OpenRecipeXML

An open xml based recipe standard.

Basic Working Draft Januari 2005

1. Concept.

Why need a recipe standard? Whats the need?.
I would be surpriced to get those questions. Last few years have made clear that working around standards improves the overall functionality and usability of technology. A standard based system increases usability more then it decreases possibilities.
For example: You have a bussiness running that needs to exchange data with other company's, how will you send over the data so that they will be able to use it? Answer: you will convert your data into a new format (with possible data loss) like MS Word, MS Excel, Adobe PDF, etc... . And at the end of the day you will be using them all since nobody seems to work with the same operating system nor the same applications.

Many people have tried to create a standard way to exchange data over many years but failed to come up with a way to do so. Untill during the late ninethies a format called extensible markup language or XML increased popularity. XML does not explicitly provides a general standard but creates the possibility to easily transform, organise and markup data.
Let's say we have a list of vegetables and their corresponding color. At some point we would like to know how many vegatables are green. We could select those out of or XML structured vegetables list and create a new list listing only those.

XML is a very powerfull way to handle data, it does not provide a standard, it provides a technology to create standards and with those handle data wetther it is structured (like databases) or unstructured (like text).
So strictly spoken we don't have standards, but a standard way to conserve data. This will be very important when working with OpenRecipeXML: the format has to provide data in a structured and extensible way so applications can easily work with just a selection of the data.

Also there has to be a way to handle languages and localisation. Where in England they would use pounds and inches other European country's preffer liters and milimeters.
OpenRecipeXML will have to provide both ways of describing quantity.

The last mayor goal of the OpenRecipeXML project is availability; I everybody would have recipes hanging around on the internet, there would have to be a possibility to list all of those and make them available as one without putting to much bandwidth use upon few servers.
In this case it would become possible to provide a dynamic list of recipes inside a website without having them all stored on your server.
A possibility could be a side project that collects recipes and mirrors them all over the internet (like for example often visited websites do.).

All together we now have a list of things OpenRecipeXML should comply to. Lets list them according to importance:

- **Provide data in a structured and extensible way.**
- **Provide functionality to localise data.**
- **Provide techniques to make data available.**

So, lets make this all possible.

2. Technology.

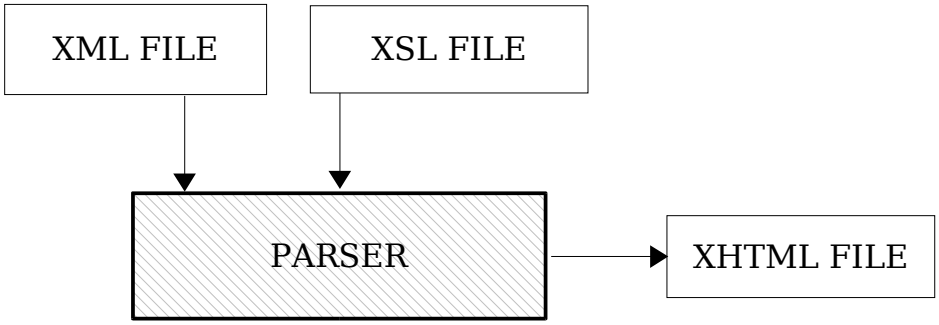
XML is a base technology and therefore has derivates. When speaking about a derivate of XML we mostly use the word language. Xhtml is probably one of the most used xml-languages. It is the succedor of html or hyper text markup language. Xhtml is standard compliant since it uses a xml structure to conserve its data. As I said before xml is a standard way to handle data, thus data can easily be converted from one xml language into another. This means that with a very simple script we can not only convert our xml-recipe into a nice looking webpage, we can also decide what to display or what to leave behind.

The script we need to do so is another xml language called XSLT. At the dawn of the xml popularity a xml-language was developed to provide markup to a document (like old style html does). The goal was to separate markup from the data itself. A similar technology called cascading style sheets or simply CSS already existed but it was not xml-based. So XSL was created. At first XSL was not only marking up data it also had some functionality to order and select data from within an xml file. Soon it became clear that another separation would be needed: one way to transform an xml file into another and another to present the data visualy. Thats when XSLT got in. XSLT stands for extensible stylesheet language for transformation. An XSLT document is parsed (technology used to “understand” an xml file) together with an xml file containing data. The XSLT document provides templates containing information on what to do with the data provided by the xml file. Every template contains a condition or conditions on which the xml document gets evaluated.

Using XSLT technology it will get very easy to create an xhtml file containing a recipe desciped in our xml file, or just a list of the ingredients used by that recipe.
And all will be platform independent since where do not need system specific software to do all of this.

Again it is proved that building upon standards creates possibilities not decreases them.

The following diagram shows how transformation works. Taking a simple recipe and creating a xhtml file to be displayed inside a browser.



In our case somebody will use a program (or dynamic website) and ask for a recipe that needs carrotes, is written in english and uses the metric system to define quantity. The OpenRecipeXML system will have to handle this command and recalculate the recipe.
For that we will need XSLT technology and a very good understanding of structure of recipes and internationalisation.

3. Internationalisation.

3.1 Systems of Measurement.

This will probably be the biggest problem when developing OpenRecipeXML since there are difficult differences between the various measurement systems used today.
Most popular are the Metric system, English System or U.S. Customary System. Both Systems are widely used and differ not only in quantity but sometimes also in defining a unit.

Following is an approximate comparison of basic units between the metric system and the U.S. Customary System; there is also a list with some other measurement systems.

