

Software Product line Technologies for Web-Based Application

Revati Gandhi

Department of Computer Science,
Illinois Institute of Technology,
Chicago, US

Sukhada Pande

Department of Computer Science,
Illinois Institute of Technology,
Chicago, US

Mayuree Nandi

Department of Computer Science,
Illinois Institute of Technology,
Chicago, US

Abstract— In the emerging technological world, the amount of software products that are being developed are huge. With every software that is getting developed, a new form of Product Line is emerging. It is certainly necessary to have some standard form Product Line to manage and develop these products faster and more efficiently. In our paper we are going to look into Software product line technologies for web based systems. We will review the basic concepts of software Product line, discuss its benefits and their drawbacks. We will also be primarily understanding the web based technology to understand which product line suits its best.

Keywords - Software technology; Software Product line; Web based system

I INTRODUCTION

Software product lines have emerged as a new software development paradigm of great importance. 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.

The Software Engineering Institute (SEI) defines the software Product Line as [1] “A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying 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.”

It has emerged as an important development paradigm allowing big companies to make improvements in the time to market, cost, productivity, quality. It provides a great capability for mass customization.



Fig : The three essential activities involved in an SPL (source: http://www.sei.cmu.edu/plp/frame_report/PL.essential.act.htm) [5]

The software product line puts constraints on the way in which the systems in a software product line are developed.

Software Product lines are the strategic reuse of the information you already have. The reuse in software product line is that it provides the business a dimension, avoids iterations, lets us focus on architecture of the product. As it is strategic reuse it lets us have preplanning and also let the product and process have a connection.

With a lot of benefits the Software product line have limitations. Which are as follows:

- It cannot be used for small-grained reuse.
- Not suitable for a single system.
- Cannot be used for component-based products.
- Or can not be applied to a single product.

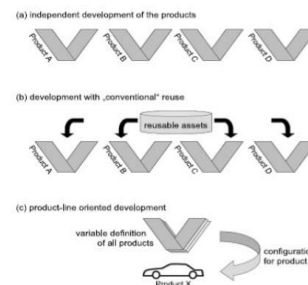


Fig: Difference between Software Product Line approach and conventional reuse[3]

Software product line engineering allows the shifting of the focus of development and evolution from the individual products to the entire product line. Indeed, the product line becomes a first-order entity of development. The advantage of this approach is that the relations between products, especially their commonalities and variabilities, become concrete entities for development and evolution. This benefit is important for strategic product line scope, adaptation of market needs and reuse of development artifacts.[3]

The World Wide Web has succeeded in large part because its software architecture has been designed to meet the needs of an Internet-scale distributed hypermedia system.

A huge amount of software applications are developed and invented every now and then for web based applications.

Thus requiring a well-defined product line for sustainable and consistent use of it.

A web-based application is any program that is accessed over the internet which is indeed using HTTP, rather than the local device .

The focus of our study is to understand the web-based application technologies and their software product line.

II. RELATED WORK

In our paper we are studying software product lines for web-based technologies, so we have done a study on what actually the software product line is and the product line in web-based technologies.

There is a lot of study on software product lines, how they can be implemented and used.

Ouali et al. studies the software product line studies its variability concepts. They compare the traditional software development methods and software product line methods ,review both of their

drawbacks and also how different SPL frameworks are.

F.Coallier et al. paper proposes Product Line engineering practices model that are built on existing models i.e SEI Product Line Practice Framework and Dikel et al.'s VRAPS model. They proposed their model in the form of 31 Product Line practice areas, grouped in five categories.

Rupali et al. paper proposes that Aspect Oriented Programming makes it possible for developers to write a code which is modular which is able to address many existing concerns like synchronization, security,performance monitoring etc.which will increase the software quality.

A. Fatolahi et al.[11] presents a good paper on model-driven approach describing how MDD works and how UML profiling works.

M. Voelter et al.[12] presents a product line implementation of Aspect-oriented and Model-driven technology, giving a brief study how AOP and MSD can be used.

III. CONCEPTS

Our paper's primary goal is to study software product lines for web-application to dive into it. We first need to understand how a web-application is structured.

