CLOUD SERVICE ARCHITECTURE FOR EDUCATION SYSTEM UNDER OBJECT ORIENTED METHODOLOGY

Ankit Kumar Singh1, Saroj Kumar2, Abhishek Rai3

[1]*M Tech Student Computer Science Department, Babu Banarasi Das University, Lucknow U.P (India)*
[2]*Assistant Professor Babu Banarasi Das National Institute of Technology & Management, Lucknow U.P (India)*
[3]*M Tech Student Computer Science Department, Babu Banarasi Das University, Lucknow U.P (India)*

Abstract

Cloud computing has become an increasingly attractive option for delivering education services. This service is secure reliable and economic than ever. This service is available in free or low cost mode depend upon architecture of cloud. These services are used daily by learners and educators to support learning, publishing and collaboration. This paper introduces an architecture corresponding to cloud services for education system under object oriented methodology. Cloud service architecture is completely remodeled under object oriented approaches like class diagrams, sequence diagrams and use cases diagrams. Further approaches are validated by formal specifications like Context Free Grammar.

***Keywords** - Cloud services, SOA, OOT Approach, CFL/CFG*

--***--

1. INTRODUCTION

Cloud Computing has developed into a general stage for service providers who can offer different kinds of services and, the users who expect to find most of the services they need in a particular location. There are many IT enterprises arranging services over cloud platform such as Google, Amazon , Microsoft Azure.

Cloud computing refers application which are provided to users as a service and datacenters (consist hardware and software) to manage and maintain these services. The cloud provides a platform in which user has to pay according to his requirement and in this way it provides a pay per use model.
Cloud Architectures differ from software architectures due to basic architectural elements, services and dynamic behavior .Due to its dynamic behavior dynamic reconfiguration is performed autonomously in order to meet the service users requirements. Dynamic reconfiguration provides reinstatement of cloud services that is used by another in case of abasement of quality criteria’s or when new service is introduced. So it is important to define an architecture corresponding to cloud so that the logical separation or valid partitions can be captured e.g SLA [9] between service provider and service consumers.

2. CLOUD SERVICES

According to the NIST definition, cloud computing specifies three delivery models to provide various services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)

A. **Platform as a Service (PaaS)** It provides a development platform with a set of services to support application design, development, testing on the cloud. Example Google Compute Engine, AWS Elastic Beanstalk and Microsoft Azure
B. **Infrastructure as a service (Iaas)** In Iaas cloud provides a virtual infrastructure so that end user and developers can scale up and down computing resources demand dynamically . IaaS provides virtualization computing power, storage and network connectivity of the data centers and offers it as a service. Examples Amazon EC2, Windows Azure Platform.
C. **Software as a service (Saas)** In this software is presented to users as a service on their demand. It saves the user form the trouble of software deployment and its maintenance. Software are usually presented in a web browser as a service. Pricing is based upon the usage of the software. Examples Microsoft Dynamic CRM online, Google Apps, Sales force **.**

