Incorporating MDA to Design Business Intelligence Services by using SaaS Model of Cloud Computing

Anil Sharma
Research Scholar
Himachal Pradesh University
Shimla, Himachal Pradesh, India

Manu Sood, Ph.D.
Professor
Himachal Pradesh University
Shimla, Himachal Pradesh, India

ABSTRACT
Cloud computing based on the latest information and computing technologies provides software programs, platforms, infrastructure etc. as services. On the other hand, Business Intelligence (BI) on the Cloud is an assistance model that helps organizations further their strategic business interests and it is for this reason, every organization is looking forward to its generation and effective usage. Cloud SaaS (Software as a Service) refers to a cloud computing service model in which the software applications are offered as services. With the continuous advancements in ICT, the organizations are under constant pressure of replacing the older ones. Model Driven Architecture (MDA) is a software development model that aims at minimizing the impact of these technological advancements on software applications development so as to become a preferred methodology for the development of cloud software services. In this paper, authors are suggesting an MDA based model driven approach to design business intelligence services by using SaaS model of cloud computing for a cloud based data warehouse.

Keywords
Business Intelligence, Cloud Computing, Cloud SaaS, Data Warehouse, Model Driven Architecture, Service Oriented Architecture, Software as a Service.

1. INTRODUCTION
Massive research and development in the field of hardware and software technologies are putting organizations into the state of dilemma for the selection of hardware and software. As such, going for updates all the times is proving to be costly and time consuming to these organizations. Therefore, the advent of Cloud Computing has come as a great relief to them. Cloud Computing is an internet based computing model where the computational resources like hardware, software, development environment and other infrastructure are made available as services. These services can easily be accessed on any computing device having communication support. These services act like utilities which are made available on-demand on pay-per-use model or can be subscribed. In a cloud computing system, there is a considerable workload shift. Local computers having lighter configurations need not to do much of computing when it comes to running applications; most of it is managed by the network of computers on the cloud. This helps in decreasing the requisite hardware, software and other related infrastructure on the user premises. The user's computer just needs interface software, such as a web browser, to get connected to the cloud and this cloud takes care of the rest [1].

Cloud computing has evolved from a range of relevant legacy technologies and concepts such as grid computing, virtualization, Web Service Definition Language (WSDL), Simple Object Access Protocol (SOAP) and Universal Description Discovery and Integration (UDDI), Service Oriented Architecture (SOA), Software as a Service (SaaS) etc. The Software as a Service (SaaS) which is a part of classification of Cloud Computing is an emerging business model in the software industry. It is a software application delivery model by which an enterprise vendor develops a web-based software application, and then hosts and operates that application over the Internet for use by its customers. Customers do not need to buy software licenses or additional infrastructure equipments, and typically only pay monthly fees or annuity payments for using the software [2].

Business Intelligence (BI) is the key to strategic planning of an organization to have better and timely decisions which helps in gaining an edge over the competitors in this highly competitive world. BI includes the formation of systems and technological support. BI software tools help the business manger to look into the past and to frame new strategies to take better decisions so that organization's future can be secured and it can survive and grow with the changing time. BI systems help a company to gather, store, access and analyze corporate data. BI consists of several related activities, including data mining, online analytical processing, querying and reporting. Today, it is difficult to find a successful organization that has not adopted BI technology for its better business management. Companies use BI to improve decision making, cut costs and identify new business opportunities. In general business intelligence is used in the areas of customer profiling, customer support, market research, market segmentation, product profitability, statistical analysis, and inventory and distribution analysis etc [3].

Model Driven Architecture (MDA) is an approach to system specification and interoperability, based on the use of formal models. In MDA, Platform Independent Models (PIMs) are initially expressed in a platform-independent modeling language, such as UML. This PIM is subsequently translated to a Platform Specific Model (PSM) by mapping the PIM to some implementation language or platform using formal rules. At the core of the MDA concepts are a number of important OMG standards: The Unified Modeling Language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI), and the Common Warehouse Meta model (CWM) [4]. These standards define the core infrastructure of the MDA, and have greatly contributed to the current state-of-the art of systems modeling. MDA is aimed at separating business logic from its

application logic. The business logic exhibits lesser complexity compared to that inherent in the underlying implementation technologies of application logic.

Cloud BI is an assistance model that is growing gradually and many of the organizations are looking forward for this. Cloud SaaS refers to a cloud computing service model in which the software applications are offered as services. The advancements in the field of computing and information require development of software solutions in a manner that is independent of technological advancement. The MDA approach may be leveraged to develop the software applications that would be deployed in the cloud, the cloud Software-as-a-Service (SaaS). A platform-independent model (PIM) of the cloud application would reflect its structure, behavior and functionality irrespective of the technology used for its implementation. The platform-specific model (PSM) of the application, on the other hand, would be more implementation-oriented and bound to a given execution platform. The transformations from the PIM to PSM would be carried out using transformation tools developed for the purpose.