As we have seen, any application that communicates via the HTTP is categorised as web-application.

All applications like Flash games, online calculators, calendars , etc. and some more intensive applications such as web-based word processors and spreadsheet applications.

These applications usually have a client and a server.

Client is the receiving end of a service or the one which request for service to the server.

A client can connect to a server through different means like domain sockets , named shared memory , through internet protocols etc. A client can be of below three types:

- Thin client
- Thick/Fat Client
- Hybrid client.

Server : A server is anything that provides services , or manages the resources .They are dedicated computers having high computing power . And serve a particular service to the client. Client sends the request to the server i.e HTTP request for the service it wants to access.Usually the client is like the web-browser we use to send the request to the server.

The request is sent to the web-server, which usually has a back-end application developed using programming languages like java, PHP, python , .net etc.

The web server processes the request using business logic . If there are any database updates , we have a database server , which then saves the information and service request is completed.

Below is the typical architecture of web-based technology.

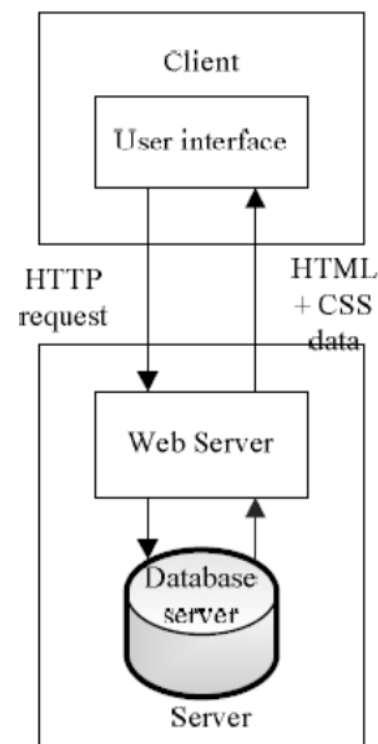


Fig : Web-based Application [6]

Then this request is sent back to the client through HTTP response.

All web applications use a similar approach, thus designing a software product line for web-based technology is efficient as maximum of them use the same approach and thus reusability can be exploited to the fullest.

In our paper we will be analysing 3 main web-based application product line technologies that are as follows:

- Object Oriented Technology
- Aspect Oriented Technology
- Model Driven Technology.

A . Object Oriented Technology.

Object Oriented technology or Object Oriented Programming (OOP) is basically based on the concept of “Objects”. It categories any real word aspect as Object and builds the applications depending on its characteristics and behaviour.

These objects may contain data in the form of fields often known as attributes and code in the form of procedure often known as methods. These different objects interact with each other and modify the characteristics of the objects.

Object Oriented technology is becoming popular. It uses the basic character of the software product line that is reusability. A large number of Object Oriented programs are extensively reusable and concluding that they are hugely independent in nature.

It has the main below features :

1. Objects: The objects have characteristics:
 - a. State which records the history of the object .
 - b. Behaviour : Effects on its state and its relation with the other objects.
 - c. Identity : Either by name or reference.
 - d. Relations: Interaction with other Objects.
2. Polymorphism: One name and different form of an object.
3. Inheritance : Object can get traits from other objects , this is the main feature that enables reusability.
4. Encapsulation: Where the actual behaviour of object is hidden from the user.

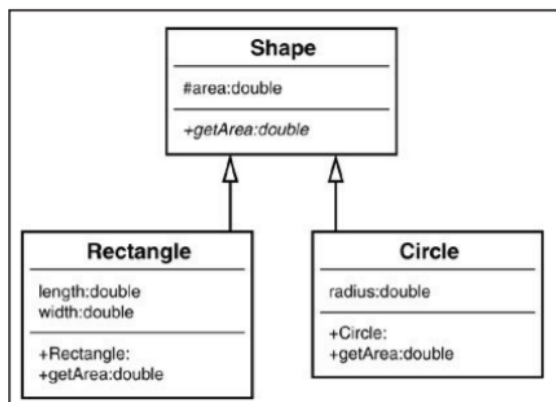


Fig : Object Oriented Approach[7]

There are various programming languages that use the Object oriented approach Like C++ ,Java, SmallTalk, Python etc.

