Chapter 2: Preliminary Study

I have mentioned a few important papers in this chapter which have relevance in my work of study. There are a few other related papers that have been mentioned in bibliography. The mentioned papers are on various topics on SOA, DDB, software tool development, software performance metrics, cloud services, current technologies etc. This chapter also contains information about software tool development technologies for their usage and relevance in current times. The literature review in these broad areas defined my topic under study at various stages. Along with these areas, I also have done a study of various DDB administration tools and their architecture. As the information is focused in topic but vast in content, I have mentioned this in a separate chapter as a study of DB or DDB administration tools in chapter 4. The resources referred are also mentioned in bibliography. I have also implemented a SOA based tool using open source technologies based on my study. The papers / works / chapters / articles that proved important for study along with their essence have been mentioned below.

The importance and popularity of software evolution increase as more and more successful software systems become legacy systems as mentioned by Tong Li. Therefore, the software evolution process, the inter-discipline of software process and software evolution, becomes a key area in software engineering. He also mentions that an approach to measuring software evolution processes should be proposed. The metric objects should include: interaction, efficiency, concurrency, operability, repeatability, decomposability, reach-ability etc to name a few. He says that in doing so the metric products will greatly promote the design, analysis and improvement of software evolution processes. The importance of software engineering and development in small independent services for a SOA based system, is truly interesting and motivating.

The vast and important work done by Thomas Erl in this area for introducing the design principles that comprise the service-orientation design paradigm and further exploring various aspects and effects of applying service-orientation in the real world is very useful for developing and maintaining complex software systems of current times. With such a broad idea in view and SOA as the broad topic, my preliminary study continued. Most of the resources referred were online refereed journals and articles. There are a number of researchers and authors who have made significant contributions in this area as mentioned in this chapter. The initial focus has been software engineering based topics like SOA, Web Services, eXtensible Markup Language, Cloud Services etc. along with the problems faced during software development and maintenance. Later on I narrowed on software tool development using open source technologies and its overall performance evaluation.

I got a very good opportunity to study an immensely related paper titled “SODDA – A Service-Oriented Distributed Database Architecture” by Breno Mansur Rabelo and Clodoveu Augusto Davis Jr. published in ICEIS 2008 - Proceedings of the Tenth International Conference on Enterprise Information Systems, Volume DISI, Barcelona, Spain, June 12-16, 2008 with ISBN 978-989-8111-36-4. The authors have taken the service oriented database architecture (SODA) as a reference and proposed their work related to service oriented distributed database architecture (SODDA). The paper mentions that after the advent of interoperability standards, such as XML, and of service-based networking which laid the ground for allowing new alternatives for the implementation and deployment of distributed databases. The paper proposes the adoption of elements from service-oriented architectures for the implementation of the connections among distributed database components, thus configuring service-oriented distributed database architecture. SODDA merges DDBMS concepts with new technologies and initiatives associated to SOA and the Internet. SODDA uses Web services to coordinate operations among distributed database nodes. Each node includes a Web service to coordinate the local database, and which is capable to respond to a client data provider, called the SODDA Hub, when the node receives requests for queries or other database operations. SODDA Hub can be seen as a common connectivity middle ware. It runs on the client’s side. All operations that are submitted to database nodes are conducted through the SODDA Hub, which is also capable of accessing the distributed database’s global catalog. In SODDA, the global catalog is replaced by a catalog service, which provides information about the location of Node Wrappers. Additionally, they have proposed services like Distributed Query Processing (comprising of various processes), Distributed Transactions Manager (DTM), the Replicated Data Manager (RDM), and the Database Recovery Service (DRS). To validate proposed architecture, they have implemented a SODDA prototype as a Microsoft .NET data provider. They conclude that SODDA intends to use some of the most interesting features of SOA to implement distributed databases. Expected benefits include easier implementation, lower communications costs, and greater accessibility.

"AutoGlobe: An Automatic Administration Concept for Service-Oriented Database Applications" published by Stefan Seltzsam, Daniel Gmach, Stefan Krompass, Alfons Kemper in Proceedings of the 22nd International Conference on Data Engineering (ICDE’06) IEEE Computer Society. This paper asserts the fact that future database application systems will be designed as Service Oriented Architectures (SOAs) like SAP’s NetWeaver instead of monolithic software systems such as SAP’s R/3. They mention a problem that the decomposition in finer-grained services allows the usage of hardware clusters and a flexible service-to-server allocation but at the same time increases the complexity of administration. In case, overloaded service instances are detected, the situation is remedied by either starting new service instances or by moving instances to more powerful servers. With this approach of automatic runtime adaption, the authors mention a tool – AutoGlobe that reduces administrative overhead and achieves a reduction of total cost of ownership as either more users can be handled with the existing hardware or because less hardware is required initially. One basic aspect of AutoGlobe is that services are virtualized, i.e., they are not running on a fixed server. Thereby, available resources are shared between all services as appropriate for a particular situation. The paper later on presents the architecture of AutoGlobe, which is based on the team of authors' ServiceGlobe platform for location-independent execution of Web services. They next describe the foundations of fuzzy controllers and the fuzzy controller of AutoGlobe. Finally, they mention the simulation study results. For implementation, the simulation environment models a realistic SAP system with the corresponding hardware. The authors have described simulated services and servers using their own declarative XML language like real existing services and servers. This paper gives importance to the architecture of the system by showing three tier architecture of the simulated SAP installation. The three layers are the presentation layer, the application server layer, and the database layer. In this system, the end-users communicate with the SAP installation using clients in the presentation layer. Their software system installation comprises three subsystems in the application and database layer: Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Business Warehouse (BW), each running its own dedicated database and central instance.

I studied XML technology for its relevance in my work and published a paper[1] *in an International journal. This paper is about removing the diagramming ambiguity that may arise in case of representation of Data Flow Diagrams (DFD. In the world of Information Technology, the working of an information system is well explained with the use of DFD. DFDs are one of the three essential perspectives of the Structured Systems Analysis and Design Method (SSADM). The sponsor of a project and the end users are briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. But, various practical problems exist with the representation of DFDs. Different tools are available in the market for representing the DFDs. These tools are user friendly and based on the object oriented features. The diagrams drawn using*

[1] "XML Based Representation of DFD" in International Journal of Advanced Computer Science and Applications (IJACSA). This paper has received citation in “Data Flow Sequences: A Revision of Data Flow Diagrams for Modelling Applications using XML” by James PH Coleman

these tools can be sent over the network for communicating with others. On the other hand, the XML is platform independent, textual information and is totally extensible. XML is structured and is an excellent way to transfer the information along with the meta data over the network. XML can be used to transfer the information related to DFDs, thereby removing the problems related to understanding the diagrammatic notations and concentrating more on the information flow in the system. This paper is aimed at understanding the problems related to DFDs and representing it in XML format for their storage and communication over the network. The discussion is divided into four main topics –introduction to XML and DFD, problems related to DFD, an XML representation for DFDs and finally the conclusion.