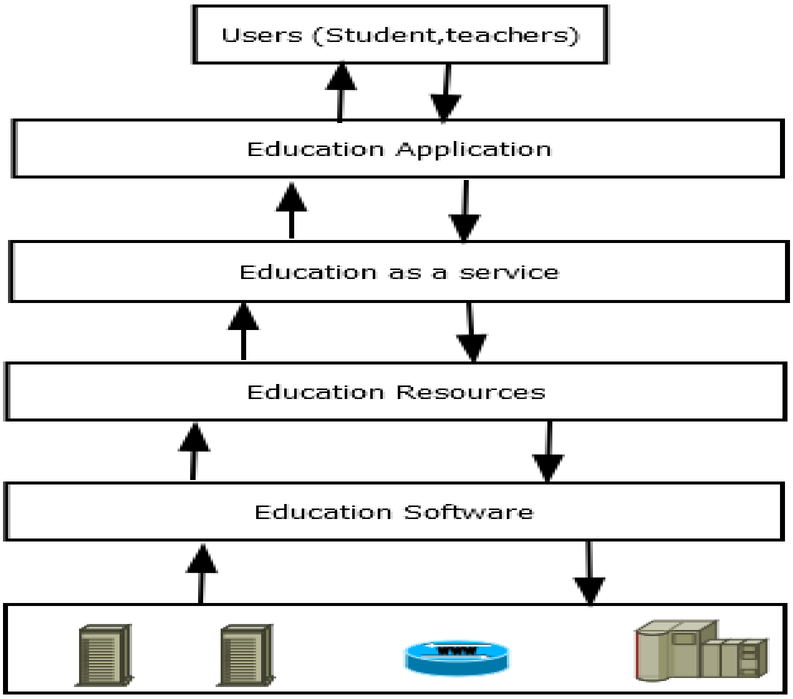


Fig 1. Education System as Cloud Service Model

Purposed Architecture of Cloud Service

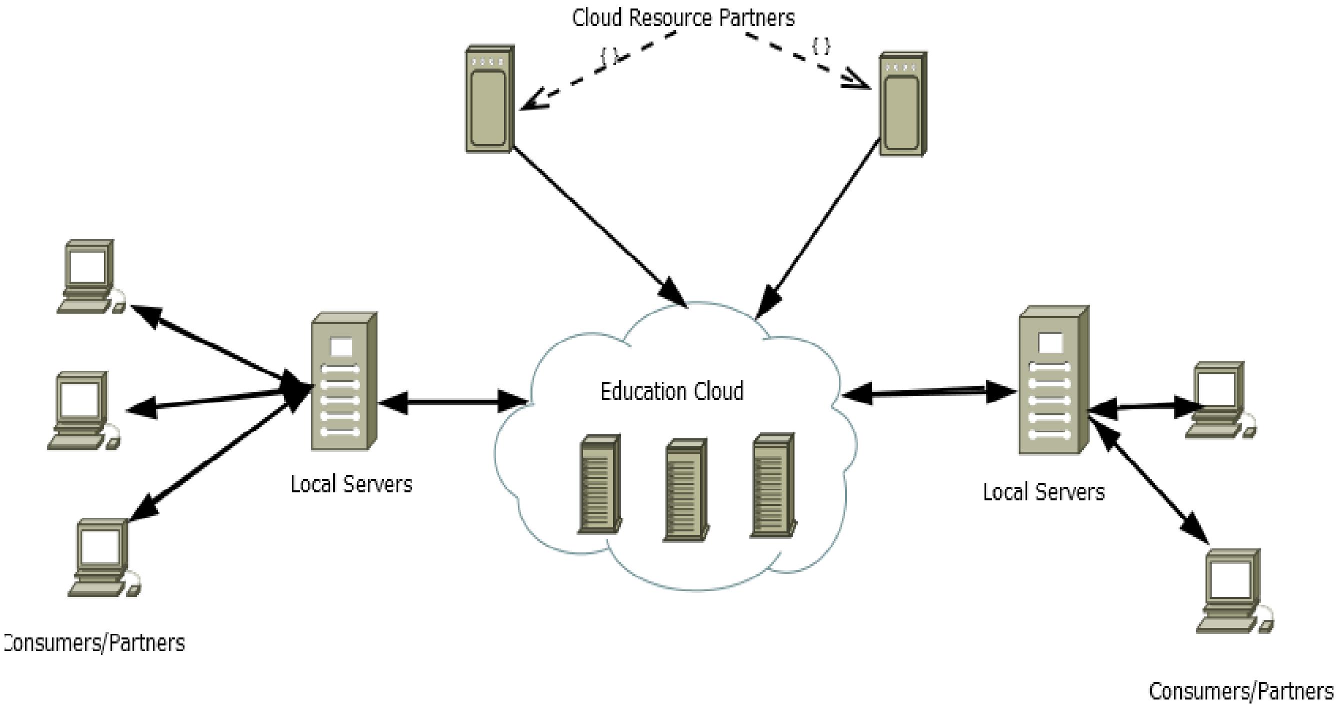


Fig 2 Purposed Architecture for Education System

3. SERVICE ORIENTED ARCHITECTURE (SOA)

SOA is essentially an architecture framework that can help cloud computing architecture to provide the required service model with agility and scalability. It also provides interoperability between application on different platforms. SOA is a design pattern which is combination of loosely coupled, discoverable and reusable services. SOA defines an intercommunication model between three fundamental units

