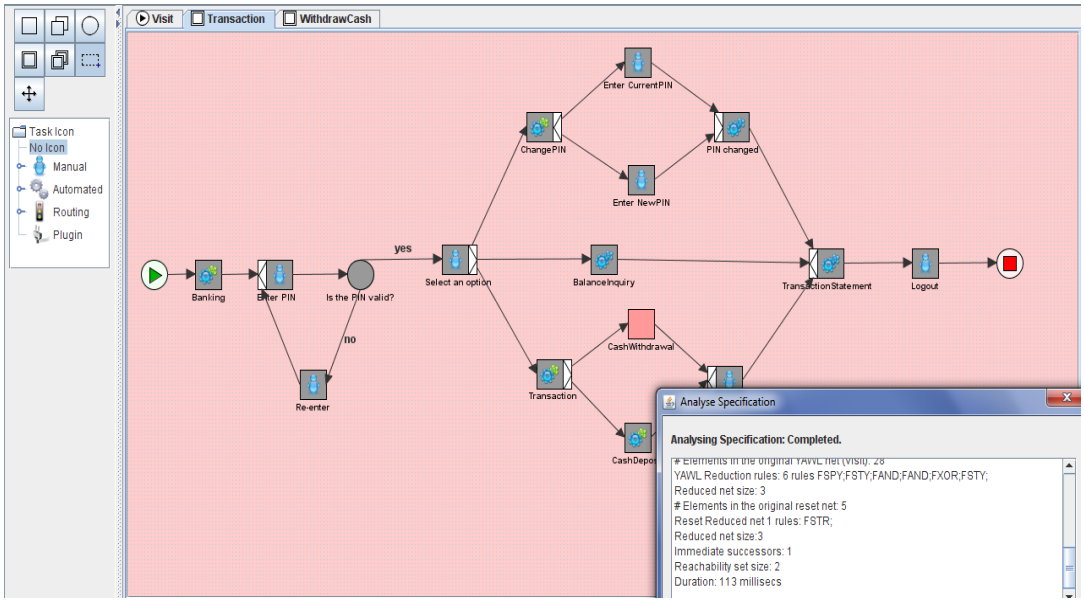

Figure 5.2: Analysing the workflow specification in YAWL

5.2 Application deployment

The packaged application can be deployed on Google AppEngine by using a single-click deploy option in-built in the Studio environment.