The paper titled “System Development As A Wicked Problem" by Raymond T. Yeh published in International Journal of Software Engineering and Knowledge Engineering (IJSEKE), discusses problems and principles based on experiences from good practices in the field to deal with the attributes of wicked problems. As a result of these principles, new process paradigms are needed. Automation of the alternative process and further research on downsizing mainframe applications to distributed applications are also discussed here. According to Albert L Lederer (University of Kentucky, USA) and John Benamati (Miami University, Ohio, USA) in a book titled 'Strategies for Managing Computer Software Upgrades' and edited by Neal G. Shaw; the challenges for implementing new information technologies and their effects need to be understood for their implementation. They have discussed their views in second chapter of the book related to continuous development in hardware, software and data management tools and the consequent challenges that have to be faced by organizations. The most important is that the technical people (project managers, system analysts, database administrators etc) are not aware of the challenges themselves as they have not used them before and are yet unaware of their end results. The maintainability of software as per the demands of the emerging new technologies burdens not only the IT people but at the same time leaves the customer frustrated due to poor software quality. In “Decomposing Composition: Service-Oriented Software Engineers,” M. Brian Blake proposes an incremental development approach that integrates software engineers with subject matter experts as a first step toward development environments for service-centric software systems. Different stakeholders must have different roles in the development process, and Blake identifies two in his article: service-oriented software engineer and service-oriented software integration engineer. He concludes that by creating new life cycles and better educating software engineers about SOA methodologies will enhance next-generation distributed enterprise systems. Industry and research organizations will need to assimilate these enhanced software life cycles and evaluate their effectiveness at the end of each phase which may be different for different domains if the heterogeneity of service-centric environments is considered. However he also mentions that strategic models should be developed that help architects navigate the variations across domains while preserving the modularity of SOA environments.

The white paper on “Architected to Last: The Expanding Relevance of Service-Oriented Architecture” sponsored by: IBM, authored by Stephen D. Hendrick in April 2011; explores SOA's relationship with and relevance to IT and business. The paper mentions the need for SOA, SOA key constructs, role of SOA in enterprise and the observed benefits of SOA. It further identifies the challenges and opportunities for SOA. To start with, SOA in itself was an evolution related to the distributed computing and componentization. The key principles that guided the development of SOA include service modularity, interoperability, standardization, identification, and provisioning. These principles remain operative today because they reflect an elegant balance between being specific enough to drive adoption yet abstract enough to be extensible. The evolving software marketplace, which is transitioning from software to services, provides a context for better understanding the role of SOA in application development and deployment (AD&D) today and over the next five years. The author mentions that a tactical emphasis on data and application integration is often what brings organizations to SOA but the real value of SOA is realized as organizations standardize their approach to application development and adopt modern application development techniques that share a service and business process focus and never lose sight of the architectural discipline that SOA brings to IT activities. In practical terms, SOA is seeing use in businesses to address integration tasks. The real benefit of SOA is that it is able to appeal to organizations on any level. There is value for both IT and the business, the architecture of SOA allows it to relate effectively to both legacy and modern applications, and SOA can support a wide array of activities ranging from simple system-to-system interoperability to being the underlying architecture for new system design with high levels of component reuse. A more recent implication of this transition from development to deployment is cloud computing. The most compelling characteristics of cloud computing are its shared service and self-service models. *The information given in this paper proved very useful to realize that SOA approach would be very appropriate for me to continue my study on my topic and also the fact that I got introduced to the importance of cloud services.*

"Software-as-a-Service Revenue Models" by Arto Ojala published in the IEEE Computer Society May/June 2013. The author gives an overview of the three main revenue models, focusing on the advantages and disadvantages for SaaS providers and their customers along with the identification of the most effective revenue model for particular situations. In accordance with Michael Armbrust and his colleagues, the author has considered cloud computing to cover not only software applications delivered through the Internet but also the hardware and system software that data centers use to provide these services. The work in this paper mentions Infrastructure as a service (IaaS) for providing computation and storage capacity, platform as a service (PaaS) for providing software development tools and an application execution environment, and SaaS that provides applications on top of a PaaS, IaaS, or private data center. The author has used the term “revenue model” in an operational sense, referring to how a firm collects revenue from its customers and has also discussed the advantages and disadvantages of traditional software licensing in comparison to pay-per-use and software renting models. An assumption made is that the SaaS providers sell the software using a public cloud, although in some cases, the SaaS providers can rent the software for use in a customer's private cloud. On the other hand, smaller companies with limited financial resources might find it more cost-effective to use rental or pay-per-use models. Based on the comparison of revenue models from the software provider's viewpoint and the criteria for selecting the most advantageous model for software purchase and the level of impact, the author suggests that if the target segment for the software is narrow, traditional licensing or software renting can work well. By contrast, the pay-per-use model is better suited to a large customer group, with further possibilities for positive network externalities. From the customer's viewpoint, software renting can provide a trade-off between traditional licensing and the pay-per-use model. It has the particular advantage of making it possible to estimate the software costs and to buy the software without separate budgeting or complicated decision processes. *I found this paper relevant in my work. I have used a cloud service – vpsdime for the purpose of renting and availing cloud space for implementing database services. I found the discussion useful for the purpose of software tool implementation on a cloud.*

In "Realizing Service-Centric Software Systems" published by Olivier Nano and Andrea Zisman in IEEE Computer Society 0740-7459/07/2007; the authors have mentioned that to achieve adaptable, flexible, interoperable, and maintainable service-centric software systems, there is a need to define processes and create methods, tools, and techniques to support cost-effective application development and use. There must be interoperability among the tools and approaches to support development life cycles; service discovery and composition; service deployment, binding, and monitoring; quality of services and security; service-level agreements; and service description languages. They have presented five articles that tackle service-centric software systems. The first by Philippe Lalanda and Cristina Marin, is on two tools to support service composition. The first tool supports application specification on the basis of syntactic descriptions of concrete services. The second tool allows domain-specific environments to be developed according to rules specified by domain experts. The second article by Danilo Ardagna, Marco Comuzzi, Enrico Mussi, Barbara Pernici, and Pierluigi Plebani describes a flexible approach for executing Web services. They base their approach on business processes that support adaptation both at design time and runtime, and they include several case study applications. The third article titled “Improving Web Service Discovery with Usage Data,” Aliaksandr Birukou, Enrico Blanzieri, Vincenzo D’Andrea, Paolo Giorgini,and Natallia Kokash present a system for improving service discovery on the basis of data provided by service clients rather than service owners. The fourth article by Tomas Vitvar, Michal Zaremba, Matthew Moran, Maciej Zaremba, and Dieter Fensel describe architecture to support service integration and interoperation on the basis of semantic languages and service models. This article also surveys Semantic Web technology for Web services. In the fifth article titled “Decomposing Composition: Service-Oriented Software Engineers,” M. Brian Blake proposes an incremental development approach that integrates software engineers with subject matter experts as a first step toward development environments for service-centric software systems.

The article on “Service-Oriented Architecture and Microsoft .NET”, posted by John Charles Olamendy on January 21, 2006; provided an overview of how Microsoft .NET and related technologies support the goals and principles of Service-Oriented Architecture. SOA is an architecture which evolved from the distributed and the components architectures and has become the new one to solve complex problems around the enterprise environment. The article gave an insight as to the approach of Microsoft .NET and its alignment according to SOA strategy. The importance of SOA approach and hence the alignment of the development environment according to this approach is aptly explained in this paper.