A. **Service Consumer Service** consumer requests services on the basis of demand and its requirement.

If consumer knows the location of service it directly communicates with the provider of the service

B. **Service Provider** Service provider executes and receives the requests of the consumer and also sends the response corresponding to requests

C. **Service Broker** Service broker provides a Service Level Agreement (SLA) between a provider and consumer.

4. OOT APPROACH

OOT approach provides the capability to model both the static and dynamic appearance of an application. So it is better for modeling the Cloud Service models. For the static approach class diagrams are defined. Classes which describe a model are defined using the class diagram which also defines the interaction between the classes.

4.1 Use Cases

Use Cases will be used to characterize the requirement of the model . Use-Cases will be used to identify the actions and actors . Actors are the anything that interact with the system. Each actor has a specified role. So a human being may be represented by several actors. There are basically three categories of actors. Cloud service consumer, cloud service provider, cloud service developer which include seven sub categories of actors

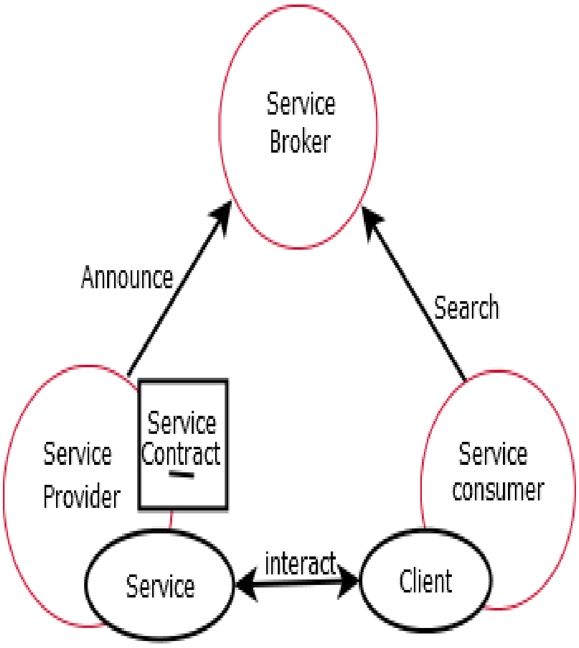


Fig 3 Service Oriented Architecture Conceptual Model

1) Cloud actor(cloud service developer) type
    1] Service Developer
2) Cloud actor (cloud service provider) type
    1] Service operation manager
    2] Service business manager
    3] Service transition manager
3) Cloud actor (cloud service consumer) type
    1] Consumer business manager
    2] Consumer service administrator
    3] Service user

1. Service developer responsibilities
    a. Designs, implements, and maintains service templates (technical aspect).
    b. These templates can be used by providers to create offerings.
    c. The person performing this role could be employed by the same organization that is a cloud service provider or cloud service consumer.
2. Service operation manager responsibilities
    a. Provides reports from monitoring and measurement (used for audit and compliance).
    b. Monitors and measures performance and utilization against SLAs
    c. Manages the technical infrastructure required for providing cloud services
3. Service business manager responsibilities
    1] Offers all type of services developed by cloud service developer.
    2] Accounts for services potentially offered by service providers themselves and services offered on behalf of cloud service developers.
    3] Establishes a portfolio of business relationships, and sets up accounts and terms for cloud service consumers
4. Service transition manager responsibilities
    a. Enables a customer to use the cloud service, including "on boarding", unification, and process acceptance
    b. Defines and creates service offerings based on templates that can be used by cloud service consumers.
5. Consumer business manager responsibilities
    1] Approves business and financial expenditures for consumed services.
    2] Establishes business relationships; sets up accounts, budget, and terms; and so on
6. Consumer service administrator responsibilities
    a. Requests service instances and changes to service instances.
    b. Purchaser of services with the business relationship.
7. Service user responsibilities
    1] Uses service instances provided by a cloud service provider
    2] The Service User role is distinct from the Consumer Service

