XU, H and GU, Y. A Web service prototype of mineralogical data presentation using MineML (Mineral Markup Language). *Application of Computers and Operations Research in the Minerals Industries*, South African Institute of Mining and Metallurgy, 2003.

# A Web service prototype of mineralogical data presentation using MineML (Mineral Markup Language)

H. XU and Y. GU
*Julius Kruttschnitt Mineral Research Centre, The University of Queensland, Australia*

In recent years, the mineral applications dealing with more complex data in distributed environment have grown rapidly. Many mineral and mining organizations have to organize internally and globally distributed manufacturing and experiment sites, which share the roles of databases and other multi-services like software programs. Thus, there are commercial and technical pressures pushing the mineral software vendors to move their current service toward Web service.

XML has gained wide acceptance for representing, storing and transferring structured data and documents of many kinds, especially in scientific communities. However, so far there is no such a Markup Language in mineral industry. A XML-based uniform encoding, which is called MineML (Mineral Markup Language), is proposed and developed in this project. The framework of the MineML specification is created. MineML will play the role as shared protocols within mineral domain for archiving, transferring the multi-nature data between the software packages on the desktop over the Internet and sharing the services between users at different sites.

Inspired by MLA's (Mineral Liberation Analyser, JKMRC, University of Queensland, Australia) Mineralogical Data Presentation Program, a proof-of-concept prototype of Web Service is developed here to validate and demonstrate the MineML specification. Certain strategies, specific technologies and tools are adopted to create such an environment. To accomplish that, a set of experimental MineML documents including correlated XML, XSD (schemas) and XSL (stylesheets) documents have been elaborated for archiving, presenting and transferring the mineralogical data information, with the persistence on flexibility and extensibility for generic mineral and mining data handling.

## Introduction

The World Wide Web has emerged as an extraordinary successful medium for transferring information between computers and people. It is also changing the way we think and act in the world of science and impacting the traditional industries such as mining and mineral. Nowadays in this industry many organizations have to organize internally and globally distributed manufacturing and experiment sites, which share the roles of databases and other multi-services like software programs. Especially in recent years, the mineral applications dealing with more complex data in distributed environment have grown rapidly.

Therefore, there are commercial and technical pressure pushing the mineral software vendors to move their current service toward Web service.

Obviously, current available Web protocol like HTML cannot handle the requirements for Web service, where the disparate information to be distributed across a wide geographical and sometimes low quality network facilities.

It was until the coming of XML[1] –XML has gained acceptance for representing structured data and documents of many kinds, especially in scientific world. It did not take long for the scientific community to realize the benefits it could obtain from using XML. Loosely or closely referring to W3C's recommendations and re-using these protocols and tools, a number of content-based MLs such as CML[2], MathML[3], and BSML[4], have been proposed and implemented as the 'XML family'. To date, these MLs in their communities have already achieved significant results.

Lack of XML-aware data encoding and application in the industry, an XML-based uniform encoding, which is called MineML (Mineral Markup Language), is proposed as a new sub-set of XML. As the essential part of MineML, it will allow mineral data information in the form of software capabilities to be distributed worldwide and plant process control to be supervised online and real-time. MineML will play the role as shared protocols within mineral domain for archiving and transferring the multi-nature data between the software packages on the desktop over the Internet and sharing the services between users at different sites.

Certain strategies and specific technologies are required to create such an environment. A sophisticated mineral domain model will be used to build the MineML schema language that is elaborated to describe the shared namespaces and metadata of the mineral XML documents.

A set of correlated XML, XSD (schemas) and XSLT (stylesheets[6,8] will be developed. MineML itself is not designed to deliver application software; however, in this project a proof-of-concept Web service prototype will be built to validate and demonstrate the MineML specification.

The outline of the paper is as follow:

We give an overview of XML and other MLs in scientific domains in the second section. In the third section we briefly discuss the MineML. A detailed description of

MineML-based Web service prototype will follow in the fourth section. A number of MineML document examples will be presented in the fifth section for better and clearer understanding of MineML. In the final section we conclude and explore the further work.

## Overview

### XML—a 'do it yourself' markup language

Promoted by the World Wide Web Consortium (W3C), the extensible markup language (XML) was introduced in 1996 as a simple, flexible, and powerful way for computers to exchange metadata and control information interactively. Very soon it was adopted in different domains as a new data structure standard for data publishing, data exchange and cross-platform computing, particularly in the scientific world.

Similarly to HTML, XML is a language based on tags. However, in XML, the tags are defined by the user, and they can be used to describe the rendering, but also to assign meaning, to the document. In a simple word, XML was designed to describe data, and to focus on what data is.[17]

XML has been designed for maximum flexibility, maximum teachability, and maximum ease of implementation. The syntax rules of XML are very simple and very strict. Because of this, creating software that can read and manipulate XML is very easy too.