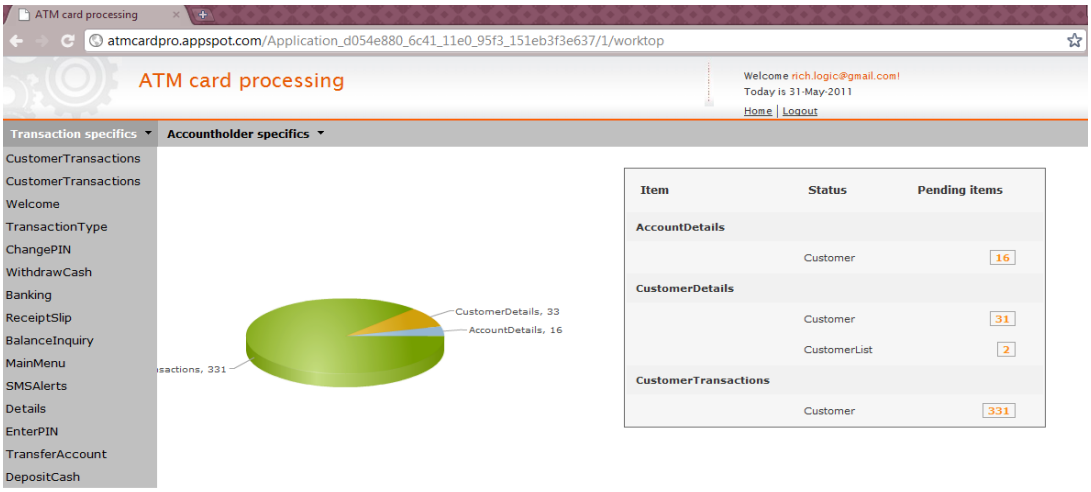

Figure 5.3: Example application deployed on Google AppEngine

Figure 5.3 shows the example application deployed on Google AppEngine. The various user interface screens can be viewed under the folder "Transaction specifics". When the user first starts to access the application, a welcome screen is displayed. Figure 5.4 shows the welcome screen of the application. On clicking the "OK" button, the user is redirected to another user interface screen, login page of the application. Here, the user enters his/her CustomerID and the system displays the name of the customer accessing the application. Figure 5.5 shows the login interface screen for the example application.

