Evaluation of a Framework for Integrated Web Services

Savita Gautam[1], Mr. Ram Krishna Paramhansh Dubey[2]

[1]M.Tech, Computer Science and Engineering, SR Institute of Management & Technology, Lucknow, India
[2]Professor, Computer Science and Engineering, SR Institute of Management & Technology, Lucknow

---***---

Abstract - *One of the most popular solutions is to use specialized software, particularly in light of the increasing need for software in the industrial sector to handle model simulation and physical prediction. This is particularly accurate given the rising desire for employing software to address these problems. This specific component is only one of the numerous reasons why using professional software is one of the most popular choices. One of the most popular options is the usage of professional software since it is an approach that is both effective and successful, which is what most professionals want when choosing a technique to achieve their objectives. This is one of the factors making the usage of specialized software one of the most often used alternatives. Users appreciate a simple process for obtaining on-demand software services, especially one that relieves them of the burden of installing and updating the application themselves. Additionally, users anticipate that the software services they use on demand would be affordable. This is essential. In this thesis, I provide an argument in support of a web-based architecture that enables users to access a broad range of software applications and buy the software services they need via online markets. Users may be able to incorporate a large range of software programs for themselves using this design. I'll first provide the evidence in support of this architectural style before making my case. Additionally, I'll go over the advantages of using a design that emphasizes the quality of the services offered to clients. We poll users to assess the quality of the user experience offered by the platform, and we utilise two common "Simulation Software as a Service" implementations to demonstrate how this integrated framework may be built.*

Key Words: Web, frame, Integration Concept in web, web coding by integration concept.

1. INTRODUCTION

Consumers who use automated programs find it easier to manage the numerous financial documentation they come across on a daily basis. They use the internet to communicate photos, movies, and articles to their friends and family. On the other side, business software may help firms manage their personal data, create financial projections, and simulate a model or process. The usage of web services is becoming a more widespread practice in today's state-of-the-art software systems [1]. Numerous internet services have been made available thanks to the efforts of Amazon, Google, and Microsoft. Online services are also available for a broad range of business applications, such as Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) (CRM). Users may connect to the web-based client interface of the Web service, which is intended to be versatile. Additionally, several users of the same online service may reduce costs for all parties, resulting in cost savings. The majority of professionals prefer to choose a strategy that is both effective and efficient for achieving their goals, which is to use specialized software applications. This is due to the rise in demands for the use of software to deal with model simulation and physical predictor in the industrial sector. [5] "The expensive expense of developing and maintaining software and hardware infrastructures for the purpose of providing services to enterprises has led to a noteworthy shift towards the utilization of third-party services which supply processing power, and data storage space to customers." "The high cost of creating and maintaining software and hardware infrastructures for the purpose of delivering services to businesses has led to a noticeable trend towards the use of third-party SaaS, or software-as-a-service, is a software delivery model in which users can access software and associated data by using a thin client and a web browser. This model provides a number of significant benefits, including lower costs, remote access, rapid deployment, and

1.1. Architecture of Service-Oriented

One kind of SOA architecture is one in which several distinct services interact with one another in a way that only tangentially connects them. A service is a package of tasks that a service provider offers to a customer in order to meet their needs. This term is used to service-oriented architecture, sometimes referred to as SOA. The environment and the present state of other services do not significantly affect a service's operation while it is operating inside a SOA. The commitment of an execution of a service is governed by the resident service rather than the global procedure. This is because a service may function independently and is not reliant on any one particular platform. Because of this, processes made out of web services often do not execute in a fashion that is consistent with the idea of serializability. In a situation like this, concurrent processes may use different services to access or alter the same data; but, the isolation characteristic that typically exists between the executions of different services won't be present.