As a meta-language, XML provides a format for identifying the data information, but doesn't assure the correct availability of the data information to the recipient. Thus, XML Schema is needed to help to obtain this assurance. The set of user-defined tags and their structure (i.e. how to use and nest the tagged elements) are generally described in a separate file called a Definition Language (XSD), which is a model describing the structure of information. The purpose of a schema is to allow machine validation of document structure, to enable large volume XML transactions over the Web. Now XML Schema has become a W3C Recommendation.

### Modelling the real world

In the plast decade, scientists used XML to develop the intermediate solutions to many of the problems they were facing when exchanging data, thus capitalized many advantages of XML such as human readability, simplicity, flexibility, system and language independence, wide used XML-aware browsers[5], etc.

In fact, technically, XML provides only data syntax. Before exchanging data in XML, however, an agreement inside a scientific community on a common semantics—schema, needs to be carried out. The definition of a schema is a long and difficult process that requires a scientific community to specify the definition of the vocabulary and the grammar used in their field. That is probably the main barrier that slows down the wide adoption of XML.

The W3C has developed and continues to develop a wide range of generic protocols based on the XML syntax to support the ML family, such as *XML Schemas*[6], *XML Query language*[7], *Namespaces specification, XSLT*[8], *XLink*[9], *etc.* However, in general, the W3C does not create domain-based MLs, just leaving that to appropriate organizations and authorities within the domains. So far there are some successful development and implementation of Markup Languages like CML (Chemical), MathML (mathematics), BSML (The bioinformatics sequence markup language), BioML (The biopolymer markup language), etc.

## About MineML

### The motivation

In recent years, the mineral applications handling with more complex data has grown rapidly. Considering the scenarios like these–

- The data with non-restricted formats, which are collected from heterogeneous sources, are sent to Lab to be analysed, processed and modelled. However, they may be converted for multi-times before getting to the processing to meet the document formats of experimental requirement.
- The source data may be submitted for archiving at any time, via the Web or Intranet, within software applications in different location or between different applications on multi-sites. The data are then validated and processed on real time, automatically upon submission. If accepted they are loaded directly into a data warehouse.
- A system of Web services that delivers customized remote-sensing data to Internet/Intranet connected clients, provides the services of remote system monitoring, technical support and supervision, gives the on-demanding feedback.

XML has actually already become the *de facto* standard protocol for communication and information delivery.[18] However, it is not practicably applied without an agreement inside a scientific community on a common semantics. Thus, an XML-based data encoding is needed in mineral industry. This Markup Language needs to be specified for the definition of the vocabulary and the grammar used in this industry, based on the generic protocols and tools provided by W3C's XML and Schema recommendations.

### The design strategy

The vocabulary and syntax are the highest priority of consideration in designing the encoding structures. It may concern the key areas like geology, mineralogy, mineral behaviour, process and flowsheet, equipment and sizing, even economic analysis, etc.

Therefore, not only the professional words and terminology, but also proficiency in schema, data tree hierarchy, Hyperlink structure, Metadata (the data which defines the form of data stored in a database), multi-layer system hierarchy are all the issues that should be considered for this encoding language.

### The goal

Building on top of the World Wide Web Consortium (W3C) encoding recommendations, ISO mineral standards and current mineral implementations, a sophisticated mineral domain model is adopted to develop the MineML specification which is elaborated to describe the shared namespaces and metadata of the mineral XML documents.

MineML is initially intended to be an encoding for mineral data information, however, it can easily convert MineML dataset into presentation formats, such as HTML Web page, chart and spreadsheet, by working with some third party component like SVG (Scalable Vector Graphics).[10]

In general, complying with that in XML specification, the design goals for MineML are:

- MineML should be compatible with XML

- MineML document should be easy to create and process
- MineML should be easily handled by wide variety of mineral software application
- MineML should be designed in formal and concise but open to all for widespread acceptance
- MineML's option features will be kept to the absolute minimum, ideally zero
- MineML document like XML and XSD should be human-understandable and reasonably clear
- MineML can fully interoperate with other MLs like GML (Geography Markup Language)[16].

Here are some examples of the potential advantages of MineML applications.

- Software developers can focus on the distinctive functionality of their program packages, without considering too much about the data formats and the information loss by reformatting and conversion while interoperates with other applications.
- Data collectors and Value-adding service providers/ laboratories will be communicating with a single and simple machine-readable format, thus the working efficiency will be improved significantly. It is also happy for database custodian and regulators who can receive the reports in uni-format and expedite the report process.