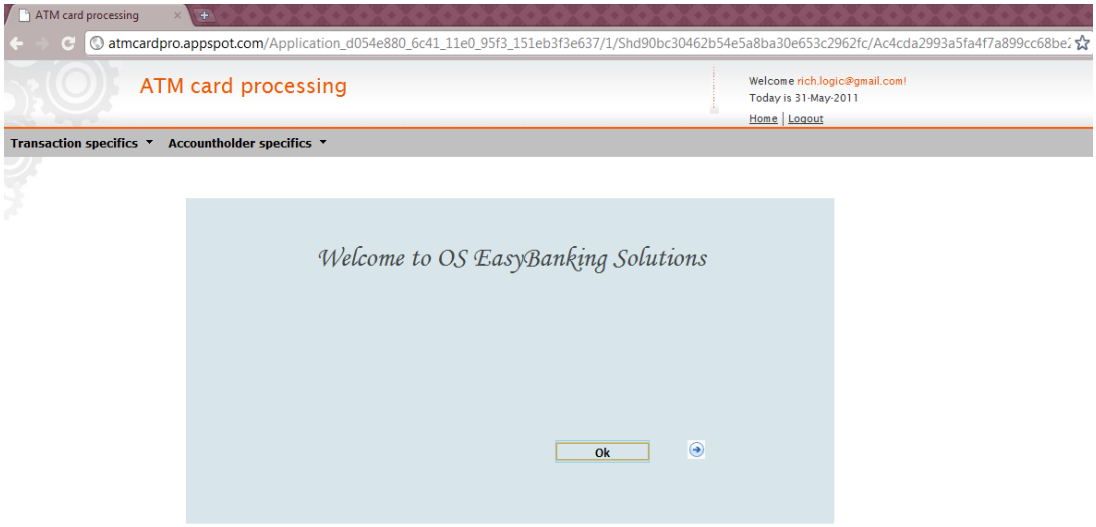

Figure 5.4: Welcome screen of the application

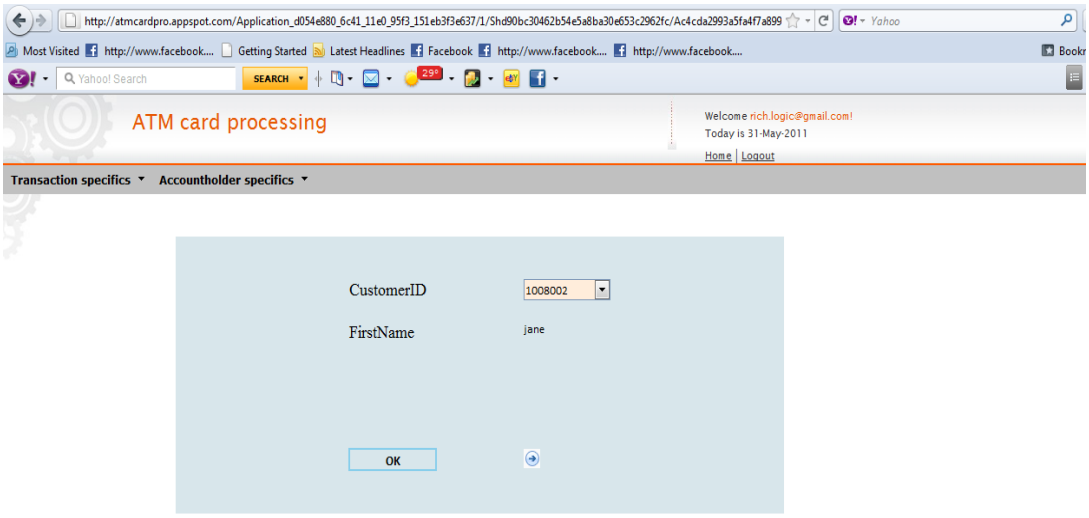

Figure 5.5: Customer login screen of the application

When the user clicks on “OK” button, another screen appears which enables the user to select an option. The user can opt for banking transactions, can subscribe for SMS Alerts and even can transfer cash from one account to another. Figure 5.6 shows a user interface screen to opt for any one of these options. After the user selects an option, the system asks the user to enter his/her PIN number to proceed for further transactions. Figure 5.7 shows the interface screen for the user to enter the PIN number. The system checks if the value entered is correct or not. If the value exactly matches with the Customer’s PIN value stored in the system server, the user is allowed to proceed further. Otherwise, the system asks the user to re-enter the PIN.

Figure 5.6: Selecting an option for Banking, SMS alerts and Transfer account

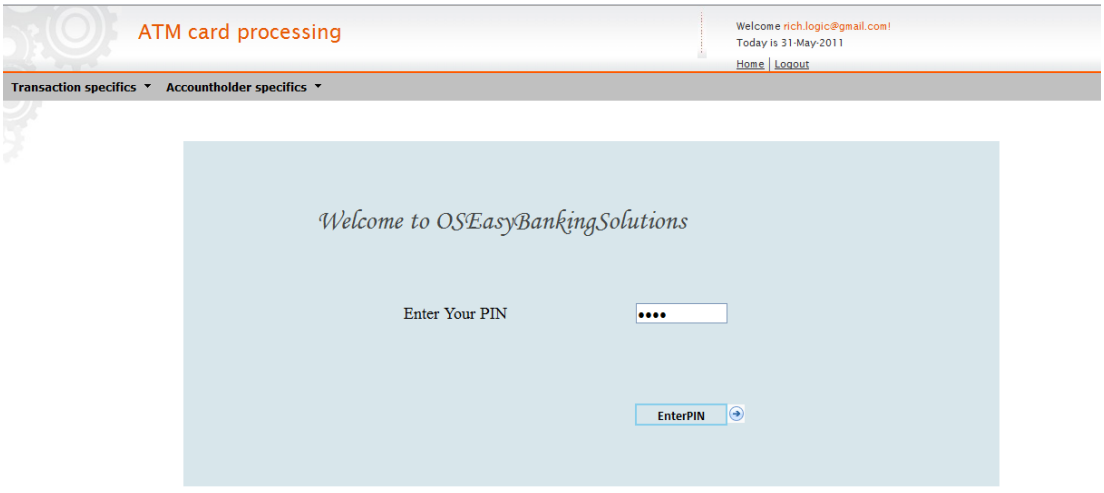

Figure 5.7: Entering the PIN to proceed for transactions

After the successful submission of the user's PIN value, the system again redirects the user to another screen where the user has the option to change the PIN, to inquiry the remaining balance in the account or to proceed for transactions. Figure 5.8 shows a user interface screen to opt for any one of these options.