1.2. Data Dependency Analysis

Xiao presents both a formal definition and a comprehensive description of a process dependence model (2006). At both the operation and process levels, this architecture specifies read and write dependencies. Dependencies for reading and writing are listed. Because of the way SOA was built, if one process fails, it might potentially call into question the reliability of every other process that depends on it. This is due to the way SOA was designed, where each process is dependant upon the one before it. Using a paradigm known as the process dependency model, it is feasible to create a collection of processes that are dependent on the one that failed. Because the model allows for the construction of dependent groups, this is possible. Executing the proper recovery procedures for the processes that are a part of the dependent set is absolutely necessary as a consequence of this. Effective techniques to assess if recovery is necessary and to dynamically carry out recovery activities, however, continue to provide a difficult topic for research and development to focus on at this moment. This is due to the fact that these methods are still challenging to develop. Process interference rules (PIRs) were proposed by Xiao and Urban (2007) to assess user-defined circumstances that determine whether the dependent process should carry on operating or initiate its own recovery procedures.

1.3. Web Service Standards

A framework called WS-Coordination may be expanded to give protocols for coordinating the actions of distributed applications. F. The WS-Coordination document's authors are Cabrera, Copeland, and Freund. These kinds of coordination protocols are used to support a variety of applications, including those that require reaching constant consensus on the outcomes of actions taken in several different locations. The structure of a coordination service, also known as a coordinator, is described in the WSCoordination standard. It is made up of the previously described component services. The first component is an activation service equipped with an action that makes it possible for an application to produce a coordinated instance or context. This service's provision is a part of the first component. A registration service that allows applications to register for coordination protocols is the second component.

1.4. Recent Data Consistency Approaches

Mikalsen, Tai, and Rouvellou (2002) developed the ground-breaking Online Service Transaction (WSTx) architecture to address the problem of transactional dependability in online service development. The transactional attitudes framework is the name of this conceptual framework. In the WSTx architecture, transactional attitudes are what allow web service providers to declare their unique transactional capabilities and semantics. They are also what allow web service users to express their unique transactional requirements. Transactional attitudes are used to indicate both these capacities and requirements. It is possible to distinguish between two different kinds of attitudes within the WSTx paradigm: provider transactional attitudes (PTAs) and client transactional attitudes (CTAs) (CTAs). Online service providers use PTAs to precisely describe the transactional behavior of their services, while customers utilize CTAs to explicitly state their expectations and the acceptance criteria for the results of their transactions. Online service companies use PTAs in order to precisely describe the transactional behaviour of their services. Each client will engage in one or more activities inside the confines of a web transaction, and each of these actions will signify a provider transaction that will take place within the parameters of the more comprehensive web transaction.

1.5. Web Service Models in a Service-Oriented Architecture

People may include several various web-based services into a single web application by employing a range of application programming interfaces, sometimes referred to as APIs for short. Application programming interface, or API, is a shorthand for that. API is an abbreviation for application programming interface. API is a contraction. "Web Services offer a new paradigm in present-day collaborative application development in companies" and "Web Services transcend systems, networks, and organizations" are two expressions that may be used to explain the influence of Web Services. [2] The example that follows shows a business whose only goal is to meet the needs and expectations of the clients it works with.

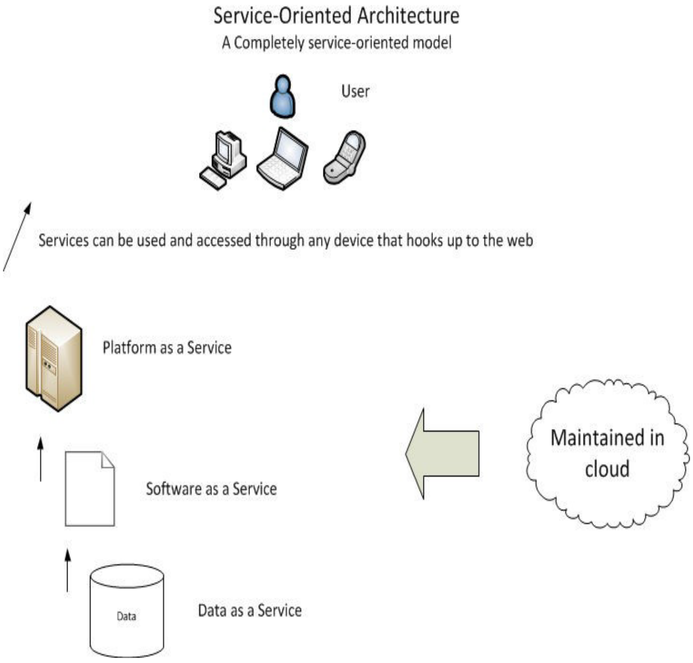

Service-Oriented Architecture
A Completely service-oriented model
User
Services can be used and accessed through any device that hooks up to the web
Platform as a Service
Maintained in cloud
Software as a Service
Data
Data as a Service

Figure-1: Web Service Models in a Service-Oriented Architecture.