Data warehouse is the central point on which BI applications run to analyze the historical data and the most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a Star Schema or a Snowflake Schema. As an illustration to support the importance of BI, the authors have taken a simple illustrative example of a Result Evaluation System (RES) of an educational institution as an instance to describe the design of data warehouse using Star Schema. The authors in [5] propose an MDA approach to design the relational schema of RES and then the star schema by using the relational schema. Authors emphasized that quick and easy access to data about students' performances is critical in taking timely strategic decisions. But access to data alone isn't enough. Institutes also need to analyze performance data from various perspectives, find patterns, and see the "big picture". With a "big picture view" of individual student's performances, teachers are in a better position to identify potential problems and take the appropriate prescriptive or preventative actions to ensure that each student performs to the best of their abilities.

In RES, data related to performance of the teachers as well as the students in different subjects and in different semesters of different courses are maintained in a data warehouse. Data related to students' performance are of two types: Internal Performance Data and External Performance Data. Internal performance data means performance score in various parameters like class tests, assignments, seminars etc. and external performance data means performance score in final/ external examinations. To record, maintain and manipulate such kind of data is a tedious and time consuming task in a distributed systems environment. Further, suppose for example, management of the institution is interested in finding out from a remote location, the students' performance in different subjects along with the teachers' performance in the respective subjects. It then requires a support such as that of the cloud computing in order to be helpful in satisfying the needs of the system users. Hence, the authors in this paper propose an MDA based approach to design BI services by using the SaaS model of cloud computing for a cloud based data warehouse.

The rest of the paper has been organized in various sections. In section 2, the authors suggest the steps to create the SaaS Architecture for the cloud BI whereas section 3 highlights the procedure to develop the web services business process model followed by section 4 and 5 that highlight the significance of application of MDA and SOA approaches respectively. Section 6 with the help of the illustrated example of the RES, explains how to generate the requisite BI services based on the cloud SaaS. Finally section 7 highlights the conclusion and the scope of the future work.

2. SAAS ARCHITECTURE FOR CLOUD BI

Cloud SaaS is a software application delivery model by which an enterprise vendor develops a web-based software application, and then hosts and operates that application over the Internet for use by its customers. Sharma et al. in [6] suggest that many of the efforts are done in the development of these services in-house but with the advancements in the communication technology these services become obsolete and need to be replaced on regular time interval. So the need of the hour is to develop these services in such a way that these should absorb the technological changes and should be scaled up accordingly. By keeping this into the context, the authors of this paper proposed the usage of Model Driven Architecture (MDA) to develop the services to formulate Cloud based Business Intelligence.

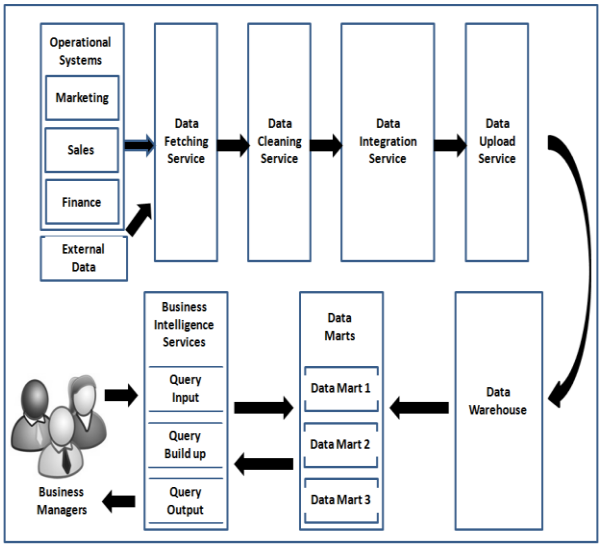

Fig 1: Cloud BI SaaS Architecture

The proposed problem needs two fold solutions. First the design of data warehouse for the RES and then design of business intelligence services on cloud computing. The authors in [5] have already suggested the MDA approach to design the relational schema for RES and further designed the Star Schema of RES using MDA. After designing Star Schema, there are many of the Service Providers those who provide the DWH as a service on cloud. Any one of them can be chosen for the reference. In this paper, authors are focusing on the design of various services which will be dealing with this cloud based DWH to generate business intelligence. The architecture of the Cloud BI SaaS is given in the following Fig. 1.

As shown in Fig 1, Data collection is done from various operational systems and data belongs to various types like marketing data, sales data, finance data etc. Business Organizations also collect data from External sources to know about what their counter parts are doing to enhance their business capabilities. Data collection from heterogeneous data sources is a complex task and the Authors propose a service to fetch data from these various operational systems. After fetching the data, next service is to perform the data cleaning process. Data cleaning includes the finding of missing values, spurious values, misspelled data etc. Data Integration Service will take place after the completion of data cleaning process and finally Data Upload Service will take charge to upload this fine tuned data to the data warehouse repository. At this point of time the whole data is centralized and to improve the performance of query response time, this data warehouse can be distributed further into Data Marts. Data Marts are created to keep the data at the nearest site i.e. data related to subject of interests to the respective users. Now Business users need to interact with the data warehouse to formulate business intelligence. Inputs given by the users will be in the form of a query. So, Query Input Service will serve this purpose. After receiving the query input successfully, Query Buildup service will take charge to build the query in the required format. And finally to send the output of the query back to the Business Managers, Query Output Service will take the charge.