Advantage of Object-Oriented Technology are as follows:

- Reduces maintainability
- Real world Modelling
- Improve flexibility and reliability.

Disadvantage of Object-Oriented technology are as follows:

- Require more memory space.
- More susceptible to security breaches.
- Speed of Object Oriented applications is slower than the others.

Case Study : Java

Java is a widely used back-end programming language that provides a large spectrum of frameworks and libraries to support almost all web-based features.

It is a portable language that can run on any web-enabled computer via that computer's Web browser. As such, it offers the standard Internet and Intranet programming language.

It has various features like Platform Independent i.e Once a code is compiled to byte code, that particular code can be run on any other system.

Multithreading that enables the program to run in parallel process. That the multiple threads can do various processes simultaneously. Garbage Collection which is one of the main features of java that clears up the memory once the application is over , thus making the program memory sufficient.

B. Aspect-Oriented Technology

A moderately modern programming prototype, that earned the software community's attention is Aspect - Oriented Programming. The prototype is as of now assessed for conventional OOP and component-based computer program improvement with remarkable results.

Based on aspectual decomposition, it is applied to catch the crosscutting matters. Cross-cutting uses extra-functional requirements which get repeated and then propagated over the whole structure.

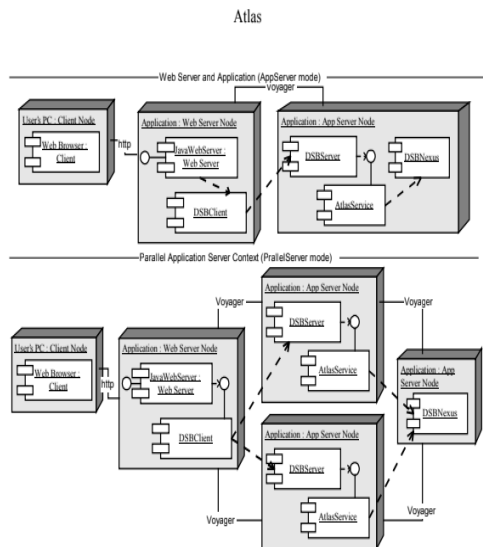


Fig : Network Components[9]

AOP generally examines crosscutting concerns throughout the structure, and makes implementation of it much easier and flexible. Major goals of AOP involves :

- making implementation easier, and
- easier debugging with lesser chances of error formulation.

AOP makes cross-cutting concerns quite simpler, having the facility of reusing the code of the implementations that applies cross-cutting over the entire system.

The architectural design gets created by recognizing, establishing , assessing all forms separately at the architectural design level. As described above the aspect oriented programming is based on aspectual architecture. The considered final design model should describe concerns and their connectivities separately, and this is done as the process part of the modelling approach.

The other feature is to describe the constructs that can define all elements to be represented in the design and the relationships that can exist between those elements. They are specifically provided to support concern modularization and the specification of concern composition, with consideration for conflicts. Here the special feature is that the model unlike others can be modularized.

Even aspect-oriented approach is concerned about the support for application and language developers.

While designing support for the application developers aspect-oriented approach works on how issues are implemented and how they are formed independently. Few aspects for support for language developers are constructing languages or DSLs , and many more.

Any aspect oriented approach can be used to specify conditions to create multiple aspects , spot and resolute varied possible differences of aspects, and describe and examine aspects to an already well formulated structure.

Case Study: Atlas

The Advanced Teaching and Learning Academic Server (Atlas) is a really helpful web-based platform used mainly for learning:

Students sign up for courses and navigate to learn through a useful manner.

Aspect-Oriented Programming for Atlas is achieved by combining it with AspectJ , it provides the aspect-oriented approach support for Java. The aspect-oriented approach is used when it uses different aspects for the proper implementation of Atlas.

One aspect being : it was desired that a student be able to select a varied environment each time they want. One way to do this would have been adding a layer, but rather we used aspect to the existing Build Pattern. When the layering approach of Design Pattern and Build Pattern was being thought, it seemed the aspect approach was much simpler to implement. As a result several tiered configuration in Atlas was achievable using aspect , and it can be run on application server or as an applet.

