## 1. Introductionary.

In early 2005 the first Working draft was published[1] as an introductionary to the concepts and ideas that could make OpenRecipeXML possible. In this document basic unit systems and differences among them where explained. The document also addressed the problems in localisation and translation. The key technology issue handled by this document was the use of xslt (XSL-Transformations)[2] and xpath[3] for calculating and combining input (xml)data into the result recipe which would probably also carry markup and styling.

Since this documents publishment a lot has changed regarding the technological issues related to data management inside OpenRecipeXML.

Working Draft 2 builds upon this document.

## 2. Data Management.

The ways of handling huge and complex datasets have evolved the last few years (as a matter of fact they always do). The speed of relational databases increases and query languages become more able. The way how data is regarded and stored changes too.

Data in OpenRecipeXML is descriptive. For example: A recipe holds a reference to 2 ingredients. This reference points to a place where information can be found about the specified ingredient. This information describes the ingredient structurally; The system must be able to understand the information in order to consequentially handle an request.
Examine the folowing situation: A recipe reffers to an ingredient by it's unique id: ING027, While processing the recipe the system would need the ingredients name and unit (if not specified). It should thus lookup the ingredient resource located at id ING027 and extract name and unit information.

If every unit and ingredient should be describen, the datasets would be huge and the system would consequentially suffer from slow queries. This is why XML syntax can't be used to describe resources. XML is hierarchic and the parser would have to traverse the whole tree or at least a specified subtree in order to find the needed values 'fullingredientName' and 'defaultUnit' for the queried ingredient.
This is where RDF/XML[4] syntax enters the scene, RDF/XML is a statement centric XML compliant syntax. It constructs datasets out of independent statements instead of hierarchically placing them.

This illustration shows the structural difference between both syntaxes.

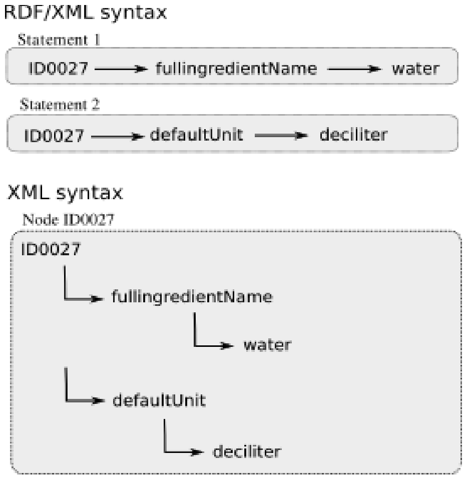

*Illustration 1: RDF/XML syntax versus XML syntax*

In order to understand the statement based approach we need to comprehend RDF's Graph model.

RDF is a w3c recommendation[5] that is developed to address the need for a standardised resource description framework. RDF stores information on a resource like a webpage (or anything else) in triples forming a statement. A statement consists of the triple subject, predicate and object.

For example:

| | |
|---|---|
| subject: | http://example.com/intro.html |
| predicate: | http://purl.org/dc/elements/1.1/title |
| object: | Introductionary Page. |

The predicate *http://purl.org/dc/elements/1.1/title* states that the object *Introductionary Page* is the title of the resource *http://example.com/intro.html.* The predicate in this example uses the Dublin Core Metadata Element Set, a set of standard descriptors for metadata like resources descriptions. In this example the subject and predicate are both uri resources, they locate a resource by URI[6].
This approach differs with relational databases. A values content description is given rather then it's datatype. A predicate could of course still restrict to a certain datatype or range/choice of datatypes.

[1] OpenRecipeXML: Basic Working Draft januari 2005. *http://www.polemian.org/OpenRecipeXML/data/workingdraft_jan2005.pdf*

[2] XSL Transformations (XSLT). *http://www.w3.org/TR/xslt*

[3] XML Path Language (Xpath). *http://www.w3.org/TR/xpath*

[4] RDF/XML Syntax Specification (Revised) *http://www.w3.org/TR/rdf-syntax-grammar/*

[5] Resource Description Framework (RDF) *http://www.w3.org/RDF/*

[6] Uniform Resource Identifier (URI): Generic Syntax *http://www.gbiv.com/protocols/uri/rfc/rfc3986.html*