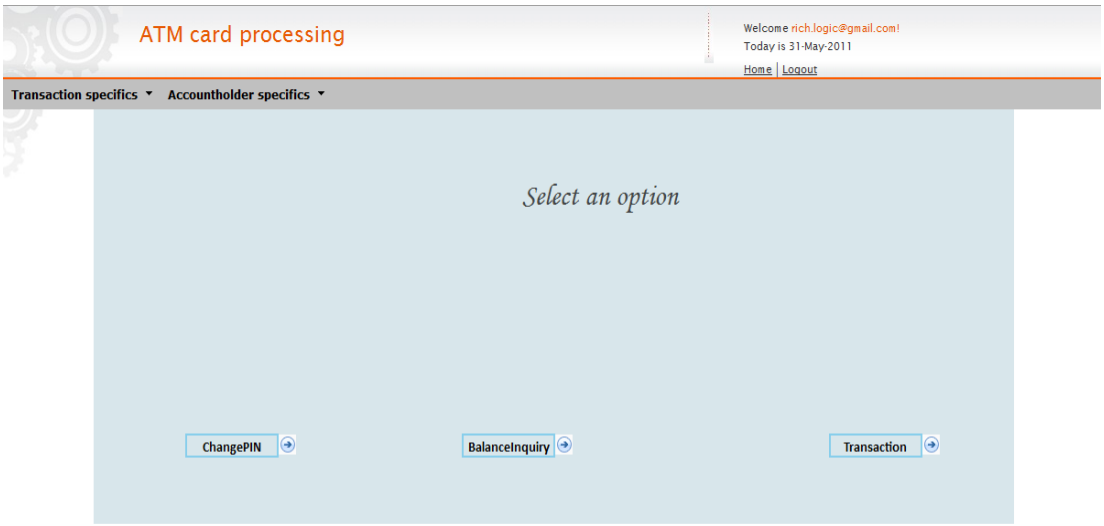

Figure 5.8: Selecting an option for changing the PIN, Balance inquiry and Transactions

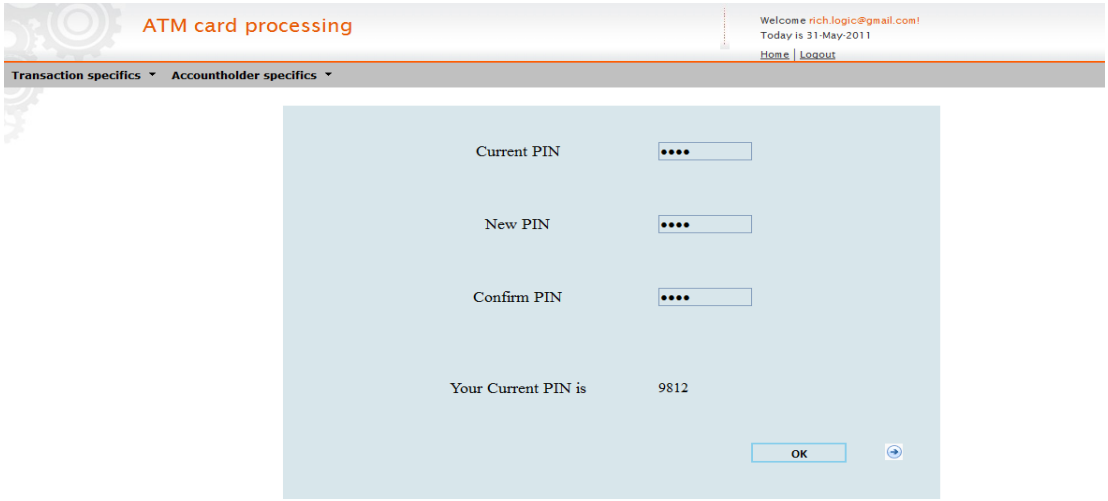

Figure 5.9: Changing the Current PIN to New PIN

Figure 5.9 shows the interface screen to change the current PIN value to new PIN value. The value is automatically changed in the system's server. Similarly, the user can select the type of transaction.

The user can either deposit the amount in his/her account or can withdraw the amount from the account provided the amount value is less than the remaining balance. Figure 5.10 shows the interface screen to select the type of transaction. Figure 5.11-5.12 shows the screens to enter the amount to be deposited in the customer's account or to withdraw the amount from customer's account.