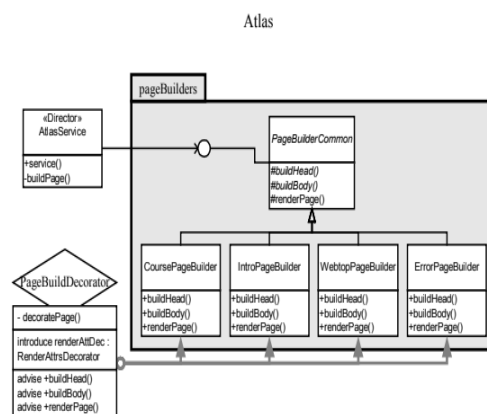


Figure : Build Pattern with an Aspect[9]

The other aspect being : when Atlas works as a distributed system it makes the situation for debugging more difficult. When we use the aspect- oriented approach we design the system into aspects so that we can add or remove quite easily the errors and debug them as the approach facilitates the separation of errors.

From the case study we learn that the aspect oriented approach helps us to streamline any object’s behaviour in the whole system and we also learn the importance of the relationship between different objects.

C.Model Driven Technology

Model driven technology is software technology that focuses on creating and exploiting domain models.

It relies on the definition of mapping that transforms high-level models to models of web platforms.

Model Driven approach improves the way software is developed by capturing their main features of system in models which are developed. During the systems life cycle , models are synchronized and transformed between different levels of abstraction and different viewpoints.

The two core concepts of Model driven approach are abstraction and automation.

Abstraction means to organise complex systems , and once this is done the automation uses Domain specific Languages (DSL) like HTML , ColdFusion which can integrate other programming languages.

A DSL is basically for building models .It consists of a meta model as well as definition of concrete syntax that can be used to design models.These syntax can be textual, graphic or tables , dialogs , trees etc.

Model Driven Technology has a defined architecture which is standardized by OMG . It can be seen as a framework of three layers of modeling .

They are as below :

- Computation-Independent Model
- Platform -Independent Model
- Platform-Specific Model

Then there is Meta-Object Facility (MOF) that is the heart of Model-driven architecture ,it provides means for building new modeling languages or transforming models from different languages to each other.

MDA can be understood by its meta-model mechanism.

A model is an abstraction in the real world , a meta-model is an abstraction of the model itself. Thus the model has to conform to meta-model.

Advantages of Model-Driven technology are as follows :

- Enable Scripters to contribute to enterprise development.
- Increase flexibility.
- It is faster as it has a higher level of abstraction.
- It focuses on testing thus leading to less error.
- It makes a robust software that is less prone to changes.

Disadvantage of Model-Driven technology are:

- Model Driven introduces a lot of rigidity.
- The modeling environment does not always support version control
- Cost of failure of the system is high.

Case Study: AngularJS

The MDA can be used to generate software code from the models. The development team builds software models using UML at the PIM layer of MDA. These models can be transformed into lower level models i.e the PSM level and code level .

AngularJS is a popular javaScript framework , one example of a Model-driven approach.

It has basically following features:

- 1) MVC controller : Where model is data shown to user , view is what seen by user and controller corresponds to logic to control or modify the data.
- 2) Template :It is a group of HTML that gets converted to DOM.
- 3) Two-way binding: Which changes the view with respect to the model.
- 4) Dependency injection: It lets AngularJs to load all dependencies before processing.
- 5) Directive:Templates that function as programmed.

Fig : Overview of AngularJS[10]

The above figure is the overview of angularJs that uses Model-driven approach for development of web based applications. Below are the steps that are used to get source code template in AngularJs.

Step 1: UML modeling is used to build models for an application based on UML profile of AngularJS.

Step 2: Model exported to XML format.

Step 3:XML file used as input to Model-to-code transcoder.

Step 4:Step 3 transformation result is AngularJS source code template.