English System / U.S. Customary System	**Metric System**
1 yard	0.9 meter
1 mile	1.5 kilometer
1 pound	0.5 kilogram
1 quart (4 quarts = 3.8 liters)	~1 liter

Table 3a: comparison of some basic system units.

- **F.P.S. System.**
- **Imperial System, U.S. Customary System, English System.**
- **C.G.S. System.**
- **M.K.S. System.**
- **Metric System.**
- **S.I. Units.**

Most of these are very old on no longer used. Modern standards are build upon them though. S.I. Units was build upon the Metric system which was build upon the MKS system that originated at the CGS system. The other family, (Imperial System, U.S. Customary System, English System) was build upon the FPS system. This group is oftenly named together but has many specific units (for example: gallons,).

The S.I. Units or International System of Units is the most widely used and thatfor a good base to build upon. If inside OpenRecipeXML everything is set to this system it would be easy to convert numeric values towards a certain System of Measurement.
So then we would need a database with all units and their corresponding values according to the list of units described in the S.I Units.
So we run into the same question we asked when starting to develop OpenRecipeXML: do we need an Xml standard for describing Units of Measure?
Yes, but lets first explore the Internet if not one yet exists. No, because the problem is far to complicated to simply hash over it.
There has been some good work done by the National Institute of Standards and Technology (NIST) in the U.S: they are working on the draft of UnitsML. There is also a good paper on the subject written by in which they create and comment a suggestion for a xml standard to units.

Still there is no good standard for it so we will have to build upon Frank Olken and John McCarthy's work. They suggest the following:

```
<unitdecl ID="meter" />
                <name> Meter </name>
                <UnitType> Base </UnitType>
                <symbol> M </symbol>
                <dimensionality>
                        <A xml:link="simple"  HREF="#length" />
                </dimensionality>
                <definition>
                        Defined as the some number times the
                        wavelength of Cesium-xxx transition
                        between two specified electronic states.
                </definition>
                <cite>
                        <A xml:link="simple"  HREF="...." />
                </cite>
</unitdecl>
```

Example 3a: sample xml code to define a unit of measure.

This system declares units as base types (ex: meter) and then builds upon them derivates (ex: inch) or composits (ex: meters per second). This will turn out to a good way to handle different but likewise units. And it's also extensible.

Another problem to deal with is the use of special units like a cup of this, a spoon of that: In every kitchen there is a special list of units to describe quantity. We will be using those over more mathematical units like milliliter or kilogram because they are far more simple, flexible and practical since a cup or a spoon are more easily to be found in a standard kitchen then a scale or measuring cup.
These formats could be declared as derivates to the base units of the I.S Units system inside the Frank Olken and John McCarthy scheme.

Since there is no usable database of units (in Xml that is) and since we will be using rather non-official units. We will have to build our own xml-database system.

It comes down to this:

- We have to use the values and units set by the S.I. Units (Metric system) to allow international use of the data.
- According to the above me have to make a list (xml formated) of those standard values to provide metric information and a format to derivate on.
- We have to build a database of units and approximate values towards the corresponding units of the S.I. Units.
- We have to create xslt documents (and other tools) to provide the technology to recalculate the values given by the recipe file.

Of course whenever a good standard way to handle units arises this will all change towards the new system. But we need to have a way to handle them now.

3.2 Localisation and Translation.

Dates and numeric values.
This might be less difficult than can be expected since over the years a lot of work has already been done by numerous people and companies. Most localisations are freely available for use in a very detailed and xml structured form from IBM's developerworks opensource database. It includes a huge number of locales (files containing country or language specific information concerning date, calendar, territorial, etc...) from different open source projects like openoffice and linux.
We will need those to display dates and numeric values.

Also a standard way to set the date information will be needed. Most popular ways to describe a date are the Unix timestamp (ex: 1105061884) and the RFC 2822 format (ex: Thu, 21 Dec 2000 16:01:07 +0200). Of course, as always there is also an ISO standard: ISO 8601 which is more compact and complete then the RFC (ex: 2004-02-12T15:19:21+00:00).
All tree standards should be supported but the use of a unix timestamp should be encouraged because it's completely numeric which makes it easier to convert it. Also it should be announced wich one is used inside the date field.
(we could also use unix timestamps for timetables since it counts secondwise).

Translation of ingredients.
When searching the internet for recipes it is a good thing to know the translation of the ingredients used. We will have to provide a database with general ingredients and a number of translations accordingly. This database will also have to extensible since we will want to add more languages and ingredients over time.
It would be a good thing to use the same technique as with the units: define base ingredients and then translate them elsewhere (a language specific file) referring to the base ingredients.

In this case every OpenRecipeXML document's ingredients could be translated in an eyeblink into any language provided. Of course every document would have to be constructed in english or would need a additional field that describes the locales/language used.

Translation of recipes.
This will be impossible without the use of translation software, so every recipe should have a language field which tells which language is used for the recipe. There should be a special part inside the OpenRecipeXML document for describing the steps of the recipe for every language.

So lets list the conclusions:

- We will use open source localisation files (XML) to provide information on displayal of dates and numeric values etc...
- Date and entries can be of tree forms: Unix timestamps, RFC 2822 format and ISO 8601 standard.
- We will build a list of general ingredients upon which can be build afterwards in order to make translations more easily.
- Every recipe will have a language specific part in which the steps to perform are described according to their language field.