2. COLLABORATIVE WEB SERVICE

Numerous research support the concept of web service cooperation, which gives it credibility. Zhengxiong Hou and his colleagues [10] present a web-based architecture for the delivery of software as a service that is built on high-performance computing resources. It creates a web gateway that customers may visit in order to access the application software while it is operating in a pay-for-use mode. This has the immediate effect of removing the difficulties inherent in conventional software, which is not service-oriented and is constrained by software licensing. This E-Commerce feature is quite identical to my integrated framework, which allows consumers to choose the sort of licensable software service that best suits their needs based on the cost of that licensable choice. A new paradigm for collaborative application development that is based on web services and provides a wide range of diverse examples to follow has been created by Irfan Awan and his colleagues [3]. On the other hand, they provided a three-layered architecture for collaborative applications that handles a variety of treatments.

2.1. Web Service Configuration and Customization

The authors Haiqi Liang et al. [1] provide a procedure for updating online services that include cooperation between the consumer and the service provider in a predefined manner to carry out the update. The technique of personalising online services via the use of collaborative programming is the word used to refer to this strategy. WS-Policy. The vocabulary supplied by the WS-Policy [38] standard is both flexible and expandable inside an XML web services architecture. As a result, it is much easier to articulate the talents, requirements, and fundamental characteristics of things.

2.2. Portal and Catalog Layer.

Before application software could be made available via the use of services, the foundational components absolutely needed to function properly had to be in place. These key components will be broken down into more detail later on in the piece. These components include of service encapsulation, service definition, and deployment for application software; service registration and discovery; service negotiation and selection; service delivery and consumption; service composition; and on-demand licencing service and accounting [10]. The picture that can be looked at to view the component architecture that can be looked at can be seen displayed in the following image, which can be seen presented below, and it can also be seen below. The picture can be seen displayed in the following image, which can be seen presented below, and it can also be seen below.

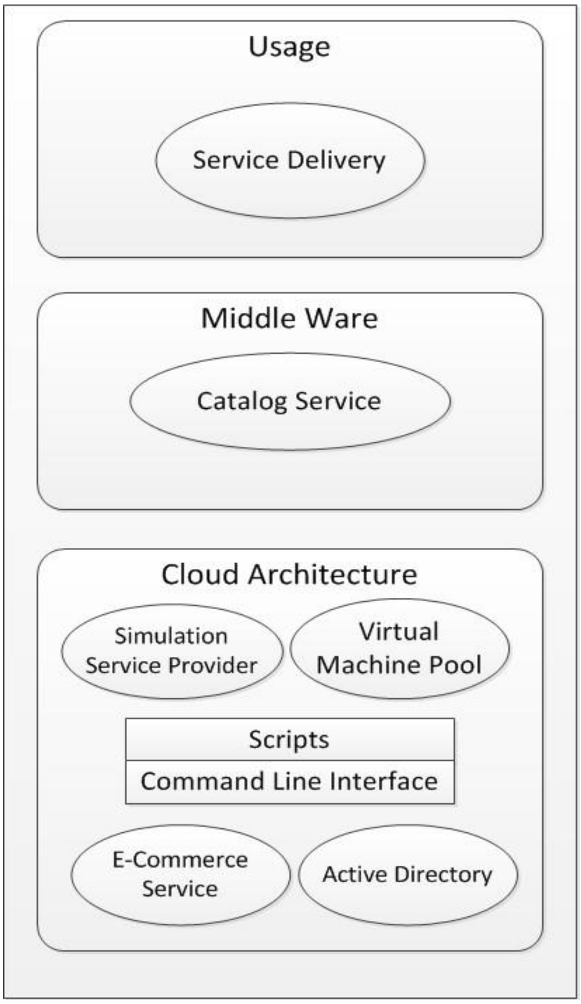


Figure-2: Portal and Catalog Layer.