I was motivated to explore and understand other topics like design patterns and its applicability in .NET. I published an article titled “From Data Handling to Presentation of Data: Encapsulating the App.config in .NET Applications through Design Patterns” in International Journal of Computer Applications (0975 – 8887) Volume 3 – No.2, June 2010. This article is about studying the .NET based application development and the implementation of the design *patterns. Various papers[2] were referred for this purpose to show how certain database related processes can be considered as regular patterns in order to simplify the application development and project management and concentrate more on logic and presentation layer than the actual handling of data. After the introduction of Object Oriented concepts, systems were developed keeping in mind not only the data and its behavior but also laying stress on the object oriented analysis and design methodologies. Currently, there is a need to have a pattern or a service based approach to most of the software tools and applications to introduce loose coupling and ease of maintenance. In the problems related to .NET and database related operations, there can be a possibility of a pattern or a service which handles the data. This service can be thought of as an entity above DAL (Data Access Layer), to relieve the developer of syntax errors and the data format errors.*

In the article titled “Successful Software Product Line Practices” by John D. McGregor, Dirk Muthig, Kentaro Yoshimura and Paul Jensen; published in IEEE Computer Society 0740-7459/10/2010, the authors have cited some software product line differences and commonalities followed by many of the aspects of software product line development that they have identified. The authors have also mentioned the successful software product line organizations like Cummins, Hitachi, Hewlett-Packard, Overwatch, SystemForge etc in order to describe their software development approach and methodologies; architecture and their challenges. They mention that a software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way in place. Organizations adopting product development strategies that include a software product line have achieved impressive results, reducing product cycle time and increasing productivity by an order of magnitude.

[2] Various papers and articles [34] to [43] of bibliography were referred for the design patterns

"ReST-Based Web Access to Learning Design Services" by Juan Manuel Dodero and Ernie Ghiglione published in IEEE Transactions On Learning Technologies,Vol. 1, No. 3, July-September 2008. This paper proposes a Representational State Transfer (ReST) architectural style of accessing a learning service and its constituent resources from a Web-based environment. The authors have also described a methodology to guide the design of learning service access. They have applied a method to integrate operations of a generic wiki service in the client interface of a learning activity and validate on actual wiki services providing ReST-based programming interfaces. The current interest in Web services appears to be deeply influenced by the architectural style as given by M. zur Muehlen, J.V. Nickerson, and K.D. Swenson. The paper starts with defining ReST as an architectural style based on the HTTP protocol that describes a navigational, resource-oriented style of design. This means that, for each resource, a URI must be provided to let client applications navigate from it to other resources. ReST is usually confronted by the Service-Oriented Access Protocol (SOAP), which actually represents a procedural, function-oriented style of design in which the service is provided with a set of procedures or functions that can be called to access the service public operations. The paper mentions an interesting solution that interposes a software wrapper between the learning design player and the service provider that instantiates and enacts the service, multiplexes service calls, and compiles interesting data from the service operation. This generic service integration approach enables switching the service if required and solves the trade-off about fine versus coarse service call granularity needed for the learning design being aware of individual operations, since the wrapper is responsible for data compilation in either case. The proposed ReST solution is simple, since it is based on few principles and well-defined operations, scalable, since it relies on stateless protocols and distributed state, and layered, allowing for any number of intermediaries between activities and services. The service access requirements of open integration, fine grain control, and user interface decoupling are fulfilled by the architectural style and design methodology. Providing activities with a user interface to access ReST-based learning services is still a matter of configuration. The authors have only considered the integration of service-specific primitive operations within learning design systems. They suggest that the actual list of functionalities that need to be integrated must cover the complete life cycle of the service, from authoring and enactment to monitoring to interoperability facilities. This is an ongoing effort that has been undertaken in the FLEXO project, funded by the Spanish Ministry of Industry and Commerce, and realized in cooperation with several Spanish companies and universities and the LAMS Foundation. This paper is very informative for web services and the issues involved therein. The web services are an important aspect for SOA.

In Tsinghua Science And Technology, a paper titled “Methodology for Web Services Adoption Based on Technology Adoption Theory and Business Process Analyses” by An Liping, Yan Jianyuan, Tong Lingyun; the paper presents a methodology that supports the deployment of web services to meet strategic company goals. Two of the most important advantages of the web services architecture are its openness and modularity. Web services are currently seen as the primary solution for dynamic integration of business functionality across the web. Web services are an architectural innovation that has the potential to alter the means by which organizations utilize information systems. The methodology presented in the paper helps organizations launch web services initiatives to make timely adoption decisions and formulate appropriate plans. It also suggests that strategies, business areas, and functions within an organization should be considered based on the existing organizational information technology status during the process of adopting web services to support the business needs and requirements. The authors have referred concepts by Chen who created a model for XML and web services technology adoption to analyze the factors affecting various adopters at various stages of e-business standards adoption. Chen has considered that companies with higher levels of IT sophistication required fewer resources to adopt web services technologies. The paper also mentions the situations for using the web services and the three types of integration used by companies ie business to business (B2B) integration, internal integration, and multi-channel implementation via web services.

At about the same time, I worked with a colleague from the management faculty and we both published a paper[3] "Transforming the Point of Sale to Point of Service - Applying SOA in the Indian Retail Scenario" in Journal of Business and Retail Management Research. In this paper we have organized the content into various sections like introduction to retailing, literature review where important papers[4] have been referred; trends in retailing and scope for Information Technology, SOA, a case study, SOA based solutions and finally the conclusion. Unorganized retailing is by far the prevalent form of trade in India – Constituting 98% of total trade, while organized trade accounts only for the remaining 2%. The retail industry in India is of late often being hailed as one of the sunrise sectors in the economy. However, in order to find its feet in the Indian environment a lot needs to be done in terms of efficiency enhancement. This paper is an attempt in exploring the benefits of applying SOA in the organized retail industry, for achieving business efficiency by transforming the point – of – sale into a point – of – service through SOA and thus ensuring maximum customer satisfaction. The paper aptly addresses the various aspects involved in retail and the IT support needed in its efficient functioning. It also shows the need for the IT related additional services in the current retail scenario by considering two supporting case studies and the possible solutions for taking care of the problems stated by the case studies. Customer sustainability and customer satisfaction can be achieved by combining the Information Technology related SOA concept with the Retail Management concept and converting the point-of-sale to point-of-service. The existing applications that are used by most of the retail outlets, need not be changed but only needs to incorporate new service(s) for handling the problems of customers and salespersons and ensuring smooth functioning of the mall/outlet for a better customer experience. A better customer experience will ensure customer sustainability which in turn boosts the profitability of the business venture. In the long run, such an approach goes on to attract more and more customers.

