# Book Review

**Aaron Swartz's A Programmable Web: An Unfinished Work**
**Aaron Swartz**
www.morganclaypool.com
ISBN: 9781627051699

DOI 10.2200/S00481ED1V01Y201302WBE005

Based on the invitation from James Hendler, the editor of the series "Synthesis Lectures on the Semantic Web: Theory and Technology" Aaron Swartz prepared a draft on Programmable Web, but before its completion, he passed away.

It is true that we yet to arrive at a concrete understanding of semantic web. Aaron tried to define it by using efficient programming and rules. The eight chapters even brief enable the readers to gain understanding the complexities, particularly the programming aspect.

We spend considerable time to sit before the web world, travel, stay and do work. After the introductory chapter, he outlines the concept URL and described how they are built and how they are structured. Root concepts and designs are well illustrated with suitable examples. In the subsequent chapter he explained how the hybrid of web is adopted, which is termed as "Representational State Transfer" or REST. In the next chapter he details the intriguing part of the export and import function.

API is required for data-intensive website. API helps the exchanging basic pieces of data between software. Thus this chapter provides building of API with good amount of illustrations.

The official RDF query language is called SPARQL(SPARQL Protocol And RDF Query Language. QUERIES AND DUMPS play a central role in building a database as described in the chapter 7. Open data and open source are explained comprehensively in the next chapter. In the last chapter on Semantic Web, he focussed the angle of web programmes.

The work is even brief can serve as a good compendium in Web programming.

**Ezendu Ariwa**
**Bedfordshire University**
**UK**

**BOOK REVIEW**

MODEL-DRIVEN SOFTWARE ENGINEERING IN PRACTICE
Marco Brambilla, Jordi Cabot, Manuel Wimmer. Morgan Claypool, 2012
ISBN: 9781608458820

Model based software development differs from the conventional software development process and used in conjunction with a range of agile techniques. Despite improvements in third generation programming languages and runtime platforms, the levels of abstraction at which PLAs are developed today remains low-level relative to the concepts and concerns within the application domains themselves, such as manually tracking the library dependency or ensuring component composition syntactical and semantic correctness. [1] A promising means to address this problem involves developing PLAs using *model-driven engineering* (MDE) [2] which involves systematic use of models as key design and implementation artifacts throughout the software lifecycle. Thus to induce the works on architectures and models, researchers have been working to create structured models. The literature in this area hence proliferates; where the current edition [3] tries to address some of the significant aspects.

This book is structured with 11 chapters written not much comprehensively, but with a focus on basic as well as practice aspects. The chapter one is the overview of the contents of the book followed by a chapter on principles of MDSE wherein the authors have presented a discussion on classification of models, models use in industry and the negative side of the MDSE. We wonder to see such a mixture of the different discussions in one chapter! The next chapter on use cases is quite interesting. The next chapter provides a view on Model Driven Architecture which is the core of the scope of the book. The modelling levels and mappings are described with the help of architecture briefly in this unit.

The next unit presents the basic considerations on the adoption of MDSE and how it can be merged with the software development approaches of many kinds. The chapter six deals with the modelling languages which provides many diagrams and a good amount of discussions on UML. The next chapter illustrates clearly how a new model can be derived and what constituents form together the new models.

Models are merged, aligned, refactored, refined and translated to get implemented in the model transformation. [4] The chapter on Model-to-Model Transformations illustrated the transformation languages. The next chapter is the continuation of the previous one which deals with Model to text transformation. The chapter ten deals with the management aspects of the model-driven software architectures. The last chapter provides a very brief summary of the book.

This book is written in lucid style with many illustrations and diagrams which would enable the readers to gain good understanding. The discussions given in the text are brief and call for more descriptions in the future works.

**References**

[1] Gan Deng, Douglas C. Schmidt, Aniruddha Gokhale, Jeff Gray, Yuehua Lin,Gunther Lenz Evolution in Model-Driven Software Product-line Architectures**.** Book Chapter 10.

[2] Schmidt, D. C. (2006). Model-Driven Engineering, *IEEE Computer*, February, p. 25-32.

[3] Marco Brambilla, Jordi Cabot, Manuel Wimmer. (2012). Model-Driven Software Engineering in Practice. Morgan Claypool. ISBN: 9781608458820.

[4] Shandal, S., Kozaczynski, W. (2003). Modle Transformation: The heart and soul of Model driven software development, *IEEE Software*, 20 (5) 42-45.

**Daisy Jacobs**
**University of Zululand**
**Natal, South Africa**