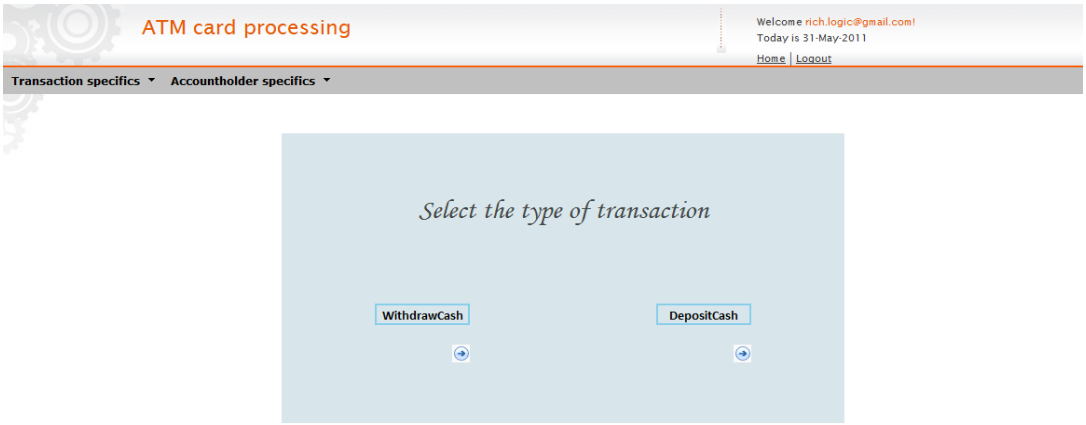

Figure 5.10: Selecting the type of Transaction

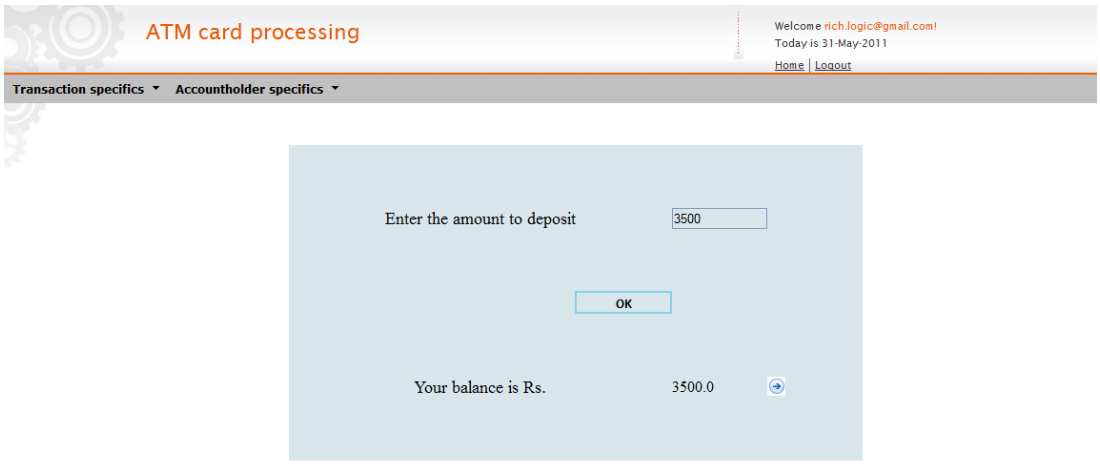

Figure 5.11: Entering the amount to deposit in Customer's account

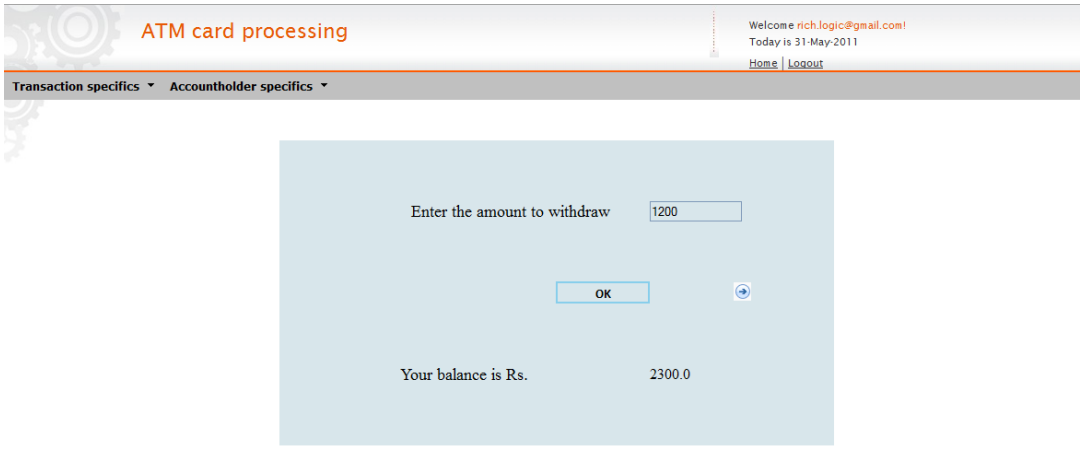

Figure 5.12: Entering the amount to withdraw from Customer's account

CHAPTER-6

CONCLUSION AND FUTURE WORK

The application that has been developed as part of this thesis is a contribution towards workflow scheduling in cloud computing environment. In this thesis, a workflow-based business application was designed supporting REST architecture style. The workflow specification for the application designed in workflow management system, YAWL was validated against YAWL syntax and semantics. YAWL, an open-source workflow management system is based on the concept of workflow patterns and Petri nets. The work also involved the design of the application in the process perspective of a visual modelling development environment, Orangescape Studio that itself runs as a cloud application. The application with its four-tiered architecture is built on REST-style service interface. From the implementation, we conclude that the REST-style web services can be combined with cloud computing platforms. It can help many enterprises to design and deploy process-oriented applications based on workflow.

Future work

- The current workflow management system, YAWL can be extended with support for more workflow patterns.
- Some future extensions can also be possible by considering multiple workflows for enterprise applications.
- Applications supporting multiple instances of workflow can also be designed and deployed on some of the cloud infrastructures offered by Google, Microsoft and IBM.

References

[1] www.classle.net/sites/default/files/44879/Cloud_Computing.doc

[2] http://wn.com/workflows