3. WEB SERVICE – BUSINESS PROCESS MODEL

XML-based format called Web Service Description Language (WSDL). WSDL supports a complete description of the operations available and the parameters required to use those business services. Once a Web service is defined, its description is published in a repository that conforms to the Universal Description, Discovery and Integration (UDDI) specification. Clients can query the repository in a variety of ways to discover an appropriate service. For example, a client might request a service that is known by name, or the client might search for a service based on a category.

Accepting requests and returning replies requires that the request be formatted as a message according to the Simple Object Access Protocol (SOAP), which is a modular protocol built on top of XML. On the Internet, SOAP messages are sent using HTTP or HTTP/S (although SOAP is actually protocol-independent). SOAP's XML foundation is important because XML provides an extensible mechanism for creating self-describing messages and content. This extension capability allows for a loosely coupled relationship between sender and receiver, which is especially important over the Internet where two parties may be in different organizations or enterprises. In addition, XML is represented in ASCII text, which can easily pass through corporate firewalls over the HTTP protocol [7]. Finally, bridging between the Web service interface and the internal implementation of the service means that the SOAP request is received by a run-time component that accepts the SOAP message, extracts the XML message body, transforms the XML message into a native protocol, and delegates the request to the actual business process (or software component) within the enterprise. These runtime extraction and transformation capabilities may be hosted within a Web services container, which provides scalability, load balancing, and other enterprise qualities-of-service for the Web service itself.

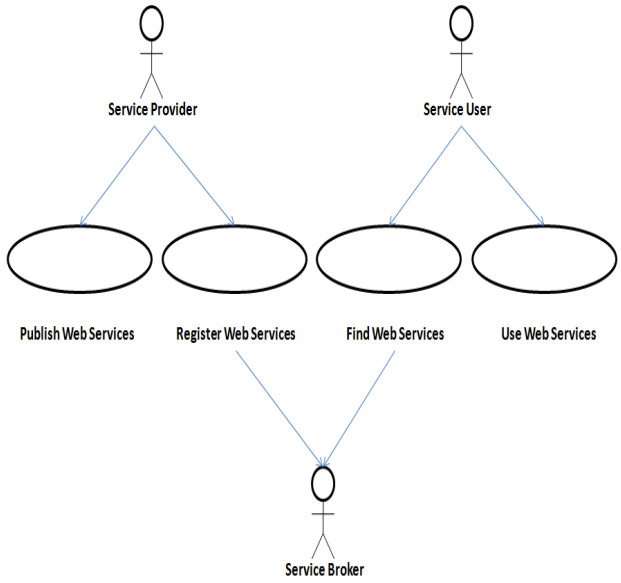

Fig 2: Web Service Business Model

As shown in Fig 2, each web service has its own life cycle and business model. There are mainly three players in web service model namely: Service Provider, Service Broker and Service Requester. Service Provider is the company or a person who actually develops the service and publishes it to the web. Service Broker is like a library where all the services get registered and these are available to the Service Consumers who request the service.

4. APPLYING MODEL DRIVEN ARCHITECTURE

Cloud SaaS refers to a cloud computing service model in which the software applications are offered as services. Because there is a drastic change in the hardware and software technologies, directly developing these cloud software services using available technologies, modeling them at a higher level of abstraction will decouple them from the undesired effects of technology change and enhance their longevity. An MDA based development of cloud SaaS (application) will enable defining these services in a technology-independent manner and will play a significant role in improving the quality of cloud software services, making them more robust, flexible and agile. Encapsulating business logic in a manner that is independent of the technical mechanisms will formally capture the essence of the applications; and will also make it possible to reuse them in a variety of contexts [8]. To illustrate the concept, authors have taken an illustrative example of the data warehouse of RES as a service available on the cloud. As such, there are many vendors offering data warehouse as a service on the cloud [9], so any one of these could also have been taken as a potential candidate for the example. But, since the authors have already been working on the RES, they thought it proper to continue with the same. Though, the concept presented in this paper can easily be applied to these example too with a little bit of extrapolation.

The RES may be accessed by anyone connected to the Internet, through a web browser interface. Authors of this paper assume a simplified approach to RES where following steps are considered for the business process:

- Student takes an admission to the institutes.
- Pay the fee to the Accounts department to confirm his/ her admission.
- Student opt a course through a list of available courses in the institution.
- Various subjects are allocated to him/ her according to the course enrolled in.
- There are different faculty members who are teaching different subjects.
- Student is awarded the marks to get the grade in the course.
- Marks are of two types namely Internal Marks and External Marks. Internal marks are awarded by the subject teacher based on the parameters like class tests; seminar, assignments, class participation etc. and external marks are given based on the final examination conducted by the institution.
- After filling all the marks details, final result is prepared by the institute clerk.
- Management take the final result and perform various analysis to check the performance student wise, subject wise, teacher wise etc.