[3] "Transforming the point of sale to point of service - applying SOA in the Indian retail scenario" in Journal of Business & Retail Management Research

[4] Papers from [29] to [33] in Bibliography were referred for literature review purpose

"Cost Analysis of the Service Migration Problem between Communication Platforms" in 2008 IEEE published by Dayyan Shayani, Carmen Mas Machuca, Monika J¨ager and Andreas Gladisch. The paper is about service migration which is the process of moving currently active services from a telecommunications transport platform to another. In this paper, the source platform is referred to as old platform, while the destination is referred to as new platform. Because emerging services pose special requirements towards the networks, operators have not only to extend the capacity of their transport networks, but also need to optimize the architecture for differentiated services with advanced packet switching technologies, referred to as Next Generation Networks (NGN). During network evolution with disruptive technology changes, a general problem faced by network operators is the service migration of active services from an existing platform into a newly setup platform. Increased competition in the telecommunications industry pushes network operators to rationalize their total costs. Models and simulations appear as major tools aiming at better understanding the expenditures and to allow a more realistic network planning. However, their focus was, in the past, mostly limited to infrastructure investments. Their current work investigates strategies for service migration at minimized overall operational costs. A novel service migration cost model based on queuing theory and hill climbing optimization is provided in order to tackle the employee allocation optimization problem. Based on this framework, a thorough techno-economic analysis of costs and benefits is presented. They have presented four study cases based on this model. They are (i) the savings implied by the migration are taken into account to predict the most beneficial migration scenario among a number of parameters; (ii) the migration order (i.e. the criterion to choose which services should be migrated first) is discussed; (iii) the impact of services with different migration conditions (e.g., different marketing and installation procedures) is analyzed; (iv) personnel contracting modalities (e.g., full-time or part-time own staff or sub-contracted staff) are also compared. The paper gives an understanding of the issues related to service migration and their conclusions. The main findings of the case studies presented are that migrating running services between platforms is a costly project that should be planned carefully for minimizing its costs. A higher number of employees performing the migration shortens the overall migration time, consequently reducing the costs of maintaining parallel platforms running. In contrast, the shorter the migration time is, the more services need to be migrated before they reach their end of life, which increases the overall migration effort. Also, the order in which services are migrated, regarding their life-time, have a strong influence on the total number of services migrated and the migration costs.

"Automating Cloud Services Life Cycle through Semantic Technologies" published in IEEE Transactions On Sercies Computing, VOL. 7, NO. 1, January-March 2014 by Karuna P. Joshi, Yelena Yesha, and Tim Finin. The authors describe a new integrated methodology for the life cycle of IT services delivered on the cloud and demonstrate ways in which it can be used to represent services and service requirements and so automate service acquisition and consumption from the cloud. They have defined an integrated ontology for processes needed to automate IT services life cycle on the cloud. They also mention that this is the first such effort, and it is critical as it provides a holistic view of steps involved in deploying IT services. Their focus for this framework is the life cycle for virtualized cloud services, where the services are composed of combining preexisting components. Hence, this life cycle does not include any requirements analysis or design phases. They have divided the IT service life cycle into five phases of requirements, discovery, negotiation, composition, and consumption and detail each phase and describe the ontologies that they have developed to represent the concepts and relationships for each phase. For demonstration, they have described a cloud storage prototype that has been developed by them based on their methodology that can be referenced by organizations to determine what key deliverable they can expect at any stage of the process. The tool (they have shown a user interface for discovering cloud storage service by specifying constraints) demonstrates how their methodology can be used to significantly automate the acquisition and consumption of cloud-based services thereby reducing the large time required by companies to discover and procure cloud-based services. As part of their ongoing work, they are working on automating complex service negotiation process where the negotiation is on a range of values for a constraint. While developing the ontology, they have referred to NIST's (National Institute of Standards and Technology) cloud computing reference architecture to identify the key stakeholders in the life cycle. They have divided the virtualized service life cycle on a cloud into five phases. In sequential order of execution, they are requirements (where two main classes are the Specification class and the "Request for Service" class), discovery (for service certification), negotiation (for defining the service data, delivery mode, agent details, quality metrics, and cost of the service), composition (to combine the various components into a single service) and consumption (where consumer will require tools that enable service quality monitoring and service termination if needed).

"Comparison of Approaches to Service Deployment" by Vanish Talwar, Qinyi Wu, Calton Pu, Wenchang Yan, Gueyoung Jung and Dejan Milojicic; published in Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS'05). The authors compare various situations and software tools in context of development and deployment. They conclude by relating the maintainability with the number of lines of code. They start with mentioning that the scale and complexity of current IT systems and services makes them increasingly difficult and expensive to administer and deploy where a service is defined as a standalone software component that encapsulates and presents useful functionality, is installed in a computing environment, and can be composed into an overall system or application. Services in this broad sense include business services as well as modules such as transaction services or databases. They can be realized as Web or Grid services or even as component services in an operating system. The authors analyze service deployment where deployment means an action to download, configure, activate, and maintain the life cycle (e.g., react to failures, terminate, and restart) of services. They quantitatively and qualitatively compare the approaches to service deployment in terms of scalability, complexity, and susceptibility to change. They have identified cases where each approach is best suited along with their hypotheses. They have also given examples of various software tools like Nixes (a tool used to install, maintain, control, and monitor applications on PlanetLab), SmartFrog (a framework for service configuration, description, deployment, and life-cycle management) and Radia (a change-and-configuration management tool, employs a model-based approach). Their hypotheses are: First, manual deployment is likely to be well suited for simple, small scale deployments because it has small barrier to entrance. Second, for larger scale, yet still simple deployments, script-based deployment presents advantages over the manual approach, even though scripts may require learning the scripting language and then the development and maintenance of scripts. Third, for large scale, complex deployments, a language-based approach such as SmartFrog should be the best though it requires some additional investment in understanding the complexity of the framework and developing the template and configuration descriptions consisting of dependencies. Finally they mention that the environments involving a lot of dynamic, run-time changes are best suited for model-based deployment. Their results indicate that the number of steps and the number of lines of code for deployment are reduced with the introduction of more sophisticated deployment tools. Maintainability and document-ability are proportional to the number of lines of code. Manageability is proportional to the number of the lines of code modified and number of steps added in response to changes in system configuration. All these findings have benefited me for the purpose of my work.

I was fortunate to study an important paper on “Global Viewing of Heterogeneous Data Sources” by Silvana Castano, Valeria De Antonellis and Sabrina De Capitani di Vimercati published in IEEE Transactions on Knowledge and Data Engineering. *This paper provided a very good orientation for tool development that I was planning for my work. I also got an insight into the reference data model and the way the paper explains the situation using mathematical concepts.* The paper mentions the problem of defining global views of heterogeneous data sources to support querying and cooperation activities. The problem is becoming more and more important due to the availability of multiple data sources within complex organizations and in global information systems. Global views are defined to provide a unified representation of the information in the different sources by analyzing conceptual schemas associated with them and resolving possible semantic heterogeneity. In the paper, the authors have proposed an affinity-based unification method for global view construction. They have also explored the concept of clustering to group all schema elements that have affinity in the source schemas. In the method: 1) The concept of affinity is introduced to assess the level of semantic relationship between elements in different schemas by taking into account semantic heterogeneity; 2) Schema elements are classified by affinity levels using clustering procedures so that their different representations can be analyzed for unification; 3) Global views are constructed starting from selected clusters by unifying representations of their elements. The authors have also described their experience of applying the proposed unification method and the associated tool environment ARTEMIS on databases of the Italian Public Administration information systems. The authors have then presented an affinity-based unification method for the identification and reconciliation of elements with affinity in heterogeneous data schemas. The proposed method can be fruitfully exploited with multiple data sources in order to reach the intended objectives of facilitating the unification process in cooperation scenarios, improving query processing and supporting formalization efforts in heterogeneity analysis and resolution. The experimentation in the approach has been conducted in the Italian Public Administration domain on a set of entity-relationship conceptual schemas of the databases of the Ministry of Labor and of the Ministry of Finance. They mention that the extensions of the approach will be in the direction of a deeper investigation of the issues related to extensional aspects in the unification method and of the analysis of semi structured data sources, such as the Web and XML document sources, to provide them with a uniform query interface. They have developed a prototype tool ARTEMIS in Java to support the process of creating global views. ARTEMIS provides functionalities for affinity-based analysis and clustering of source schema elements and for the unification of candidate clusters into global views. The work in this paper was beneficial to me for representing my work related to the developed tool; to be presented in a mathematical model with the help of relational algebra expression. *I have narrowed down on three sub processes that are crucial to the working of my tool. These sub processes have been represented considering a relational database management system and is discussed in chapter 4.*