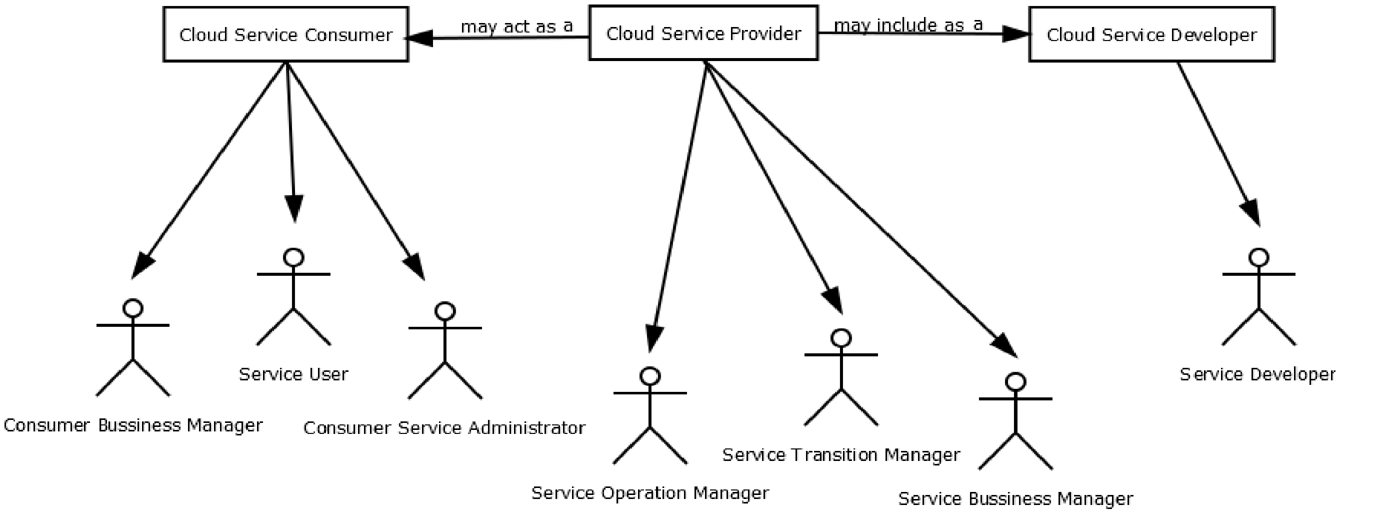

Cloud Service Consumer
may act as a
Cloud Service Provider
may include as a
Cloud Service Developer
Consumer Bussiness Manager
Service User
Consumer Service Administrator
Service Operation Manager
Service Transition Manager
Service Bussiness Manager
Service Developer

Fig 4 Cloud Actors

The participation between the classes to perform the use cases can be performed by dynamic diagrams, such as sequence diagrams or activity diagrams.

4.2 Class Diagram

1) Education resource catalog: Education resource catalog contains education data offer, education data request ,and service contract. It contains a variety of information about the services including the cost of services, SLA and who can use the services
2) Education data request: Education data request is used by the consumer to request the provisioning and modification of education resources. Education data request needs service contract which may be created before or simultaneously with data request.
3) Education data offered: Education data offer is selected from the service provider resource catalog and then provisioned into request class. Education data offering created by the service developers contains SLA, cost and billing information and other necessary information.
4) Service contract: It is a contract between a service provider and service consumer. It depends on service offering and indirectly depends on the education data catalog. Contracts are modified, established and eventually terminated.
5) Provider info: Provider information is the identifying information that is offered by the cloud service provider when a relationship is established.
6) Consumer info: Consumer information is the identifying information that is provided by the cloud consumer when the relationship is built.