2.3. Manifold Flow Predictor

Research conducted by P&G on fluid flow has been more productive thanks, in part, to the use of a computer system within the corporation known as the Manifold Flow Predictor site. The programme in question was created in collaboration with a different company that goes by the name TotalSim [52], and the name of the application is formed from the names of the two individuals who were initially responsible for its creation. P&G has found that using this method has shown to be useful for their job when it comes to achieving their goals (MFP). When creating this website, we made use of both the high-performance computing capabilities offered by OSC and an open-source software instrument called OpenFoam [44]. Both of these resources were quite helpful. My suggestion that they take on the responsibility of constructing the gateway by utilising Drupal and presenting the results of the simulation during a session of Virtual Network Computing was accepted by them, and they said that they would do so. In addition, my suggestion that they present the results of the simulation during a session of Virtual Network Computing.

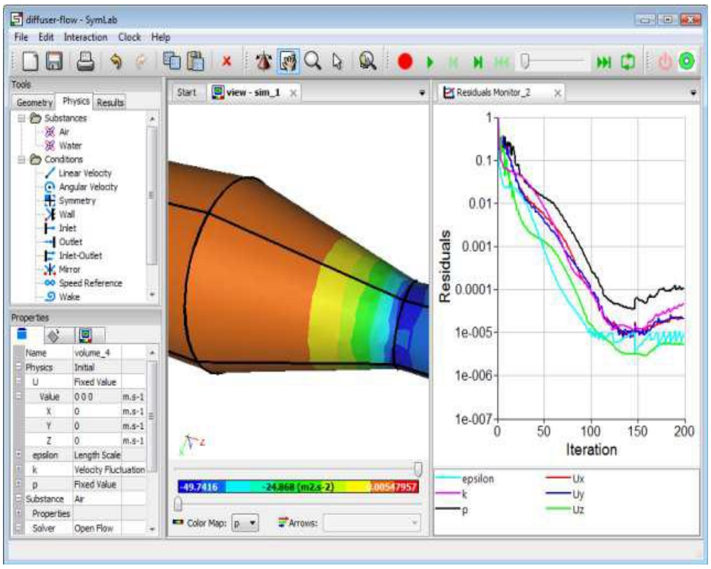


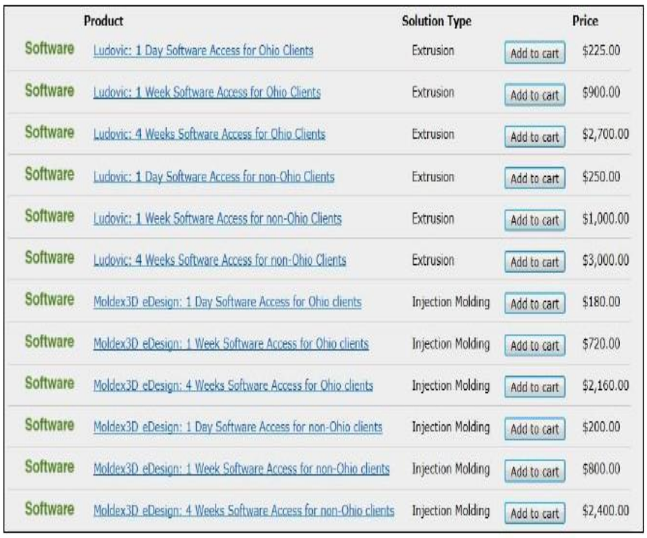
Figure-3: Example of an OpenFoam simulation result.