"An Enhanced Extract-Transform-Load System for Migrating Data in Telecom Billing" published in IEEE ICDE 2008 by Himanshu Agrawal, Girish Chafle, Sunil Goyal, Sumit Mittal, Sougata Mukherjea. The authors have put forth their discussion in light of Data Migration project undertaken by IBM Global Services for a large European Telecom Service Provider. The paper starts with the importance of data migration that has become a priority in many industries, spawned by a variety of business needs. Most of the existing tools for Extract, Transform and Load (ETL) process of data migration are piece-meal and do not present a complete solution. Moreover, while research has focused on the problem of schema mapping, a key step in the ETL process, most of the current algorithms do not perform well on real-world data. Their study shows that previous researchers have suggested the use of Domain Knowledge to enhance schema mapping. In this paper, the authors use domain knowledge in an innovative manner to improve schema mapping in an 'actual' industrial setting. Data Migration is an important problem in many domains and companies spend millions of dollars on such projects. In this paper the authors have presented DMT, an enhanced Extract-Transform-Load system for migrating data in Telecom Billing. Following are the major contributions of this paper: a comprehensive view of the Data Migration problem and describe an end-to-end system for ETL, utilizing domain knowledge in innovative ways to enhance schema mapping and presenting experimental results from an actual industrial setting to demonstrate the utility of domain knowledge using the authors' approach. They mention that their system combines existing tools for various stages of the ETL process and utilizes connectors, wherever applicable, to reduce the manual intervention required. To improve the accuracy of schema mapping, they have used domain knowledge to first group attributes related to each other, followed by using these algorithms to map each group on the source side to the corresponding group on the target side. Similarly, they have used the sets of related attributes to filter the mappings generated by running existing algorithms on the source and target schema. The domain knowledge for Telecom includes ontology, a thesaurus and several domain-specific rules. To evaluate their system, they have used data from a Data Migration project and achieved significant improvement in schema mapping precision. Their results validate the utilization of domain knowledge for industrial settings. They also mention that although they have focused on Telecom billing during their experiments, their techniques can be applied to other domains as well.

“Towards a Better Understanding of Software Evolution: An Empirical Study on Open Source Software” by Guowu Xie, Jianbo Chen and Iulian Neamtiu, published in 2009 IEEE Proc. ICSM 2009, Edmonton, Canada. This paper discusses software evolution is a fact of life where researchers have proposed hypotheses on how software changes, and provided evidence that both supports and refutes these hypotheses. The authors performed an empirical study on long spans in the lifetime of seven open source projects in order to explain the software evolution process. Their analysis covered 653 official releases, and a combined 69 years of evolution. They have first tried to verify Lehman’s laws of software evolution and their findings indicate that several of these laws are confirmed, while the rest can be confirmed depending on the laws’ operational definitions. Second, they have analyzed the growth rate for projects’ development and maintenance branches, and the distribution of software changes. They have found similarities in the evolution patterns of the programs under study, which resulted in constructing rigorous models for software evolution. The authors also mention a few laws related to the verification of each of Lehman’s software evolution laws on their test applications. The first law postulates that a program must continually adapt to its environment, otherwise it becomes progressively less useful. The second law postulates that as a program evolves, its complexity increases, unless proactive measures are taken to reduce or stabilize the complexity. They mention a suggestion by Lehman that the evolution of large software systems is a self-regulating process, i.e., the system will adjust its size throughout its lifetime. The next law, also known as “invariant work rate”, stipulates that the rate of productive output tends to stay constant throughout a program’s life time. The next law suggests that incremental system growth tends to remain constant (statistically invariant) or to decline, because developers need to understand the program’s source code and behavior. Another law stipulates that programs usually grow over time to accommodate pressure for change and satisfy an increasing set of requirements. Another law stipulates that over time software quality appears to be declining, unless proactive measures are taken to adapt the software to its operational environment. All the applications studied by the authors have points in their history where the development has forked into a development branch and a stable (maintenance) branch. The authors mention an important factor in program evolution; that is to understand which parts of the code change, and how frequently. Analyzing the reasons that lead to these hot spots i.e., parts that change frequently; can actually facilitate evolution. From the study, the authors furnish results that indicate that Continuing Change, Increasing Complexity, Self Regulation, and Continuing Growth are still applicable to the evolution of today’s open source software. They also mention that different branches of open source programs evolve in parallel, which confirms the parallel evolution hypothesis proposed by other researchers. *I found the paper valuable in terms of not only looking at tool development under the dynamic situation but also in considering the importance of open source technologies. The work done for handling dynamic situation in software development truly made me clear on my service oriented approach. At the same time I started identifying open source technologies for my work on tool development.*

"A Multidimensional Weighted-Attributes Framework (MWAF) For Evaluating Agent-Oriented Software Engineering Methodologies" by Abdel-Halim Elamy and Behrouz Far in IEEE CCECE / CCGEI, Ottawa, May 2006. This paper explains Agent-Oriented Software Engineering (AOSE) and is concerned with the use of agents in the development of distributed systems. The situation considered is especially in open and dynamic environments that require autonomous processes with communicative and cooperative features. The explain that for deciding what AOSE methodology is the best to adopt for developing a potential multi-agent system depends on which aspects are considered more important for the prospective application under consideration. However, till the publishing of the paper, there was no industry-wide agreement on the kinds of concepts a methodology should support. This lack of standardization made the research in this area an open issue with accompanying challenges. Evaluation is a crucial and critical task here to identify the differences between several AOSE methodologies. In this paper, the authors propose the Multidimensional Weighted-Attributes Framework (MWAF), a new framework that provides a reliable solution with accurate results based on applying state-of-the-art statistical procedures to evaluate AOSE methodologies and come up with a set of measures that help in selecting the most appropriate methodology for developing prospective agent-based applications. The authors have also proposed the Multidimensional Weighted-Attributes Framework (MWAF) as a general-purpose framework that can be adopted to evaluate software products based on the user feedback. The main idea of MWAF is to define the most common and important criteria (i.e., dimensions) of the software being evaluated, identifying the attributes describing each of these dimensions, and then evaluating each dimension through its attributes against all the potential software products that are given for evaluation. The evaluation data have been collected by assigning two parameters to each evaluated attribute: a weight to identify the importance of the attribute, and a rate to measure its strength or effectiveness. Further, they have analyzed the data by means of statistical procedures to rank the evaluated methodologies according to the effectiveness of their attributes. *I found the paper very useful in order to identify and later evaluate the performance parameters for my SOA based DDB administration tool. The authors have designed questionnaire and performed necessary statistical analysis. The paper has shown the identification of attributes depending on various factors like Agency-related attributes, Modeling-related attributes, Communication-related attributes, Process-related attributes, Application-related attributes, User perception attributes, Upgrading-related attributes and Supporting properties like ontology, security and collaborative services.*