4.3 Object Database Correspond to Class Diagram

1) Education Resource Catalog (Edu_Res_Cata)

Name	Content	Discription
Provider Id	String ID(i18n)	The Id of cloud service provider
Name	String(i18n)	The human friendly name of this artifact
Supported Operation	ID(Reference) [*]	The IDs of all the supported Operation objects available for this object

2) Education Data Request (Edu_Data_Req)

Name	Content	Discription
Name	String(i18n)	The human friendly name of this artifact
Consumer Id	String ID(i18n)	The Id of cloud service consumer
Contract Id	String ID(i18n)	The ID of the service contract that is requested
Supported Operation	ID(Reference) [*]	The IDs of all the supported Operation objects available for this object

3) Education Data Offered (Edu_Data_Offer)

Name	Content	Description
Name	String(i18n)	The human friendly name of this artifact
Provider Id	String ID(i18n)	The Id of cloud service provider
Contract Id	String ID(i18n)	The ID of the service contract that is requested
Supported Operation	ID(Reference) [*]	The IDs of all the supported Operation objects available for this object

4) Service Contract (Ser_Con)

Name	Content	Description
Name	String(i18n)	The human friendly name of this artifact
Provider Id	String ID(i18n)	The Id of cloud service provider
Consumer Id	String ID(i18n)	The Id of cloud service consumer
Offering Id	String ID(i18n)	The identifier of the service offering on which this contract is based. The ID is unique at the cloud service provider's level.

5) Provider Information (Pro_Info)

Name	Content	Description
Provider Id	String ID(i18n)	A globally unique identifier of a cloud service provider. It is used to establish namespaces in which other fields are unique.
Name	String(i18n)	A human-friendly name of the provider
Address	String(i18n) [*]	Human-friendly addresses of provide
Contact Information Id	ID (Reference) [*]	The IDs of various contact points into the cloud service provider, such as administrative contact and technical contact

6) Consumer Information (Con_Info)

Name	Content	Description
Consumer Id	String ID(i18n)	A globally unique identifier of a consumer. It is used to establish namespaces in which other fields are unique.
Name	String(i18n)	A human-friendly name of the consumer
Address	String(i18n) [*]	Human-friendly addresses of the consumer
Contact Information Id	ID (Reference) [*]	The IDs of various contact points into the cloud service provider, such as administrative contact and technical contact

7) Relationship (Rela)

Name	Content	Description
Description	String(i18n)	A human-friendly description of this artifact
Provider Id	String ID(i18n)	The Id of cloud service provider
Consumer Id	String ID(i18n)	The Id of cloud service consumer
Credentials Token	ID (Reference) [*]	The IDs of various credential Tokens

4.4 Sequence Diagram

The Steps of the procedure are summarized below

1) First all user request are send to the local server with necessary user identification information like user id and password.
2) The verification module of the local server verifies the user and also sends the credentials information to the user.
3) The user sends the request for the resources according to his her requirement.
4) After getting the exact specification from the user local server sends the token to user which contains pricing information, encryption system and other data security information.
5) The user sends the request a local server to establish service contract.
6) Local server forwards this request to Education Resource Catalog which is developed by service provider and also contains offering information.
7) ERC forwards the request to Cloud (service provider) for the authorization of request And establishment of contract.
8) Cloud authorizes the request and forwards it towards the ERC.
9) ERC forwards the local server and local server also forwards the authorization to user.
10) The local server forwards the resources to the requested user as soon as it receives the resources to the Cloud.