[3] M.Armbrust, A.Fox, et al., "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No. UCB/EECS-2009-28, University of California, USA, February, 2009.

[4] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds: towards a cloud definition," SIGCOMM Computer Communication Review, vol. 39, pp. 50–55, December 2008.

[5] M.Armbrust, A.Fox, R. Griffit,et al., "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[6] R.Buyya, S.Pandey and C.Vecchiola, "Cloudbus toolkit for market-oriented cloud computing", In Proceeding of the 1st International Conference on Cloud Computing (CloudCom2009), Beijing, China, December 2009.

[7] H.A.Reijers, "Design and Control of Workflow Processes", PhD thesis, Eindhoven University of Technology, Netherlands, August 2002.

[8] https://www.orangescape.com

[9] R. Sakellariou and H. Zhao, "A Hybrid Heuristic for DAG Scheduling on Hetero-generous Systems", The 13th Heterogeneous Computing Workshop (HCW 2004), Santa Fe, New Mexico, USA, April 26, 2004.

[10] E.Deelman, "Mapping abstract complex workflows onto Grid environments", in Journal of Grid Computing 2003; 1:25–39.

[11] Y.Du and X. Li, "Application of Workflow Technology to Current Dispatching Order System", International Journal of Computer Science and Network Security, 8(3), 59-61. March 2008.

[12] D.Cybok, "Workflow Management for Grid Computing: A Grid Workflow Infrastructure", msg systems ag, Munich, Germany, 2008.

[13] C.Hoffa, G.Mehta, T.Freeman, et al., "On the Use of Cloud computing for Scientific Workflows", 4th IEEE Int.Conf.on eScience, December 2008.

[14] R.Buyya, C.S.Yeo, S.Venugopal, J.Broberg and I.Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype and Reality for Delivering Computing as the 5th Utility", Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2008.

[15]http://community.zdnet.co.uk/blog/0,1000000567,10008269o2000458459b,00.htm.

