

AppML : An Idea of Mordern Web Development

Miss Priyanka Mangal¹ Mr. Akhilesh Dubey²

Lecturer, CSE
MIT Mandasur
Lecturer , CSE
MIT Mandasaur

Abstract: AppML combines the most up-to-date techniques and ideas of modern web development. AppML is very easy to understand, and develop. AppML is not a programming language. It used to describe applications. By using AppML you can create Internet applications without programming. AppML allows the programmer to redefine both data and functions while the application is running. Since AppML applications are written in XML, AppML applications are self-describing. In this paper ,we are just providing all the key concept related to AppML .

Keywords : AppML,XML

Introduction

Writing Web applications is a lot of work: Creating forms, programming business logic, developing database access. As part of the Mozilla project, the Extensible User Interface Language (XUL) created a lot of buzz around the concept of declaring -- rather than programming -- user interfaces. Instead of writing code in a particular programming language, an XML document contains a description of all the used GUI elements and their connections.

Along these lines, fellow XML site author Jan Egil Refsnes has developed an XML vocabulary for defining whole Web applications: Application Markup Language (AppML).

AppML is built around a handful of concepts that should be familiar to every Web application developer:

- database
- reports
- filters
- lists
- forms

<AppML> combines the most up-to-date techniques and ideas of modern web development, with focus on low cost, speed, and simplicity:

- Service Oriented MVC Architecture

- Extremely Low Bandwidth Consumption
- Optimized for Cloud Computing
- Full Separation of Content from Presentation
- Intelligent, Agile, and Rapid Web Development
- Highly Scalable and Testable
- Easy Configuration and Reconfiguration
- Smart Support for User Accounts and Roles

<AppML> uses MVC Architecture
MVC stands for Model, View, Controller
The Model describes your application.
The View displays your data.
The Controller controls your application (of course).

Modelling Language:

A modelling language is used to express information or knowledge in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure. There are various available modelling languages. AppML is one of them. AppML is a application modelling language which is used to extend attribute of HTML. It is also called self describing language. In this paper we are discussing about additional features of this modelling language than other available modelling languages.

XHTML

The specification must be short, formal and concise

What is AppML?

AppML stands for Application Markup Language which is used to describe data, application and exchange data using XML. It is open source initiative. The main purpose to use this modelling language is : Platform independent application, compile free and expandable application. It is also reduce our development cost. HTML is used to describing static documents only ,it is not suitable for describing web application so for describing web application AppML add some data attribute and provide controller to HTML. AppML is not a programming language. AppML is used to describe applications. AppML uses the Internet standards XHTML and XML. Traditional applications are precompiled, with predefined data structures and predefined functions, while AppML allows the programmer to redefine both data, and functions while the application is running.

AppML History In 1999 Refsnes Data developed the first version of AppML. AppML was based on HTTP request communication between web client and the web server. Later this method became well known as AJAX. In September 2000, a development project for a large Norwegian customer was started. The goal of the project was to convert a huge information system from a Windows desktop application to a modern Internet application, using only AppML. The AppML-based system was launched in 2001, several months before schedule, as the world's first commercial AJAX application. The project was a huge success, with development time reduced by 75% compared to ordinary web development. Since then, new applications have been added, and the system now covers over 1000 running applications.

AppML Design Goals

- AppML applications must run over the Internet
- AppML applications must be platform independent
- AppML applications must use Internet standards only (HTML, CSS, JavaScript)
- AppML applications must support a variety of application needs
- AppML applications must be self-describing
- AppML applications must be easy to develop, maintain and change
- AppML applications must be future proof
- AppML must be compatible with XML and

How does it work?

You can **describe** the elements of your application using AppML. You **save** the description as an XML file on your Web server, and you ask a Web Service to **execute** your application. Anytime you want to **change** your application, you just have to **edit** your AppML description. Your Web Service will take care of the rest.

