

## Introduction

The web has become a very important and popular platform that millions of users are familiar with. The maturity of the web lays the foundation for a move towards more complex web applications that is expected to replace or take over for many desktop-programs.

This next generation of web applications is known as Rich Internet Applications (RIA). RIA technologies introduce new possibilities both in visual richness, and usability. Asynchronous JavaScript and XML (Ajax) is one of these new technologies.

Test-Driven development (TDD) is a programming technique that focuses on conformity of code to tests. TDD has been embraced by many developers. With the new kind of applications that Ajax represent it becomes interesting to see which challenges that Ajax introduce to TDD.

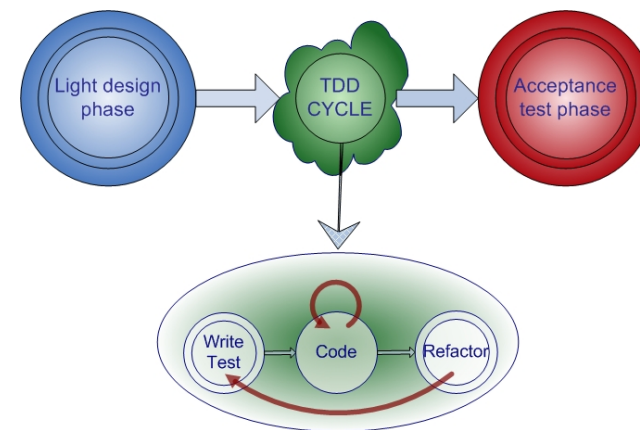
We have also investigated the advantage of design patterns, and discussed possible standardization issues when using Ajax.

## Test-Driven Development

TDD is considered as one of the Agile software methodologies, and share their fundamental ideology. A TDD managed software project will introduce a light design phase, a defined test cycle in the implementation phase, and an acceptance test phase.

The light design phase makes later changes easier to maintain and update. The TDD cycle introduce tested and refactored code that will affect the time spent in the acceptance test phase.

The unit test suite produced will work as a part of the system documentation.



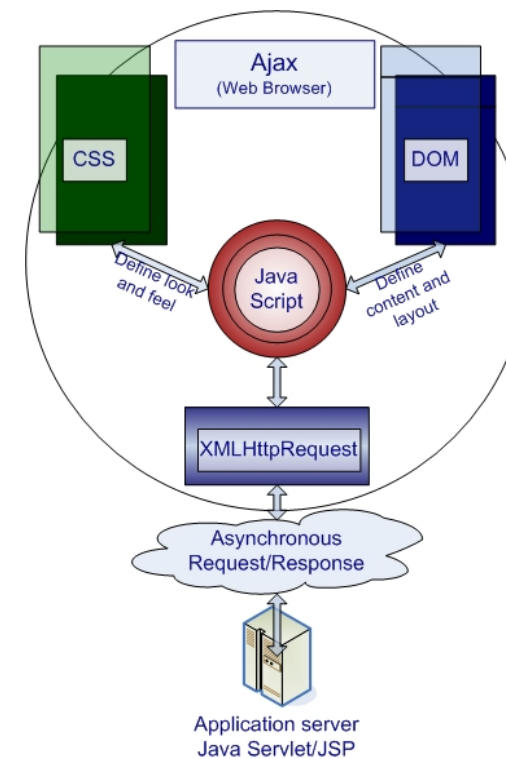
## Ajax

What makes Ajax interesting is that it is based on existing web standards like HTML, CSS, DOM, XML and JavaScript. Modern web browsers are in a sense already Ajax-enabled. Ajax unlike other RIA technologies does not require the user to install various plug-ins.

Unlike ordinary web applications an Ajax application would be able to change and update parts of the view without reloading the web page entirely. A user could for instance read e-mails in one part of the webpage while another part of the web page is updated automatically with new incoming messages. This update is done by the web application calling a web service asynchronously. When the response has been completed the view is updated without interrupting the user.

Any Ajax application would rely heavily on JavaScript. JavaScript has not always been taken seriously by developer but this new dependence requires a new view on JavaScript. In our thesis we also examine how an Ajax application can be

structured similar to an object oriented way of thinking by separating code and through the use of design patterns and frameworks.



## Results

We have identified challenges attached to the usage of TDD and JavaScript development. To be able to create unit test cases for JavaScript code the developer need a test framework able to test JavaScript. In our prototype development we used the JUnit framework.

- Our work show that unit testing of asynchronous request/response can only be achieved by forcing the unit test to pause until a response is received, or by the usage of a mock object to simulate the behavior of the XMLHttpRequest object.

- The creation of mock objects is not easily done in JavaScript. Lack of object oriented structure hindered us to achieve success.

Ajax asynchronous nature makes it harder to predict when changes to the view will occur. Careful design and structure becomes important. There are a growing number of design patterns for Ajax that can help the programmer and make the application more object-oriented. Frameworks such as Direct Web-Remoting (DWR) can also be helpful in adding a new abstraction level and simplify the programming task.

One the success criteria for Ajax are the open standards that it is based on. We have identified a possible standardization issue attached to the XMLHttpRequest object. As the object is not standardized, web browsers interprets it different ways.

## Conclusion

JavaScript has not always be taken seriously but we have shown how Ajax applications can be developed in a more structured way using UML, design patterns, frameworks and TDD.

We have shown that there are challenges attached to use of TDD on Ajax enabled web development. Overcoming these challenges is possible to a certain extent, but we hope to see better support for this in future unit testing frameworks.

Ajax is based on open web standards except the XMLHttpRequest object. A W3C work group has released a working draft. We suggest developers to monitor this process closely.