A use case diagram capturing the functionality of the system is depicted in the following Fig. 3. To illustrate the example, authors are using Unified Modeling Language (UML), but MDA is not restricted to it only. The characteristics of the actors in the system are:

- Student is the person acting as hub of the system on which the whole RES is running.
- Accountant is the person taking care of all the financial transactional related to the fee deposited by the student.
- Faculty again is the key person of the RES who is teaching and awarding the marks to the student for his/ her grading purpose.
- Clerk is the person who is actually recording the all the transaction related to the marks on a centralized database.
- Management is a person or a group of persons using this result database to perform various types of analysis like student wise analysis, faculty wise analysis, subject wise analysis etc. Management performs these types of analysis to take better decisions to improve the overall performance of the institute which include students, subjects and faculty as a whole.

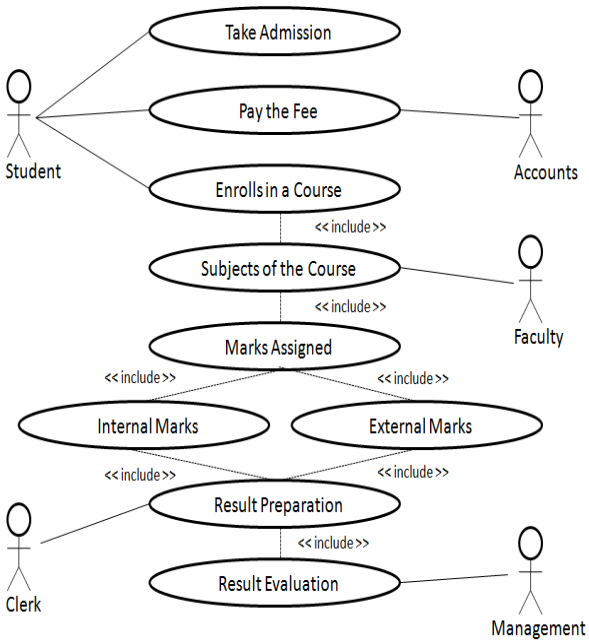

Fig 3: RES - Use Case Diagram for RES

5. SERVICE ORIENTED ARCHITECTURE

SOA is a way of designing, developing, deploying managing services on web. These services represent reusable business functionality; through standard interfaces, service consumers compose applications. Services in SOA are governed by a set of key principles that include loose coupling, autonomy, abstraction, reusability, compos ability, statelessness, discoverability, and adherence to a service contact [7]. These principles enable the services to evolve independently. SOA results in the creation of business solutions that consist of inherently interoperable services. An XML-based, vendor-neutral communications framework established by web services-driven SOA enables cross platform integration and intrinsic interoperability among the services. The standards comprising this framework are – Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP) and Universal Description, Discovery, and Integration (UDDI). WSDL is an XML-based standard for service description. A service description specifies the name of the service, the data to be provided to the service and the data that would be returned by the service. SOAP provides XML compliant communications format required by the services. The service description registry and discovery is realized through UDDI.

SOA results in the creation of business solutions that consist of inherently interoperable services. Service Oriented Architecture (SOA) represents an architectural style in which the automation logic is decomposed into smaller, distinct units of logic, called services. Individually, these units may be distributed, yet they are autonomous and isolated from each other. These services communicate with each other by exchanging messages through well-defined interfaces [6]. Fig. 4 represents the different SOA services and their respective interfaces and these services communicate with each other by exchanging messages through well-defined interfaces.

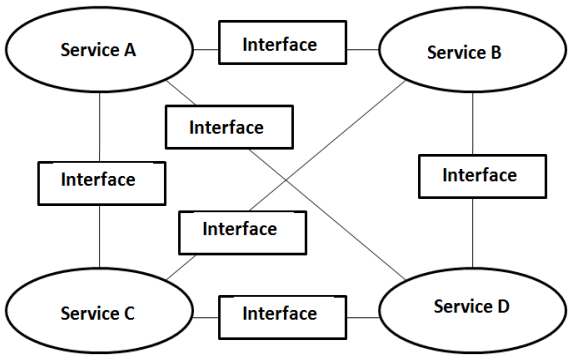

Fig 4: Services and Interfaces of SOA

As mentioned earlier, SOA and web services are the technologies among others, from which cloud computing has evolved. So developing a cloud based on SOA, the individual software application deployed therein may itself be service oriented. So, by taking the above into context, authors propose the various services and their interfaces in the cloud. A cloud may provide a variety of software applications as services which serve the business requirements of its different customers. Based on the requirements of the customers, these software services in the cloud may need to interact with each other. This interaction is accomplished through formal, standardized interfaces defined using WSDL. A cloud with cloud software services interacting through interfaces is depicted in Fig. 4. An important aspect of SOA is the separation of the service interface from its implementation [10]. As a result, a client need not be concerned with the implementation details of the service in order to use it. The service implementation performs the necessary processing. A change in the implementation of the service does not prevent the client from communicating with the service, so long as the interface remains the same.