## A MineML-based Web service

When we plan to systematize know-how we should model the target according to the feasibility. To test the experimental encoding language, a prototype of MineML-based Web application is designed to approach delivering service for current users of MLA (Mineral Liberation Analyser, JKMRC, University of Queensland, Australia) DPP (Data Presentation Program). It is our initiative to present the quantitative mineralogical data, which were collected from the MLA system, via the Internet.

Two different versions are developed for this prototype: Non-.NET version and .NET version. A set of experimental MineML documents of XML, XSD and XSL have been drafted and elaborated for the demonstration.

### Basic concept and information infrastructure

Certain strategies and specific technologies are required to create such an environment. The mechanism will focus on how to convert the data source into XML architecture and how such architecture can be implemented and consumed by the Web page.

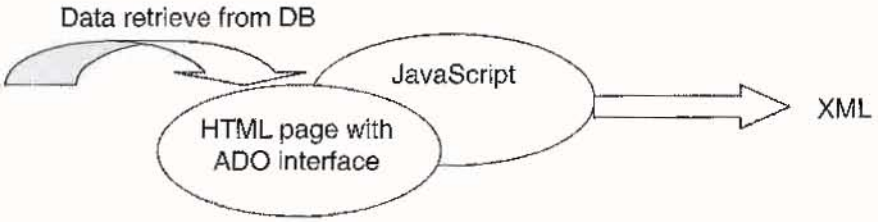
As Figure 1 shows, XML can be created and consumed by a browser. Figure 2 shows how the MineML data are defined by schema. XSD itself is stored on server site and is referred to when requested. A framework of proposed system architecture is presented in Figure 3.

Considering the mass weight of instances of source databases (some are as big as tens of Megabytes), a template database was created from the Windows-based DPP program. It was reformatted according to the MLA DPP result report interface and plays the role as an intermediate database. This stage is called Data Pretreatment, and all the sequent procedures can access the database via the ADO (ActiveX Data Objects)[12] interface.

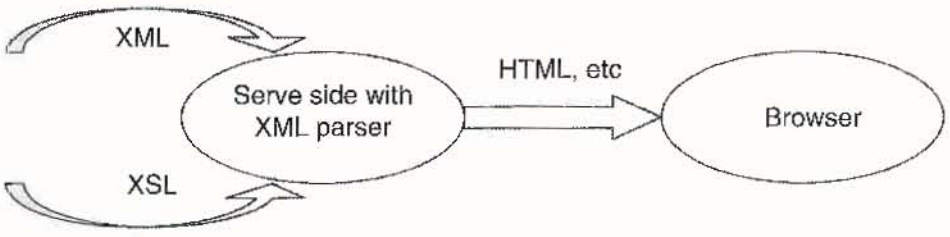
The Non-.NET version prototype is based on the WWW (World Wide Web) augmented by ASP and Microsoft XML DOM (Document Object Model)[13]. Other languages and tools involved here are JavaScript, VB Script, C# and MS XML Parser to process the data and convert XML document on the server. The .NET version is based on MS ASP.NET Framework. The whole system is built on Microsoft IIS Server 5.0 and MS SQL Server 2000. Basically the prototype is a server side application, however, specific format report such as Spreadsheet is sent to the users' browser by working on both server and the client environment.

### System architecture

The actual system architecture diagram for this prototype is




XML can be created directly from a Web page with ADO (Active Data Objects) interface to connect with DB, which is through a JavaScript parser embedded in the page.




XML can be processed on the server with the support of an XSL stylesheet, to create the human-readable documents or forms

