WEB APPLICATION FRAMEWORKS

Jargon fills our everyday lives on the Internet. The days when HTML and JavaScript used to be "cool" technologies are long gone. They are still very much an underlying part of the Net, but Web developers and designers create websites in a more powerful and sophisticated manner today — so much so that the term "website" does not do justice to the systems that deliver content to our Web browsers. The "websites" we visit aren't just static information, but live and dynamic forms of information that interact with us. Hence, in the world of frameworks, they are called Web applications. I will explain the key concepts of this domain, so that by the time you are done reading this article, you will be able to hold a meaningful conversation on Web Application Frameworks.

Let me begin with the concept of the framework. If you look at the Wikipedia definition, it's filled with even more unfamiliar terms. Hence, I will break it down into simple language for you.

Imagine a cube made of only wires (like the structure of graphite that we all studied at school). This is called a wire-frame model of the cube. We can stick paper sheets onto the frames, wrap threads on the wires, paint them, and do lots of other modifications. We can do these because the model, or the cube (the "wire-frame"), allows us to. If I were to port this metaphor to the software world, the model of the cube (a bunch of wires) would become a collection of libraries.

Libraries are pieces of code offering a specific set of functionalities. Now, just like the cube example, the way to use these libraries is through software functions (or "methods" for all ye Java folk!). These functions are called the API (Application Programming Interface) of the particular framework. Thus,

a software framework is a collection of
libraries that can be combined with custom (i.e., user) code through an API.

Extending this concept to Web applications, there are frameworks available that are called (you guessed it) Web Application Frameworks. The libraries include functionalities like data access (to databases), session management (for multi-user capabilities), etc.

What do Web application frameworks do?

These have a wide array of features, some of which I will describe here:

- **URL mapping:** Frameworks, owing to their power, tend to have complex engines. In order for the engine to understand the desired operation, the URLs used in frameworks can get ugly (for the human user, as well as for search indexers, crawlers, etc). To simplify the URL into a more legible form, the URL mapping feature comes to the rescue. For example, you may have seen, on LFY's earlier website, a URL
  like www.lfymag.com/index.asp?id=13. Instead of this, what if it was mapped to www.lfymag.com/home?. That would not only be simpler to understand for the user, but for a search crawler too. In short, URL mapping is URL simplification.
- **Web templates:** A template is nothing but a predefined structure into which we can plug data or information as required. In the context of the framework, this is code that generates the remaining content on-the-fly, as required, on top of a predefined static set of pages. This way, the actual number of hard-coded pages drops drastically, and the performance of the application goes up significantly. Moreover, the data is handled more efficiently; only the required data must be retrieved (usually from a database backend).
- **Database:** Continuing from the point above, all the major frameworks today support a large number of available database products, from a variety of vendors like MySQL, Oracle, Microsoft SQL Server, etc. Database functionality for the Web application is exposed by the framework in its API, enabling functions like adding or deleting records, fetching data, and the like.

- **User management:** This is a very important feature in a framework. A Web application must support user logins, roles and all access-related features, which are powered by the framework's security API.

Behold! Architecture

A cornerstone in the concept of frameworks is the architecture. Taking forward the example of the cube, as in any structure, architecture is an important aspect. From the point of view of the Web application framework, architecture means the way the framework is built and developed, a popular example being the MVC (Model View Controller) architecture.

There are books on the subject, so explaining it completely is beyond the scope of this article. In short, the speciality of this architecture is that it splits the framework into three blocks or parts: the Database Model, the User Interface (View) and the Business Logic (or controller — the brains of the system, the glue that holds the other two blocks together).

Popular frameworks

I will list three popular Web application frameworks called Content Management Systems (CMSs). A CMS is a high-level framework that is used for rapid application development (RAD). If you need to quickly deploy a website/Web application, a CMS is the way to go. It has all the pieces in place and is ready for customisation. It has an inbuilt control panel, administration sections, configuration panels and guidelines to skin and plug in more functionality. As the name suggests, it manages content or data as an inherent feature.

Joomla!

The Joomla! Project has its roots in Mambo, the product of a company called Miro Construct Private Limited. The Mambo CMS, initially a closed-source proprietary system, is now licensed under the GPL.

- Code platform: PHP
- Database back-end: MySQL

Drupal

The brainchild of Dries Buytaert, Drupal (in Dutch, druppel means ‘water drop’), is another popular CMS. It has a large and ever-growing community, along with user groups that have been formed around the world.

- Code platform: PHP
- Database back-end: MySQL/PostgreSQL/SQLite

Django

Developed by Lawrence Journal-World and released under the BSD License, this CMS is named after the jazz musician Django Reinhardt. It’s built purely using the MVC architecture, and emphasises the DRY (Don’t Repeat Yourself) software engineering principle, which, to put it simply, reduces the redundancy of data/information/code in the system.

- Code platform: Python
- Database back-end: MySQL/PostgreSQL/SQLite/Oracle

The depth is infinite in the world of frameworks, as it is in knowledge everywhere. It’s time to pause and ponder over all this information that has emerged. Now that “Framework” is no longer an alien term to you, go ahead and explore! May the frameworks be with you.

Source : http://www.opensourceforu.com/2011/03/all-you-need-know-about-web-application-frameworks/