Fig. 5 represents the various services which will serve the purpose of RES data warehouse. All these services and their interfaces will be developed by using the MDA approach. Interoperability of the services will also be taken care with the help of MDA. The very first service is Data Fetching service which will collect data from heterogeneous data sources and will pass this data to the next service namely Data Cleaning Services because this collected data is need to be cleaned. This data might have some problems related to spurious values, incomplete values, missing values etc. After the data cleaning process Data Integration service will integrate the whole data and will pass over the charge to the Data Upload Service which actually mounts the data to the data warehouse. Query Input Service will interact with the data warehouse and this interaction will be bi-directional. Query Buildup Service build will build the query based on the inputs given by the business user and finally query Output Service will produce the result which will be used as Business Intelligence.

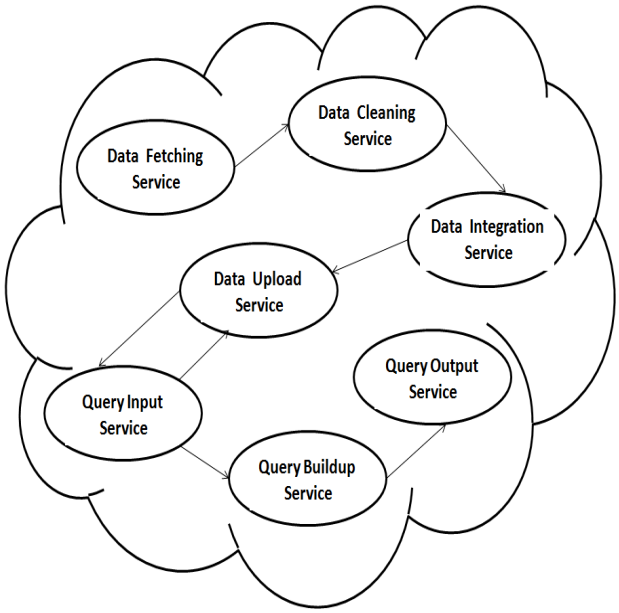

Fig 5: RES Services and Interface Model

To illustrate the concept with the help of an example, authors are taking the example of RES and identifying a service which will provide the sum of marks in a particular subject of a particular student. Sum of marks will depend upon two different types of marks – Internal aggregate marks and External aggregate marks. Internal Marks aggregation and External marks aggregation are two different services and to produce a final result is another service. So to perform this task these services need to interact with each other. In a SOA based cloud this interaction is enabled through XML based communication framework that uses SOAP messages, as depicted in Fig. 6. As depicted in Fig. 6, the end user submits a request in the formof HTML and also receive the response in the same format.

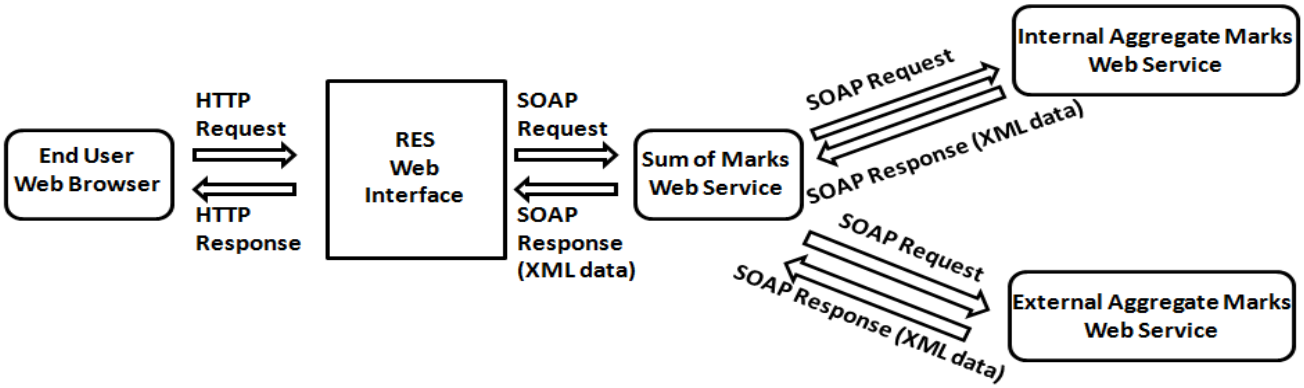

Fig 6: XML based Interaction between the Cloud SaaS

RES web interface uses the request from the client for the result evaluation and it pass a request by using SOAP protocol to the SumOfMarks web service which needs the help of others services to generate the final result and it passes the request further to InternalAggregateMarks and ExternalAggregateMarks webservices again using the SOAP protocol and the result is received include XML which further passed to the RES web interface in the same format. The HTML is finally returned to the client's browser. The advantage of using XML instead of HTML is that only raw data is required to be transferred which does not include presentation markups, thereby reducing network traffic. Also, the code required to make a request is much simpler than that required to extract data from a HTML page.