[16] K. Liu, Y. Yang, J. Chen, X. Liu, D. Yuan and H. Jin, "A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-intensive Cost-Constrained Workflows on Cloud Computing Platform", International Journal of High Performance Computing Applications, vol.24 no.4 445-456,May,2010.

[17] C.Barnatt, "A Brief Guide to Cloud Computing: An Essential Introduction to the Next Revolution in Computing", Robinson Publishing, 1st ed., Aug 2010.

[18] R.Buyya, C.S.Yeo and S.Venugopal, "Market-Oriented Cloud Computing: Vision, Hype and Reality for Delivering IT Services as Computing Utilities", In Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Dalian, China, September, 2008.

[19] I. Foster, Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop, 2008, 1-10, Austin, Texas, USA, November 2008.

[20] K.Bakshi, "Cisco Cloud Computing: Data Center Strategy, Architecture and Solutions", Cisco Systems, 2009.

[21] P.Mell and T.Grance, "The NIST definition of Cloud Computing" , National Institute of Standards and Technology, Information Technology Laboratory, Version 15,July 2009.

[22] D.Xu and H.Liu, “Reviewing some Cloud Computing Platforms”, in Proceedings of the Second International Symposium on Networking and Network Security (ISNNS), 2010, pp.161-164.

[23] I.P.Sokolov, “Cloud Computing: Overview, Concepts and Business Deployment Scenarios”, Bachelor’s thesis, Vienna University of Economics and Business, Austria, 2009.

[24] P.C.Heckel, “Hybrid Clouds: Comparing Cloud Toolkits”, University of Mannheim, April 2010.

[25] B.Botelho, “VMware Entering Final Phase of Virtualization Evolution: Cloud Computing”, IT Knowledge Exchange, May 2008.

[26] P. F. Brown and R. M. B. A. Hamilton, “Reference model for service oriented architecture 1.0,” 2006.

[27] W.Li and P.Svard, “REST-based SOA Application in the Cloud: A Text Correction Service Case Study”, IEEE 6th World Congress on Services, 2010.

[28] “The Business Process Model”, white paper, Sparx Systems, 2004.

[29] R.Buyya, S.Pandey and D.Karunamoorthy, “Workflow Engine for Clouds”, in Cloud Computing: Principles and Paradigms, 1st ed., R.Buyya, J.Broberg, A.Goscinski, Ed.New Jersey: John Wiley and Sons, 2011.

[30] G.M. Juve, “A Resource Provisioning System for Scientific Workflow Applications”, Master’s thesis, USC Viterbi School of Engineering, University of Southern California, USA, December 2008.

[31] J. Yu and R. Buyya, “Workflow Scheduling Algorithms for Grid Computing”, Technical Report, GRIDS-TR-2007-10, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, May 2007.

[32] J. Yu and R. Buyya, “A Taxonomy of Workflow Management Systems for Grid Computing”, Journal of Grid Computing, pp. 171-200, September 2005.

[33] D.Hollingsworth, “Workflow Management Coalition: The Workflow Reference Model”, TC00-1003, January, 1995.

[34] A.V.Parameswaran, A.Chaddha, “Cloud Interoperability and Standardization”, SETlabs briefings, vol.7 no.7, 2009.

[35] H.Tan,W.M.P. van der Aalst, “Implementation of a YAWL Work-list handler based on the Resource Patterns”, In Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design,2006,pp.1184-1189.

[36] W. M. P. Van der Aalst, L. Aldred, M. Dumas, et al., “Design and implementation of the YAWL system”, In Proceedings of the 16th International Conference on Advanced Information Systems Engineering(CAiSE 04), Riga, Latvia, June 2004.

[37] J.Zhang, X.Lu, H.Duan, H.Nie, “A Medical information system architecture based on workflow technology”, IEEE International Symposium on IT in Medicine Education, 2009, pp.1117-1121.

List of Paper Published/Communicated

1. Ravneet K.Chawla, Ms. Anju Bala, “Developing workflow-based application in Cloud environment”, International Conference on High Performance Computing (Communicated)