AppML Concept:

AppML is built around a handful of concepts that should be familiar to every Web application developer:

- database
- reports
- filters
- lists
- forms

Database

The <database> element defines a record set in a database, by specifying a main table and optionally a SQL select statement for the subset of data to be worked on, as well as the primary key fields for that table. An example for a CD collection could look like this:

```
<database>
  <connection>music_db</connection>
  <maintable>cd_catalog</maintable>
  <sql>select * from cd_catalog</sql>
  <keyfield>cddb_id</keyfield>
</database>
```

Reports

Reports are read-only representations of data in table format, specified by a database element like above. Various elements can be used for specifying the layout of the report table, defining headers and footers. Defaults are provided for all options in order to achieve results quickly:

```
<htmlreport>
<sql>
select artist, title from cd_catalog where
year=1997;
</sql>
<header>
<h1>Michael's CDs of 1997</h1>
</header>
<footer>
```

courtesy of AppML

```
</footer>
<table border="0" class="app">
<thead>
<tr>
<th class="cdheader">Artist Name</th>
<th class="cdheader" align="left">CD Title</th>
</tr>
</thead>
</table>
</htmlreport>
```

AppML Features:

1.AppML is a Declarative Language

AppML is **not** a programming language. It is a **declarative** language, used to **describe** applications. Traditional applications are written in a programming language and **compiled**, with predefined data structures and functions. AppML allows the programmer to redefine both data and functions **while the application is running**. Since AppML applications are written in XML, AppML applications are self-describing.

2.AppML is Browser Independent

Since AppML only uses internet standards like HTML (XHTML), CSS, XML, and JavaScript, AppML will run in all browsers.

3.AppML is Open Source

In this tutorial there is an AppML case study, where AppML is used to create a complete application. At the end of the case study, there is a page where the full AppML source code can be downloaded.

4.AppML is Extensible and Flexible

One of the nicest thing about AppML is that it is extensible. You can use AppML to describe existing applications, or only a small part of an application. You can also add your own elements to an AppML description and extend the capabilities of AppML to your own needs. Since AppML is so flexible, and can be used to describe only small parts of an application, you can start using AppML today! Start using it against an already existing application. Rebuild a small part of the application.

The AppML Description

This is a simple AppML description:

```
<appml>
<title>CUSTOMERS</title>
<database>
<connection>northwind</connection>
<keyfield
type="text">customerid</keyfield>
<maintable>customers</maintable>
<sql> Select CompanyName, City,
Country from Customers </sql>
</database>
<filters> <query>
<field>CompanyName</field>
</query> </filters>
<htmlform /> <htmllist /> <htmlreport
/>
</appml>
```

The AppML description above describes a connection to a database and an SQL statement to extract some data from the database. Copy the example above into Notepad and save it on your Web server as "example.xml". You have now created your first AppML application.

Executables Will Die, JavaScript Will Live

Compiled executables (compiled from languages like C or Java) cannot run on different hardware. Executables (EXE files, ActiveX and COM objects, DLL-files) are components that prevent the development of applications that can run over the internet. Future application will not use, or rely on, executables or any other components installed on the client's computer.

Web Applications Will be Internet Services

The history is full of large, purpose built, applications. Many of these died very quickly, because they could not survive requirements changes.

Applications should be flexible, generalized, and gracefully adjust to changes, without being crumbled or destroyed.

Applications should be able to scale from supporting a few to millions of requests per day. Applications should be able to spread from one server to many, or to move between servers, without breaking the application.

Applications should be able to cooperate with other applications.

Applications should not contain large masses of

code.

Applications should be broken down into smaller services, that are easy to create and easy to maintain.

Applications should be a set of Internet services that can return data to submitted Internet requests. Applications should request services via standard Internet protocols without maintaining a permanent connection to the server.

Future Applications Will be Easy to Create and Edit

Clients and servers will exchange data in an easy understandable way.

Applications will not be coded, if it can be avoided.

Applications will be created and modified, by editing models, not by editing code.

Application descriptions will be readable by humans.

Application descriptions will be self describing.

Applications will be written by users, not programmers.

Three Little Web Developers...

Once upon a time there were three little web developers developing a new web site.

1. The first web developer was using AppML.
2. The second web developer was using his favourite server programming language.
3. The third was using a professional enterprise web development framework.

The first web developer had a demo up and running in two days. After collaboration with the users, an

existing prototype was ready in a week. And after two weeks of testing, an intelligent, fast and easy to use, web site was ready to publish.

The second web developer had his web site ready after 6 months. But the WWW had changed its requirements, and was not satisfied. The web developer could not make major changes to his project because it contained too much code. So he started the development of version 2.

The third web developer never managed to complete his work. The professional web development framework was very difficult to use, very hard to understand, and almost impossible to test.

Conclusion:

AppML is an interesting concept that deserves attention from anyone who is determined to slash development time for typical data entry Web applications. It can easily be mixed in with existing Web application architectures. It provides the features to extend the existing capabilities of the HTML. , Jan has implemented a complete Web application for the Norwegian Handball Federation in a fraction of the time it would take with conventional tools.[1]

References:

- <http://www.webreference.com/xml/column69/2.html>
- <http://w3schools.com>