6. CLOUD SaaS AND MDA

In this section, authors present the model-driven software development approach for service-oriented integration solutions for SOA. As is evident, the technologies are constantly evolving. The technology evolution has serious impact in B2B context as it is more difficult to control the impact of change when external partners are involved. Rather than directly developing the cloud software services using available technologies, modeling them at a higher level of abstraction will decouple them from the undesired effects of technology change and enhance their longevity. An MDA based development of cloud SaaS (application) will enable defining these services in a technology-independent manner and will play a significant role in improving the quality of cloud software services, making them more robust, flexible and agile [3]. Encapsulating business logic in a manner that is independent of the technical mechanisms will formally capture the essence of the applications; and will also make it possible to reuse them in a variety of contexts [8]. Web service is a fundamental technology underlying the cloud computing paradigm; and is evolving too. Based on MDA approach, a formal, semantically rich platform independent model of the Web service capturing the information and functionality provided by it, may be defined which may then be used to generate the artifacts that support the service over some other set of technologies.

Hoyer et al. in [11] describe that the development process starts with the definition of the requirements. The next step is to model the data types and the workflow according to the defined requirements and the available legacy applications. Afterwards, model-driven transformation techniques are applied, generating formal interface descriptions by transforming the workflow modeled, by means of a UML activity diagram into a service model. Finally, a second transformation step is used to generate Web service interfaces in WSDL and corresponding data types in XML Schema. The requirements needed for designing the integration solution can be captured using manifold techniques and there is no consensus on any technique, since only the customer knows what he expects from the final software solution, but cannot express it in an unambiguously and well-formed form. So, author left it to the developer to choose any technique. Based on the defined requirements, the needed data objects are specified. Hence, the desired data objects are modeled as UML class diagrams by using Classes with typed Properties and Associations. Many UML modeling tools support the generation of SQL database schemas from UML class diagrams and vice versa.

To generate standardized Web-based interface descriptions and data types, the next step is to transform the model described in [12], which, among other details, specifies the interfaces for each legacy application and the study progress workflow itself. The transformation rules are formalized in the transformation language "Queries, Views, Transformation" (QVT) [13]. The transformation rules are described by mapping the Meta elements of the source meta model to the target meta model. Since the source and target Meta Model is the UML Superstructure [14], the transformation itself is independent from a concrete platform or technology and thus can be reused for other integration projects of the same kind.

As the final modeling step, authors transform the service model into concrete artifacts that use Web service technologies, namely WSDL [15] and XML Schema [16]. Each UML *Interface* is transformed into a WSDL document and the UML *Classes* are mapped to an XML schema [16].

The transformation rules are mainly straightforward. Each Service Interface is transformed into an abstract part of a WSDL file with exactly one port type. The port type contains the same number of operations as the UML Interface specified. The generation of the messages for the input and output of the Web service depends on the WSDL style. To finalize the integration, the required Web services have to be implemented. The generated WSDL and XML Schema files are used to create skeletons for the adapter logic implementation of the web service. For this purpose, existing approaches are applied that are part of several development tools. The final workflow is implemented in the Business Process Execution Language (BPEL) [17] and is provided as a Web service. The BPEL code can be generated from the UML activity diagram.

To illustrate the whole process, authors have taken the example of RES. Like, as mentioned earlier also that the development process starts with the requirement gathering; so in RES the major requirement is to generate the result of the students which is enrolled in a course and that course is being taught by different teacher according to different subjects. Every subject teacher evaluates the student on two different parameters namely internal marks and external marks. At the end management wants to analyze the performance of students with respect to course and subject. At the same time management also want to evaluate the performance of every teacher as well. So, after getting the exact requirements, the next step is to design the work flow of the system. Work flow simply explain how the inputs will be received, in which order these inputs will be received and how these inputs will be interrelated with another input level. The complete flow of the sequence is shown in Fig. 7. Taking the source model, authors have used a model-to-model transformation to generate service interfaces. The transformation generates a service interface for each invoked application. So, after defining the workflow of the system, the next step is to transform the work flow model to the service model of SOA as shown in Fig. 8. On the basis of the service interfaces and the data type classes a model-to-text transformation creates WSDL documents (one for each service interface) and one XML Schema document ("ResultEvaluationSystem.xsd"). The available operations of the port types in the WSDL documents match the operations of the service interface. To facilitate the reusability of the XML Schema definitions the ResultEvaluationSystem.xsd file is imported into the "types"

section of each WSDL document. Fig. 9 illustrates the generated artifacts and the import of the central XML Schema definition at the bottom.