3. ANALYSIS PORTION

We outline how we performed the user surveys that we did in order to assess the true advantages that the integrated framework offers, and we provide some of the data that we obtained from those surveys of users. Both the Polymer Portal and the Manifold Flow Predictor projects that we were working on benefited from our usage of the SMaaS architecture, which was advantageous to both of those projects. The surveys are used as examples throughout the whole of this chapter, and all of the information was gathered via the use of questionnaires in addition to interviews. The goal of the questions is to assess the practicability, with a main emphasis on user experience qualities such as real cost, degree of happiness, and simplicity as some of the most important factors to take into consideration.

3.1. Cost

Our Polymer Portal was developed on an SMaaS architecture; it operates in a "pay-for-use" mode, which allows customers to acquire a service licence for a shorter length of time at a reduced price. One of the people who utilizes Polymer Portal said that he constantly uses Moldex3D to imitate plastic injection. Despite this, the price of the licence for Moldex3D is rather costly. It costs approximately $50,000 for one year, and that does not include a software update or technical support, both of which cost an additional $20,000 each. Additionally, payments and maintenance for the underlying hardware are still required. His firm is a mid-sized polymer company, and as such, they do not want to spend a significant amount of money on costs that are incurred immediately. Polymer Portal, on the other hand, fixes all of these problems. The user has the ability to use the simulation service inside a high-performance virtual machine that is setup and managed by the Ohio Supercomputer Center by purchasing a one-day Moldex3D licence for just $180. The licence gives the user access to the simulation service. Another user of the Polymer Portal said that he got a lot out of the training course that was offered. He is interested in analysing the extrusion process of a certain material, but he is unable to do so since the programme that he needs to utilise, Ludovic [40], is difficult to use.



	Product	Solution Type		Price
Software	Ludovic: 1 Day Software Access for Ohio Clients	Extrusion	Add to cart	$225.00
Software	Ludovic: 1 Week Software Access for Ohio Clients	Extrusion	Add to cart	$900.00
Software	Ludovic: 4 Weeks Software Access for Ohio Clients	Extrusion	Add to cart	$2,700.00
Software	Ludovic: 1 Day Software Access for non-Ohio Clients	Extrusion	Add to cart	$250.00
Software	Ludovic: 1 Week Software Access for non-Ohio Clients	Extrusion	Add to cart	$1,000.00
Software	Ludovic: 4 Weeks Software Access for non-Ohio Clients	Extrusion	Add to cart	$3,000.00
Software	Moldex3D eDesign: 1 Day Software Access for Ohio clients	Injection Molding	Add to cart	$180.00
Software	Moldex3D eDesign: 1 Week Software Access for Ohio clients	Injection Molding	Add to cart	$720.00
Software	Moldex3D eDesign: 4 Weeks Software Access for Ohio clients	Injection Molding	Add to cart	$2,160.00
Software	Moldex3D eDesign: 1 Day Software Access for non-Ohio clients	Injection Molding	Add to cart	$200.00
Software	Moldex3D eDesign: 1 Week Software Access for non-Ohio clients	Injection Molding	Add to cart	$800.00
Software	Moldex3D eDesign: 4 Weeks Software Access for non-Ohio clients	Injection Molding	Add to cart	$2,400.00

Figure-4: Price of Simulation Services on Polymer Portal.

The illustration demonstrates that there are many different types of licenses available to users to choose from, and users are free to choose a license based on their particular requirements. Customers choose one-day licences the most, according to the studies, since not only are they inexpensive, but they also provide users a greater degree of freedom.

4. CONCLUSIONS

The purpose of this thesis is to offer a framework that saves money, streamlines procedures, and improves the quality of the user experience by combining numerous online components and software services into a single website. In order to accomplish this, this thesis will combine numerous online components and software services. The most important elements of the framework of the argument are included in the following phrase: First, I will outline the architecture, and then I will proceed to discuss the key components that compose this framework. In addition, we apply this architecture in the construction of two other products, namely Polymer Portal and Manifold Flow Predictor. These products are named after their respective names. These are the only two goods in the world that are exactly like each other. Customers have the option of using Simulation Software as a Service (SaaS), which is offered by both of these systems. Software as a Service is an acronym for "Software as a Service," and user surveys are an additional instrument that we use in the process of evaluating the two products in terms of their usability and the overall user experience. This comparison is being carried out so that it can be determined which of the two goods is the most advantageous option. We have shown that customers benefit from this design in a number of ways, including cheaper prices, more flexibility of software services, enhanced user experiences, and enhanced software performance.