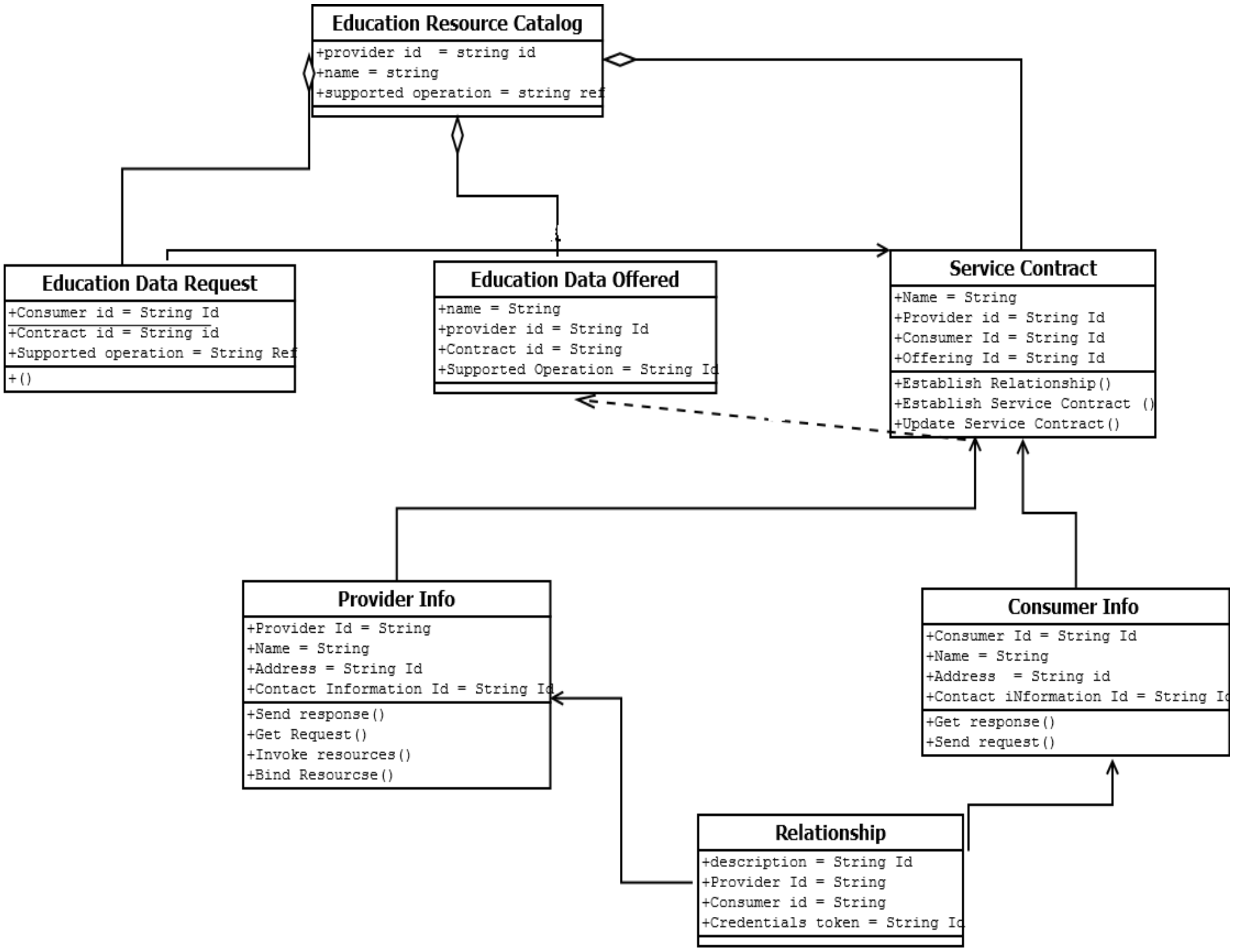


Fig 5 Class Diagram of Education System as cloud Service Model

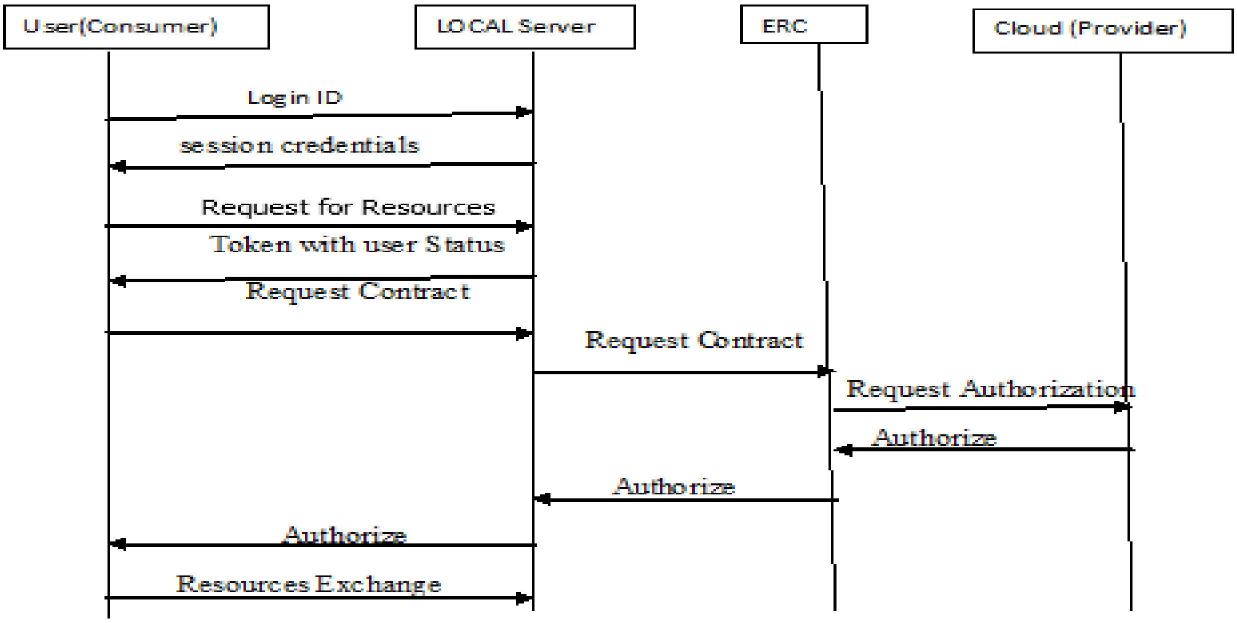


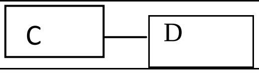
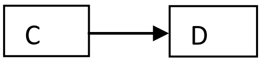
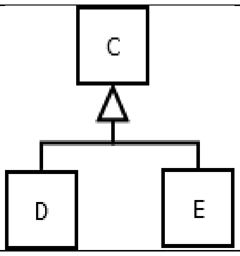
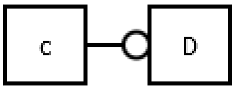
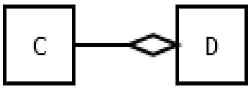
Fig 6 Request Initialization Procedure

5. CONTEXT FREE LANGUAGE/CONTEXT FREE GRAMMER

This section indicates to our method to develop a formal specification corresponding to OOT Approach by using a syntax and semantic approach. The main terminal symbols and instance variable are mapped to terminal symbols.

A class diagram consists also of applications, which will be identified when the denotation of context-free grammar are defined. Associations represent the cooperation between classes and have to be included in a context-free grammar. For generalization we propose the production A → B | C. The non-terminal A can be implemented either with the non-terminal B or non-terminal C.

Table 1 From a Class Diagram to CFG [1]

Association	Class Diagram	Grammar
Classes	Class C Attribute	Class (non-terminal) attribute (terminal)
Association		C→D
Navigability		C→D
Generalization		C→D/E
Aggregation		C→D
Composition		C→D

The language generated by CFG could have numerous productions or be infinite. Then it is not advantageous to perform static validation. So there are set of rules for calculating variability and bringing out all possible use cases in an automated process achieved by performing an in-depth search to derive all string instances reachable from the start symbol.