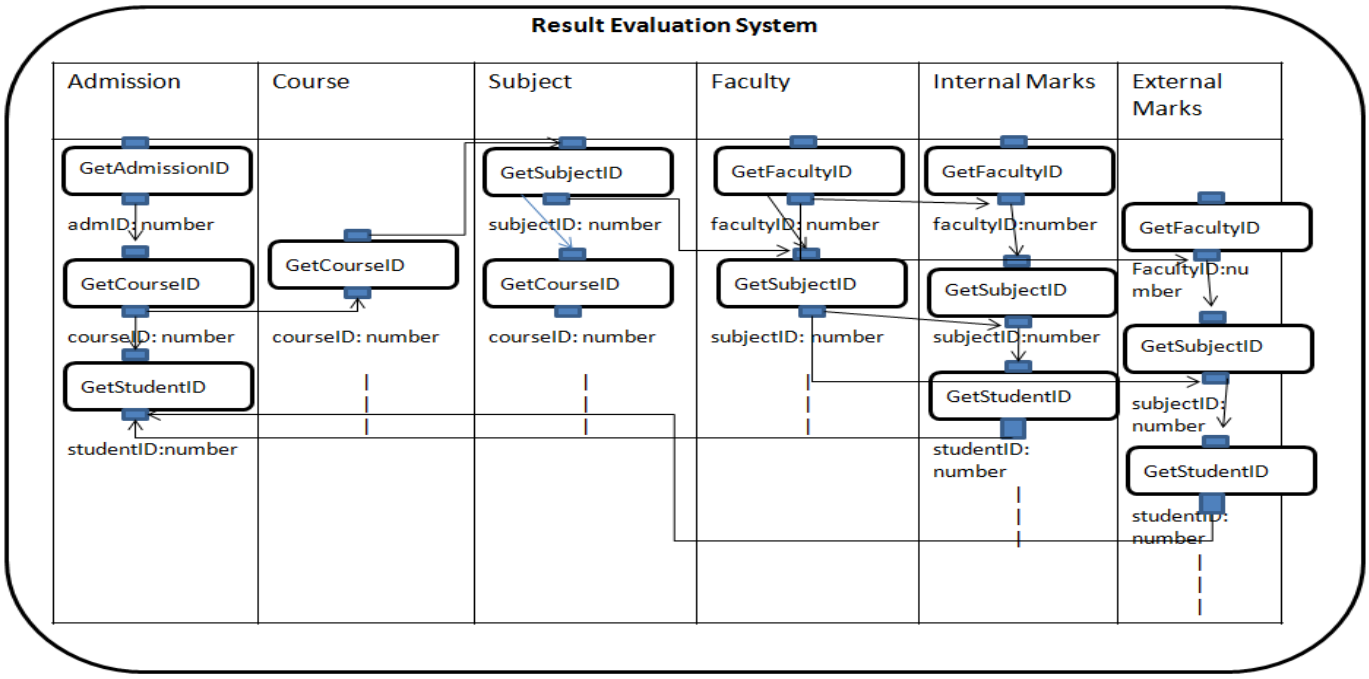

Fig 7: Workflow Diagram of RES

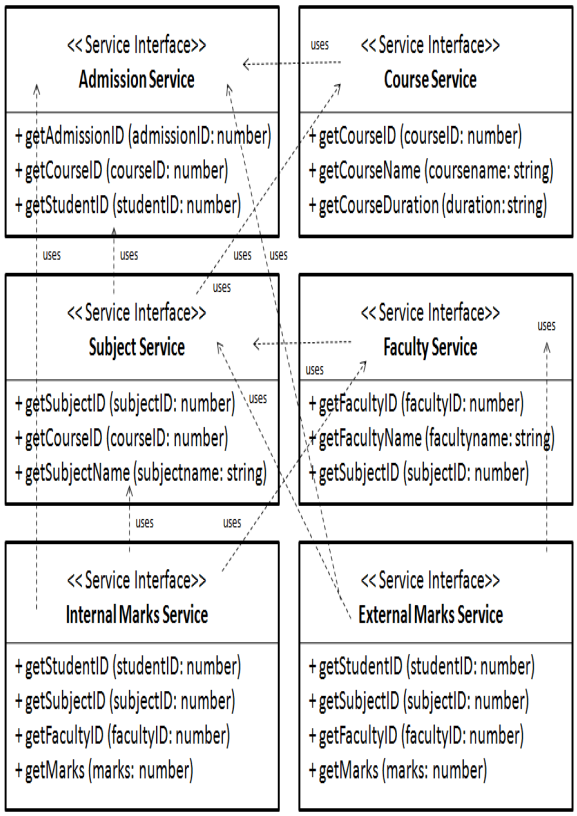

Fig 8: Transformation to Service Model

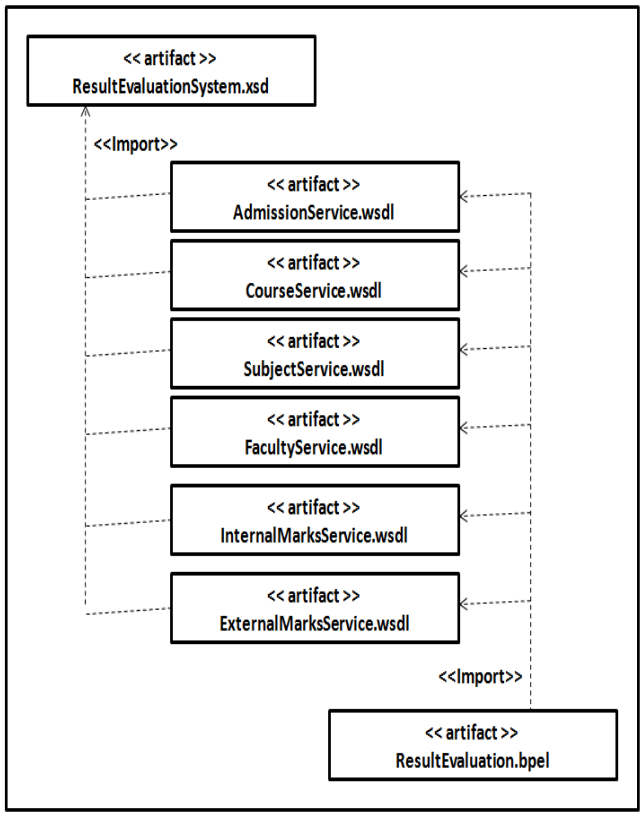

Fig 9: Code Artifact (XSD, WSDL, BPEL)

```
<wsdl: definition name="ResultEvaluation" tagetNamespace="...">
 <wsdl: type> <xs: schema ...>
   <xs: import schemaLocation= "ResultEvaluationSystem.xsd"...>
   <xs: element name="getAdmissionID" > <xs: complexType> <xs: sequencce>
    ...

 </xs: schema> </wsdl: types>
 <wsdl message name= " get admission ID request">
     <wsdl part name= "parameters"  element= " tns:getAdmissionID"/>
 </wsdl: message>
 ...

<wsdl: portType name= "ResultEvaluationPortType">
  <wsdl: operation name= "getAdmissionID">
      <wsdl: input message= "tns: getAdmissionIDRequest"/>
      <wsdl: input message= "tns: getAdmissionIDResponse"/>
  </wsdl: operation>
...

</wsdl: portType>
...

</wsdl: definitions>
```

Fig 10: Business Process Execution Language (BPEL)

Finally, the generated WSDL documents are used to create base. Authors have implemented the adapter logic of the required Web services. The study progress process itself is implemented in the Business Process Execution Language (BPEL) [17] according to the UML Activity. Authors have used an XSL transformation to generate XHTML from the tree map data structure defined before and Fig 10 gives the result of the engineered solution, showing a late prototype of the Result Evaluation system.

7. CONCLUSION AND FUTURE WORK

In this paper, authors proposed a MDA approach to design business intelligence services for cloud SaaS Model of SOA using cloud based Data Warehouse. Example taken as an illustration in this paper outlined how a service-oriented integration of existing distributed legacy applications can be realized through model-driven development approach. Our approach supports the developers in creating an appropriate integration process. Existing legacy applications and their data types in use are identified. Due to the bottom up nature of our development approach, unnecessary data transformation can be avoided and functionality that can be executed in parallel can be identified. Afterwards the integration workflow, which is modeled using UML activity diagrams, is automatically transformed into a technology independent service model. This service model can be further refined and transformed into Web service interfaces and XML Schema definitions. Thus, only functionalities that are required for