"System of Systems for Quality-of-Service Observation and Response in Cloud Computing Environments" published in IEEE Systems Journal by Paul C. Hershey, Shrisha Rao, Charles B. Silio and Akshay Narayan. This paper presents a System of Systems (SoS) approach to enable Quality of Service (QoS) monitoring, management, and response for enterprise systems that deliver computing as a service through a cloud computing environment. The authors mention that the traditional approaches for defining and evaluating quality, fall short in addressing the challenges faced by service providers and operations centers that struggle to monitor and manage quality of service. This is because they examine QoS from a limited perspective rather than from a SoS perspective applicable to a net-centric enterprise system in which any user from any location can share computing resources at any time. The authors have provided a concrete example for application of this new SoS approach to a real-world scenario (viz., distributed denial of service). Their simulated results confirm their approach. The work starts with mentioning that QoS specification and monitoring for cloud services is a complex and challenging issue as there are few universal benchmarks or standards. The usual quality metrics such as uptime and reliability may still be considered applicable in the context of cloud systems. Apart from these parameters, it is less clear what the QoS parameters unique to such systems are and how they should be applied in specific contexts. A lack of common metrics to be applied to cloud offerings from various providers is also a barrier to standardization of cloud offerings from different providers. The approach presented in this paper introduces a SoS to provide a clear and concise view of QoS events within cloud computing environments that proactively informs enterprise operators of the state of the enterprise and, thereby, enables timely operator response to QoS problems. The authors define various steps like 1) Define a SoS for monitoring, management, and response 2) Derive framework for QoS monitoring, management, and response in cloud computing environments 3) Identify Cloud computing metrics 4) Measuring performance metrics 5) Identify suitable locations within the cloud computing environment for metric detection and 6) Identify potential implementation schemes from which to collect and analyze the cloud computing QoS metrics. The authors also describe a real-world scenario in which the EMMRA CC (Enterprise

Monitoring, Management, and Response Architecture (EMMRA) for Cloud Computing Environments) SoS approach is applied to a complex cloud computing environment that is exposed to a cyber security threat (i.e., distributed denial of service - DDoS). In this scenario, the Cyber Security Plane has been used to observe cyber security threats across all domains in order to detect and enable proactive response to a DDoS security breach within any of these domains that could compromise the transactions and cause potentially devastating consequences to the end user. The authors built a prototype of an online transaction processing application where each layer of the application was hosted on a different cloud instance; the cloud itself was deployed in a private network which allowed them to demonstrate the dependence of performance metrics on the component-induced degradation. They also note that they could hardly feel the presence of a network-induced delay in the transactions because the setup was deployed in a private gigabit Ethernet supported network. Network-induced delay is a significant component of induced degradation in enterprise systems that have a global presence. They also notice that throughput is additive in nature. The new approach presented in this paper enables cloud computing service providers and operations centers to meet committed customer QoS levels using a trusted QoS metric collection and analysis implementation scheme that extends traditional monitoring, management, and response for IaaS and SaaS to a complete SOA stack that includes business logic (BaaS) and governance (GaaS). This paper was immensely beneficial to me in my work in identifying the performance parameters for the SOA based tool developed.

"The Case for Cloud Computing" in IEEE Computer Society 2009 published by Robert L. Grossman. The author distinguishes between clouds that provide on-demand computing instances and those that provide on-demand computing capacity. On-demand services and resources have been available over the Internet for some time, but the increased focus on cloud computing is due to three important differences: sale, simplicity and pricing. He mentions that a storage cloud provides storage services (block- or file-based); a data cloud provides data management services (record-, column-, or object-based); and a compute cloud provides computational services. The author also emphasizes that due to cloud computing, the “unit of computing” has moved from a single computer or rack of computers to a data center of computers. Not only has cloud computing scaled computing to the data center, but it has also introduced software, systems, and programming models that significantly reduce the complexity of accessing and using these resources.

This work is relevant to my study on cloud services and I could identify the various cloud services for the tool under development. I have also studied the cloud services available in context of the education sector on which an article[5] *was published. This article on this study includes cloud and its types and the layers involved in cloud architecture. It also looks into the scope of applications in education sector. There are also details of existing technologies used in the education sector like Microsoft Live@edu, Microsoft Office Web Apps and Google Apps. The work also gives a comparison of features provided by companies and its suitability for education. Lastly, the article gives the cloud computing architecture which mentions the facts on recent times which has seen the focus shifting from the application development to the application deployment. The various stakeholders (that we consider based on the role) participate in this system differently. This cloud also consists of the most important services ie SaaS, PaaS and IaaS that are used to automate the education sector.*

As seen in the Figure 2.1 Education Cloud Architecture*, the students, that constitute the bulk of stakeholders, and the lecturers are allowed to use the software services (SaaS) and at the same time, they have the facility for storage of data (IaaS).*

[5] Applicability of cloud computing in Acedemia in IJCSE. This paper has been cited in “Cloud Computing Support to University Business Processes in External Collaboration” by Imre Petkovics, Pere Tumbas, Predrag Matković, Zoltan Baracskai

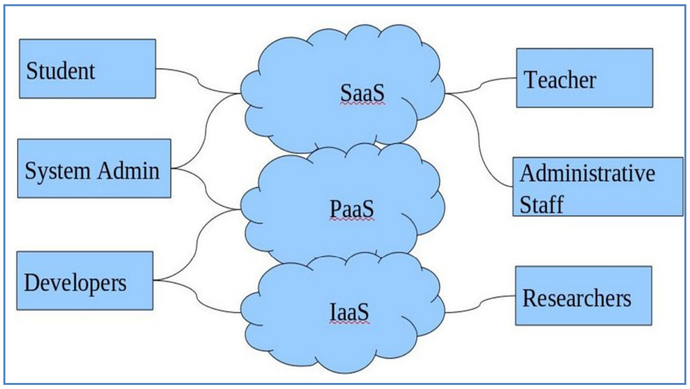

Figure 2.1 Education Cloud Architecture

This is evident from the Table 2.1 which shows a comparative of the most widely used applications. They are involved in the teaching learning process and hence their requirement on the cloud and the applications is limited to service usage and storage. Along with them, there are administrative staff who are involved in the maintaining of the documents and hence can be viewed as users of services and storage.

Another paper[6] about the need of cloud based System for higher & technical education in India as a scenario and proposing architecture for the implementation of cloud computing in educational system has been published. Finally, it concludes with the comparative study of various cloud based services available at different levels of cloud architecture for academia. The Indian Education sector has seen a tremendous rise in the field of higher education which has led to the demand for the automation of education sector at all the levels in order to cater to the need of information of various stakeholders. Due to burst in the field of communication technology everyone expects the access of relevant information (what they should); in fast, accurate and anytime any where manner. Information management of the educational sector including statutory bodies for the purpose of transparency and control through various information systems is the need and expectation of stakeholders.