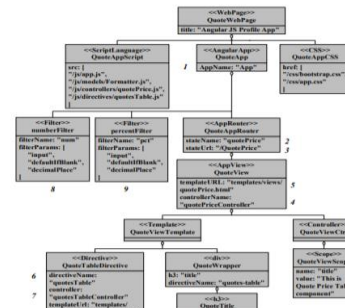


Fig: UML profile of AngularJs.[10]

IV. CONCLUSION AND FUTURE STUDY

In our paper we did an extensive study on how the Software Product Line approaches for web- based applications , studied each of the approaches in detail and its related examples which implement these software product line techniques for a easier development and standard approach to develop web applications which are the most important aspect of the internet today.

As for future work there are a large amount of new technologies that are emerging in the market, which uses these same SPL technologies or modify them and use them to achieve a robust and reusable product , which would be stable.

As such we cannot conclude as which SPL approach is better as it entirely depends on the requirement of our web-applications developed. But a large amount of new technologies or frameworks are emerging which implement the model-driven approach like MEAN stack i.e MongoDB, Express.js, AngularJS (or Angular), and Node.js or the Node.js. Also there are new PHP 7 emerging which uses object-oriented technology and aspect-oriented approach both.

Comparative analysis:

The analysis is as given in the table.

	Object Technology	Aspect Oriented	Model Driven
Advantages	-Reduced maintenance -Real world modelling -Improved reliability and flexibility	-Compliments object oriented -Modularizes cross-cutting concerns -Increases maintainability and understandability	-Enables scripters to contribute to enterprise development -Increases flexibility
Disadvantages	-Requires more memory space -More susceptible to security breaches	-Not well documented -Runtime overheads	-Not high-level enough to allow powerful abstraction -Cost of failure is high
Example	-Java -Python	-PHP -Spring Framework -COBOL -JavaScript -.NET Framework	-Angular JS

REFERENCES

- [1] SEI Product Line Hall of Fame web page, http://www.sei.cmu.edu/productlines/plp_hof.html
- [2] International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.2, April 2011
- [3] Ouali, Sami & Kraiem, Naoufel & Ben Ghezala, Henda. (2011). Framework for Evolving Software Product Line. International

- Journal of Software Engineering & Applications (IJSEA). 2. 10.5121/ijsea.2011.2204.
- [4] Odysseas Papapetrou and George A. Papadopoulos. Aspect Oriented Programming for a component-based real life application: A case study. 2004 ACM Symposium on Applied Computing
- [5] François Coallier, Roger Champagne "A Product Line engineering practices model" Department of Software and IT Engineering, École de technologie supérieure, 1100, rue Notre-Dame Ouest, Montréal (Québec), H3C 1K3, Canada
- [6] H. Wang and J. Yang, "Research and application of web development based on ASP.NET 2.0+Ajax," *2008 3rd IEEE Conference on Industrial Electronics and Applications*, Singapore, 2008, pp. 857-860. doi: 10.1109/ICIEA.2008.4582637
- [7] Iqbaldeep Kaur, Navneet Kaur, Amandeep Ummat, Jaspreet Kaur, Navjot Kaur "Research Paper on Object Oriented Software Engineering"
- [8] Mik A. Kersten and Gail C. Murphy. "Does Aspect-Oriented Programming Work?" . *Communications of the ACM*, October 2001.
- [9] Mik A. Kersten and Gail C. Murphy. Atlas: A Case Study in Building a Web-Based Learning Environment using Aspect-oriented Programming. *ACM SIGPLAN Notices* September 2001.
- [10] W. Chansuwath and T. Senivongse, "A model-driven development of web applications using AngularJS framework," *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, Okayama, 2016, pp. 1-6.
- [11] Fatolahi, Ali & S. Somé, Stéphane & Lethbridge, Timothy. (2011). Model-Driven Web Development for Multiple Platforms.. *J. Web Eng.*. 10. 109-152.
- [12] M. Voelter and I. Groher, "Product Line Implementation using Aspect-Oriented and Model-Driven Software Development," *11th International Software Product Line Conference (SPLC 2007)*, Kyoto, 2007, pp. 233-242. doi: 10.1109/SPLINE.2007.23