the solution are exposed as Web services. The application of the standardized modeling language UML and of Web services, which allow integration in a standardized manner, allows the usage of wide spread modeling and development tools. The model-driven development approach targets a high level of formalization. And therefore supports automatic transformations of models into more concrete models or code, which helps to avoid misunderstandings and reduces errors during the development process.

8. REFERENCES

[1] Sharma, R. and Sood, M., 2011, Cloud SaaS and Model Driven Architecture. In Proc. of International Conference on Advanced Computing and Communication Technologies (ACCT 2011), ISBN: 978-981-08-7932-7.

[2] [Online] http://www.wikinvest.com/concept/Software_as_a_Service.

[3] [Online] http://www.neevtech.com/services/business-intelligence/.

[4] [Online] http://www.omg.org/mda/.

[5] Sharma A. and Sood, M., 2013, Exploring Model Driven Architecture approach to design Star Schema for a Data Warehouse. In Proc. of Advances in Engineering and Technology Series, Vol. 7, published by Elsevier, ISBN: 978-81-910691-6-3, pp. 466-471.

[6] Sharma, R., Sood, M. and Sharma D, 2011, Modeling Cloud SaaS with SOA and MDA. In Proc. of the International Conference on Advances in Computing and Communications (ACC 2011), Vol. 190, Part 5, pp511-518, DOI: 10.1007/978-3-642-22709-7_50.

[7] Thomas, E., 2005, Service Oriented Architecture: Concepts, Technology and Design, Pearson Education.

[8] Frankel, D. and Parodi, J., 2002, Using MDA to develop Web Services, [Online] http://www.iona.com/archwebservice/WSMDA.pdf

[9] [Online] http://bluforcloud.com/2014/cloud-appropriate-data-warehousing analytics/

[10] Mahmood, Q.H., 2005, Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration, Oracle Technology Network.

[11] Hoyer, P., Gebhart, M. Pansa, I., Dikanski, A. and Abeck, S., 2010, Service Oriented Integration Using a Model Driven Approach, International Journal on Advances in Software, vol 3 no 1 & 2.

[12] Emig, C., Krutz, K., Link, S., Momm, C. and Abeck, S., 2008, Model-Driven Development of SOA Services, Cooperation & Management, Universität Karlsruhe (TH), Internal Research Report.

[13] [Online]. http://www.omg.org/spec/QVT/1.0

[14] [Online]. http://www.omg.org/cgibin/doc?formal/09-02-02

[15] [Online]. http://www.w3.org/TR/wsdl20/

[16] [Online]. http://www.w3.org/TR/xmlschema11-1/

[17] [Online] http://docs.oasisopen.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html