[6] "Need of Cloud based System for Higher & Technical Education in India" is [23] of bibliography

Table 2.1 Comparative data of various cloud services

Sr. No	Company	Apps Name	Feature	Suitability
1	Google	Google Apps	Word Processing, Presentaion, ...	Office work, Assignments, Sharing data
2	Google	Google Drive	5GB Free	Storage and Syncing (Windows & Mac, Mobile)
3	Dropbox.com	Dropbox	2GB	Storage and Syncing (Window, Linux, Mac, Mobile)
4	Microsoft	SkyDrive	-	-
5	Microsoft	Office 360	-	Office work, Assignments, Sharing data
6	Adrive.com	aDrive	50GB	Storage
7	Box.net	Box	5GB	Storage
8	YouTube	Youtube	-	Video Hosting
9	Google	Picasa	-	Image Hosting
10	Yahoo	Fliker	-	Image Hosting
11	Microsoft	Microsoft Live@edu	-	Word Processing
	Microsoft	Microsoft Office Web Apps	-	Office work, Assignments, Sharing data

These stakeholders belong to different diversified background and have different perspectives and information needs for their participation. The technological development in the abstraction and encapsulation of the IT resources has been successfully implemented with the help of cloud architecture. This technology not only caters to the various stakeholders; it also ensures the sharing, availability, security and reliability of information involved. The cloud computing architecture gives so many options for the benefit of efficient use of data, memory and computing power so as to make green computing a reality. At the same time, I have also studied the issues related to cloud computing as explained below.

“Understanding Cloud Computing Vulnerabilities” by Bernd Grobauer, Tobias Walloschek and Elmar Stöcker Copublished by The Ieee Computer And Reliability Societies March/April 2011. The authors define indicators based on sound definitions of risk factors and cloud computing and its vulnerability. They refer to the Open Group's risk taxonomy, where vulnerability is the probability that an asset will be unable to resist the actions of a threat agent. Vulnerability exists when there is a difference between the force being applied by the threat agent, and an object's ability to resist that force. The authors have discussed further in reference to three examples of such vulnerabilities in context of cloud specific vulnerability. They are virtual machine escape, session riding and hijacking, and insecure or obsolete cryptography. They refer to NIST that describes five essential cloud characteristics: on-demand self-service, ubiquitous network access, resource pooling, rapid elasticity, and measured service. In relation to these characteristics, the authors have given examples of vulnerabilities with root causes in one or more of these characteristics. The examples are unauthorized access to management interface, internet protocol vulnerabilities, data recovery vulnerability and metering and billing evasion. They continue further that most services are likely web services, which share much vulnerability with web applications. Indeed, the web application layer might be realized completely by one or more web services such that the application URL would only give the user a browser component which shows that the supporting services and API functions share many vulnerabilities with the web applications layer. Such problems when known in advance can surely be handled and they also prove useful to people adopting or incorporating cloud computing.

"Expanding the Criteria for Evaluating Socio-Technical Software" in IEEE Transactions On Systems, Man, And Cybernetics—PART A: Systems And Humans, Vol. 38, NO. 4, JULY 2008 published by Brian Whitworth, Victor Bañuls, Cheickna Sylla, and Edward Mahinda. This paper compares two evaluation criterion frameworks for socio technical software. This work benefits my work in identifying parameters for performance evaluation of a tool. The authors mention about research on the technology acceptance model (TAM) which confirms that perceived usefulness and perceived ease of use are relevant criteria for users evaluating organizational software. However, information technology has changed considerably since TAM's 1989 inception, so an upgraded evaluation framework may apply. They say that the web of system performance (WOSP) model suggests eight evaluation criteria, based on a systems theory definition of performance. Their paper compares WOSP and TAM criterion frameworks in a performance evaluation experiment using the analytic hierarchy process method. The authors find that subjects who used both TAM and WOSP criteria preferred the WOSP criteria, were more satisfied with its decision outcomes, and found the WOSP evaluation more accurate and complete. They suggest that as socio technical software becomes more complex, users may need or prefer more comprehensive evaluation criterion frameworks. In the discussion on the subject related to current technology evaluation criteria, the technology evaluation suggests the three variable types: 1) System variables (useful, easy to use, secure), 2) User variables (Age, gender, experience, attitude) and 3) Organizational variables (Corporate goals, technology infrastructure, social structures, normative influences). The system performance goals given in this paper are security, flexibility, extendability, reliability, functionality, usability, connectivity and privacy for WOSP. These papers provided the necessary thought process for my work later on as discussed in chapter 5.

"On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software" by Muhammad Umair Ahmed Khan and Mohammed Zulkernine in 2009 33rd Annual IEEE International Computer Software and Applications Conference. The major contributions of this paper are a comparison of various secure software development life cycle (SSDLC) processes and secure software development (SSD) methods for requirements engineering which can be helpful for software developers in selecting an appropriate SSDLC process, security specification language, and security requirements engineering process according to their needs. The properties (like specification of SSD activities for the requirements engineering phase, design phase, implementation phase, security assurance phase, resources available to the developers etc) identified by the authors to compare various software security requirements specification languages can be used to choose a particular language depending on the application requirements. The authors also mention that the identified properties can guide software developers in designing a mechanism for translating one language into another which is useful as the tools developed for the target language can also be used to for the source language (after translation). They summarize their analysis on various SSDLC processes and SSD methods for requirements along with the related open issues. The authors also mention that to avoid security vulnerabilities, there are many secure software development efforts in the directions of secure software development life cycle processes, security specification languages, and security requirements engineering processes. They have compared and contrasted various secure software development processes based on a number of characteristics that such processes should have to analyze security specification languages with respect to desirable properties of such languages. Furthermore, they have also identified activities that should be performed in a security requirements engineering process to derive comprehensive security requirements.

"Performance and cost evaluation of an adaptive encryption architecture for cloud databases" published in IEEE Transactions on Cloud Computing by Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti. This article had been accepted for publication in the mentioned journal. It is about cloud database as a service which is a novel paradigm that can support several Internet-based applications, but its adoption requires the solution of information confidentiality problems. The authors propose a novel architecture for adaptive encryption of public cloud databases that offers an interesting alternative to the trade-off between the required data confidentiality level and the flexibility of the cloud database structures at design time. They have implemented the first proxy-free architecture for adaptive encryption of cloud databases. They also propose the first analytical cost estimation model for evaluating cloud database costs in plain and encrypted instances from a tenant's point of view in a medium-term period. It takes also into account the variability of cloud prices and the possibility that the database workload may change during the evaluation period. Their model is instanced with respect to several cloud provider offers and related real prices. This paper is concerned with database services and takes an opposite direction by evaluating the cloud service costs from a tenant's point of view. Their approach is quite original because the previous papers that they have referred are mainly about evaluating the pros and cons of porting scientific applications to a cloud platform, such as focusing on specific astronomy software and a specific cloud provider (Amazon), and presenting a compo-sable cost estimation model for some classes of scientific applications. The authors propose an architecture design which has been handled in three sub section ie adaptive encryption schemes, meta data structure and encrypted database management. The authors also mention the encryption algorithms that have been organized into structures called onions, where each onion is composed by an ordered set of encryption algorithms, called (encryption) layers. For the cost estimation of cloud database services the authors mention three main parameters: i.e. time, pricing and usage. The time identifies the time interval for which the tenant requires the service. The pricing refers to the prices of the cloud provider for subscription and resource usage. The usage denotes the total amount of resources used by the tenant. Their work also covers the cloud pricing models and usage estimation. They also evaluate the performance to verify whether the overheads of adaptive encryption represent an acceptable compromise from the performance point of view for guaranteeing data confidentiality in cloud database services. To this purpose, they have designed a suite of performance tests that allowed them to evaluate the impact of encryption and adaptive encryption on response time and throughput for different network latencies and for increasing numbers of concurrent clients. The authors note that there are two main tenant concerns that may prevent the adoption of the cloud as the fifth utility. They are data confidentiality and costs. The authors have addressed both issues in the case of cloud database services. The data confidentiality concerns have been addressed by proposing a novel cloud database architecture that uses adaptive encryption techniques with no intermediate servers. This scheme provides tenants with the best level of confidentiality for any database workload that is likely to change in a medium-term period. Moreover, they also propose a model and a methodology that allow a tenant to estimate the costs of plain and encrypted cloud database services even in the case of workload and cloud price variations in a mid-term horizon. By instantiating the model with actual cloud provider prices, they say that they can determine the encryption and adaptive encryption cost of data confidentiality.