REFERENCES

[1] Haiqi Liang, Wei Sun, Xin Zhang, Zhongbo Jiang "A Policy Framework for Collaborative Web Service Customization" SOSE'06, 2006

[2] N. Dhyanesh, G.C. Vineel, S.V. Raghavan "DEVISE: A Methodology for Building Web Services Based Infrastructure for Collaborative Enterprises" WETICE'03, 2003

[3] Irfan Awan, Muhammad Younas "Differential Treatment of Web Services-based Collaborative Applications" The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings

[4] James D. Herbsleb "Global Software Engineering: The Future of Socio-technical Coordination" FOSE'07, 2007

[5] K. Selcuk, Wen-Syan Li, Thomas Phan, Minqi Zhou "Frontiers in Information and Software as Services" ICDE 2009

[6] Jyoti Namjoshi, Archana Gupte, "Service Oriented Architecture for Cloud based Travel Reservation Software as a Service" Cloud 2009

[7] Vidyanand Choudhary, "Software as a Service: Implications for Investment in Software Development" HICSS'07, 2007

[8] Song Guo, Fan Bai, Xiaolin Xu, "Simulation Software as a Service and Service- Oriented Simulation Experiment" IRI 2011

[9] Espadas Javier, Concha David, Molina Arturo "Application Development over Software-as-a-Service platforms" ICSEA 2008

[10] Zhengxiong Hou, etc "ASAAS: Application Software as a Service for High Performance Could Computing" HPCC 2006

[11] Wei Sun, Xin Zhang, etc "Software as a Service: Configuration and Customization Perspective" SERVICES-2 2008

[12] H. Karhunen, M. Jantti, A. Earola "Serivce-Oriented Software Engineering (SOSE) Framework" IEEE 2005

[13] G. Ma, P. Lu "PBSWEB: A Web-based Interface to the Portable Batch System"

[14] E. Knorr, "Software as a Service: The Next Big Thing",

http://www.infoworld.com/d/applications/software-service-next-big-thing-319

[15] Ubercart Module: Ubercart, http://drupal.org/project/ubercart

[16] LDAP Module: LDAP, http://drupal.org/project/ldap

[17] E-Weld Predictor, https://eweldpredictor.ewi.org

[18] ITaP, http://www.itap.purdue.edu/

[19] T. Richardson, etc "Virtual Network Computing", Internet Computing, IEEE 1998

[20] M. Younas, K-M Chao, C. Laing "SUDA: Service Oriented Design Activities"

18" International Conference on Computer Supported Cooperative Work in Design (CSCWD 2004), IEEE CS Press, May 2004, Xiamen, China, pp. 419-424

[21] P. Dewan, A. Shma "An experiment in Interoperating Heterogeneous Collaborative Systems" 6th European Conference on Computer Supported Cooperative Work - ECSCW'99, 1999, Copenhagen, Denmark

[22] F. Liu, etc "SaaS Integration for Software Cloud", CLOUD 2010

[23] T. Richardson, etc "Virtual Network Computing" Internet Computing, 1998

[24] M. D. d. Assunçao, A. d. Costanzo, R. Buyya. "Evaluating the Cost- Benefit of Using Cloud Computing to Extend the Capacity of Clusters." HPDC'09, 2009

[25] Wei-Tek Tsai, Wu Li, Hessam Sarjoughian "P4-SIMSAAS: Policy Specification For Multi-Tendency Simulation Software-As-A-Service Model", Qihong Shao,April 2011, ANSS '11: Proceedings of the 44th Annual Simulation Symposium

[26] Kadlec, J.; Jaros, D.; Kuchta, R.; "Implementation of an Advanced Authentication Method within Microsoft Active Directory Network Services" ICWMC, 2010 6th

[27] Ashish G. Revar, Madhuri D. Bhavsr "Securing User authentication using Single Sign-On in Cloud Computing"