Table 2 Rules for Variability Calculation [4]

- Var (A ::= B1 … Bn) = Var(B1) * … * Var(Bn)
- Var (A ::= B1 | … | Bn) = Var(B1) + … + Var(Bn)
- Var (A ::= a) = 1 (single non-terminal)
- Var (A ::= eps) = 1 (empty production)
- Var (A ::= AB) = Var(B) (left recursion)
- Var (A ::= BA) = Var(B) (right recursion)

A. Context Free Grammar

1. Relationship → Provider info | Customer info
2. Provider info → Service contract
3. Customer info → Service contract
4. Service contract →Education data offered Education| resource catalog
5. Education data request → Education resource catalog |Service contract
6. Education data offered → Education resource catalog
7. Education resource catalog → eps

REFERENCES

[1] Toma·z Kosar1, Marjan Mernik 1, Viljem Zumer 1, Pedro Rangel Henriques 2, Maria Jo~ao Varanda Pereira 3," Software Development with Grammatical Approach, 1.University of Maribor, Faculty of Electrical Engineering and Computer Science, Slovenia E-mail: ftomaz.kosar, marjan.mernik, zumerg@uni-mb.si.2.University of Minho, Department of Computer Science, Portugal E-mail: prh@di.uminho.pt, 3.Polytechnic Institute of Bragan»ca, Portugal E-mail: mjoao@ipb.pt

[2] Tahereh Nodehi1, Sudeep Ghimire2, Ricardo Jardim-Goncalves3*, Antonio Grilo4 "On MDA-SOA based Intercloud Interoperability framework" CMSS - VOL. I, ISSUE 1/2013 Faculdade de Ciências e Tecnologia da, Universidade Nova de Lisboa, Campus de Caparica, Monte de Caparica, Portugal 1) UNL, t.nodehi@campus.fct.unl.pt 2) UNL,UNINOVA,sudeepg545@gmail.com 3) CTS, UNINOVA, rg@uninova.pt UNIDEMI, acbg@fct.unl.pt

[3] Everton Cavalcante" Architecture-Driven Engineering of Cloud-Based Applications" ECSA 2013 DIMAp {Department of Informatics and Applied Mathematics} UFRN { Federal University of Rio Grande do Norte Natal, Brazil evertonrsc@ppgsc.ufrn.br

[4] Faizan Javed1, Marjan Mernik2, Barrett R. Bryant, Jeff Gray3, "A Grammar-Based Approach to Class Diagram Validation".1. Department of Computer and Information Sciences University of Alabama at Birmingham 1300 University Boulevard Birmingham, AL 35294-1170, USA javedf@cis.uab.edu, 2) Faculty of Electrical Engineering and Computer Science University of Maribor Smetanova 17 2000 Maribor, Slovenia marjan.mernik@uni-mb.si, 3. Department of Computer and Information Sciences University of Alabama at Birmingham 1300 University Boulevard Birmingham, AL 35294-1170, USA {bryant, gray}@cis.uab.edu

[5] Shrabanee Swagatika, P.K. Pattnaik "Design Criteria of SOA for Cloud Based Infrastructure Resource Management as a Service" International Journal of Instrumentation, Control & Automation (IJICA), Volume 1, Issue 1, 2011

[6] Rajpal Choudhary " A SURVEY ON CLOUD COMPUTING ARCHITECTURE" Int.J.Computer Technology & Applications, Vol 3 (4), 1400-1405 2012 ISSN:2229-6093 Singhania university,Pacheri Bari(Jhujhunu) rajpal_choudhary2002@yahoo.com

[7] Ruchika Saini1, Pawan Prakash1," A New Approach to Volunteer Cloud Computing" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 11, Issue 6 (May. - Jun. 2013), PP 43-45

[8] Said Ghoniemy 1, Sultan Aljahdali 1, Ashraf Fahmy 2," A Dedicated Web-Based Learning System" Universal Journal of Computer Science and Engineering Technology 1 (2), 84-92, Nov. 2010. © 2010 UniCSE, ISSN: 2219-2158.1. Department of Computer Engineering Department of Computer Engineering ,College of Computers and Information Systems ghoniemy@tu.edu.sa, a.fahmy@tu.edu.sa, 3. Department of Computer Science, College of Computers and Information Systems Taif University, Taif, Saudi Arabia , aljahdali@tu.edu.sa.

[9] Linlin Wu and Rajkumar Buyya "Service Level Agreement (SLA) in Utility Computing Systems" Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia, Email: {linwu, raj}@csse.unimelb.edu.au .