During this phase, work was carried out on a few technical topics like xml, SOA, web services etc. along with an application oriented retail management topic which were published in a few journals. *A study of such interesting work helped me immensely in formulating my own ideas on current software technologies and SOA and also the applicability of cloud services in Indian Higher Technical Education system for which various papers have been published in journals. After a couple of years, I was able to define my focus of work and come up with the basic concept and idea of a desired software tool. The idea struck due to work carried out as part of job responsibility on a governing body website. I then weighed the possibility of implementing the idea in the situation and finally designed an architecture of a tool with a purpose and implemented it as a software tool. In context of this tool, I narrowed my focus on using open source technologies for the purpose of development and deployment. The technologies that I have used are listed below with their features.*

- **LAMP**

According to Linux Solutions, the acronym LAMP refers to a solution stack of software, usually free and open source software, used to run dynamic Web sites or servers. The original expansion is as follows:

- Linux, referring to the operating system;
- Apache, the Web server;
- My SQL, the database management system (or database server);
- P (one of several scripting languages: Perl, PHP or Python. *I have used PHP for the software development*)

The combination of these technologies is used primarily to define a web server infrastructure, define a programming paradigm of developing software, and establish a software distribution package. Though the originators of these open source programs did not design them all to work specifically with each other, the combination has become popular because of its low acquisition cost and because of the presence of its components which come bundled with most current Linux distributions. When used in combination they represent a solution stack of technologies that support application servers.

- Very low cost compared to other technologies
- GNU public licensing
- LAMP available as free software
- Best for Web applications
- LAMP has been proven faster, more flexible, and easier than any alternative

Choosing LAMP is the best way to gain complete control and power over the website. Each of the components in the LAMP stack is an example of Free or Open Source Software (FOSS). The benefit of the FOSS approach is three-fold. First, the nature of FOSS software means applications are available for free download, making them readily available to a wide range of people without payment. Second, licenses are open and thus have few restrictions on their use and the deployment of applications based on the FOSS technology. It is possible to develop and deploy LAMP-based projects without paying any license fees for distributing the software which makes it popular. Third, and a major reason for the growth and use of FOSS technology (including LAMP), is that because users have access to the source it is much easier to fix faults and improve the applications. In combination with the open license, this simplifies the development process for many enterprises and gives them flexibility and many other benefits as discussed below.

- **Customization**: As the LAMP components are open source, they have built up a huge array of additional components and modules that provide additional functionality. The open source approach enables any technical person to do the same, customizing components and functionality to suit the needs.

- **Ease of Development**: Powerful applications can be written using LAMP technology in relatively few lines of code which is straightforward enough that even non programmers can modify or extend the application.

- **Ease of Deployment**: There are neither licensing issues nor the need to compile applications, due to which deployment is often as easy as copying an application to a new host. Most hosting services provide LAMP-based environments as standard, or they can be deployed using a Linux distribution, such as Fedora or Debian.
- **Security**: With many developers developing the software and years of use by a wide range of users and community groups, LAMP technology is secure and stable where problems are normally fixed very quickly, and without the need for a costly support contract. Help and support during the development and deployment of LAMP-based applications is available.

PHP ie PHP Hypertext Preprocessor can generate dynamic page content; create, open, read, write, delete, and close files on the server; collect form data; send and receive cookies; add, delete, modify data in database; restrict users to access some pages on the website and encrypt data. With PHP, a person is not limited to output HTML. The output can be images, PDF files, and even Flash movies along with any text, such as XHTML and XML. It also has benefits as given below.

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is easy to learn and runs efficiently on the server side

- **JSON**

During my study on SOA, I came across XML technology for the purpose of implementing independent and complete services. Along with XML, I also read about JSON. There are many differences between JSON and XML as they were designed for different purpose. Once compressed, both formats take about the same space.

The most important aspect of JSON was that it had simple syntax, which results in less "markup" overhead compared to XML and it is easy to use with JavaScript as the markup is a subset of JS object literal notation and has the same basic data types as JavaScript but it supports only a handful of different data types. In short, XML is about the document and JSON is about data. While both JSON and XML are used to represent data in hierarchy or tree structure, XML is far more descriptive and verbose. The serialized syntax of JSON has less redundancy than the syntax of XML. In case of XML, every element in the tree has a name (the element type name), and the element must be enclosed in a matching pair of tags ie XML has data and meta data, both. By contrast, JSON expresses trees in a nested array type of notation similar to that of Javascript. In fact, a JSON document can be parsed as Javascript to result in the corresponding array. XML may be more natural and useful for large, structured documents, while JSON is often the simpler and less verbose format for transmitting data sets. The JSON object is built in two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

In its simplest form, a JSON object is just a comma delimited set of name/value pairs. For example:

```
{"name one":"value one","name two":"value two"}
```

The data model of XML and JSON is different. Apart from the lack of markup, JSON also doesn't have all the other XML features, such as Processing instructions, Comments and Attributes. Hence, not only the data itself is more lightweight, but also any library implementing the data model. This makes rendering, parsing, processing JSON with non-JavaScript languages a lot faster. Also JSON is actual JavaScript code, and thus much more lightweight for a JavaScript parser to process. For the purpose of data-interchange, XML is not well suited. It carries a lot of baggage, and it doesn't match the data model of most programming languages. Though XML is difficult to do away with as it is already in use, newer technologies must surely be considered as per the situation. Comparing XML and JSON on various factors given below, appropriate selection between the two can be done.

- **Simplicity**

XML is simpler than SGML, but JSON is much simpler than XML. JSON has a much smaller grammar and maps more directly onto the data structures used in modern programming languages. The JSON data model is simpler; it has fewer different kinds of object and they have fewer properties.

- **Extensibility**

JSON is not extensible because it does not need to be. JSON is not a document markup language, so it is not necessary to define new tags or attributes to represent data in it.

- **Interoperability**

JSON has the same interoperability potential as XML. The JSON data model is closer to the data models available in common programming languages such as Javascript, so less data conversion is needed when using these languages.

Openness

JSON is at least as open as XML, perhaps more so because it is not in the center of corporate / political standardization struggles.

- **JQUERY**

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on the website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that can be called with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. The jQuery library contains features like HTML/DOM manipulation, CSS manipulation, HTML event methods, Effects and animations, AJAX and Utilities. jQuery is used because of its advantages given below.

- **Lightweight –** jQuery is a lightweight library and because of its lightweight characteristics it provides high performance
- **Cross browser support –** jQuery supports cross browser, means while working with JavaScript the most important factor we have to consider is browser whereas while working with jQuery there is no need to worry about browser because jQuery functions are completely browser independent
- **Easy Dom traversing –** With the help of jQuery, the parent or child of any HTML element can be found at any time without writing much code.
- **Easy Plugins –** jQuery is highly extensible. It allows the extension of the existing behavior or add new behaviors.