**Figure 1. How XML can be created and consumed by a browser**

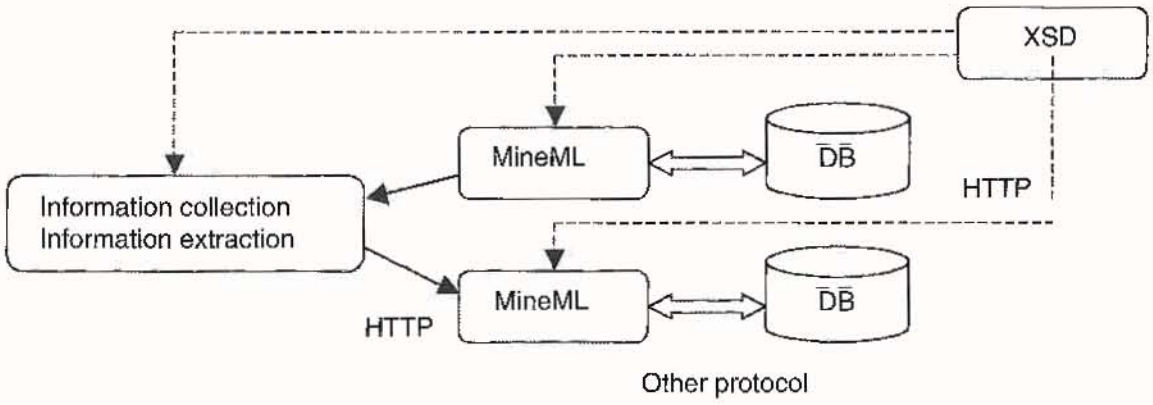



**Figure 2. How the MineML data are defined by schema (XSD), XSD itself is stored on a server site and is referred to when requested**

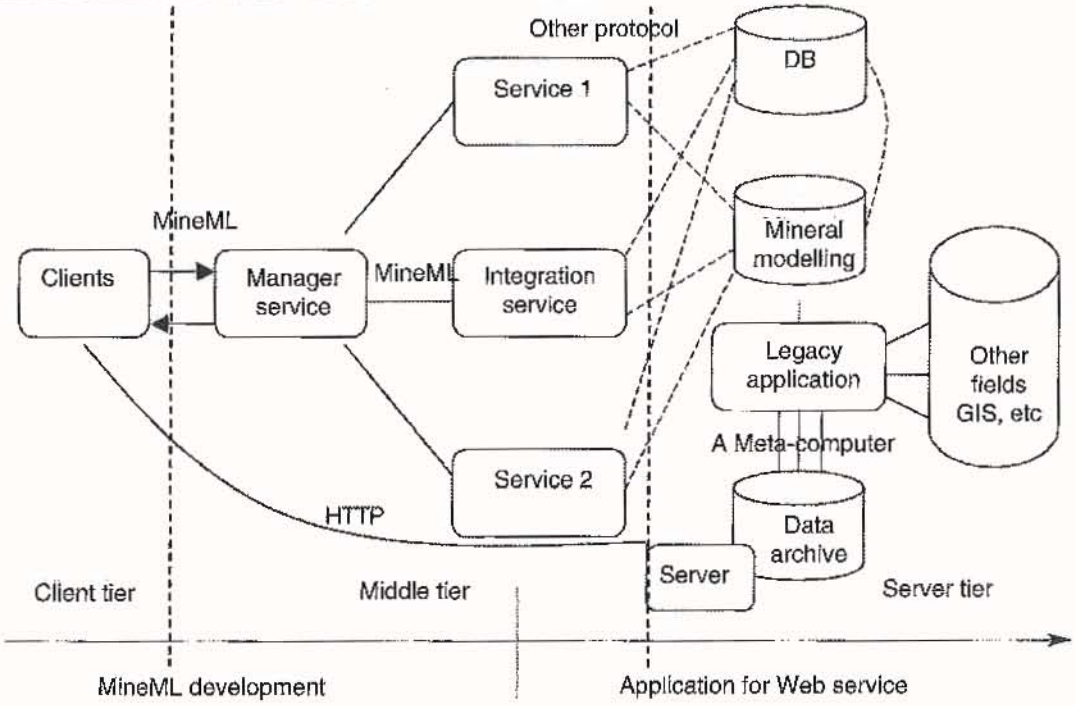



**Figure 3. The proposed prototype of MineML-based Web service**

shown in Figure 4. Basically, it consists of 3 components: Server component, Client component, and ExcelWriter component.

***Server component: Construct MineML document from database***

Microsoft XML Document Object Model (DOM) syntax are used to access and manipulate the data on the server. Storing data natively in XML is inefficient, and the whole scale conversion of existing database is expensive, leaving on-the-fly database to XML conversion the optimal solution.

A design decision was made to use an XML parser on the server, but in some cases (such as directly building XML for creating spreadsheet), parseless alternative is also needed. For both, I implemented three different methods to convert data into XML document or on-the-fly XML stream:

- Constructing an XML tree using the DOM
- Directly constructing XML text without using the XML control
- Building XML by complex DOM method of Microsoft parser.

***Client component: Build Web table and report from XML document***

In the client's, MS XML parser are used to extract data from server. In Non-.NET version, the data comes in XML format and is processed by DHTML using the DOM, CSS (Cascading Style Sheets),[14] and Data-binding to create the HTML page table. For instance, the data island as shown below is created to take care of this task, in which 'recordXML' holds the result data that comes from the server.

Simply said, the DHTML document is structured in the following way: all HTML elements that have to do with the presentation are located at the beginning of the code, followed by the JavaScript, and finally the data island for user's option to use.

In .NET version, XSL stylesheet is adopted to work with

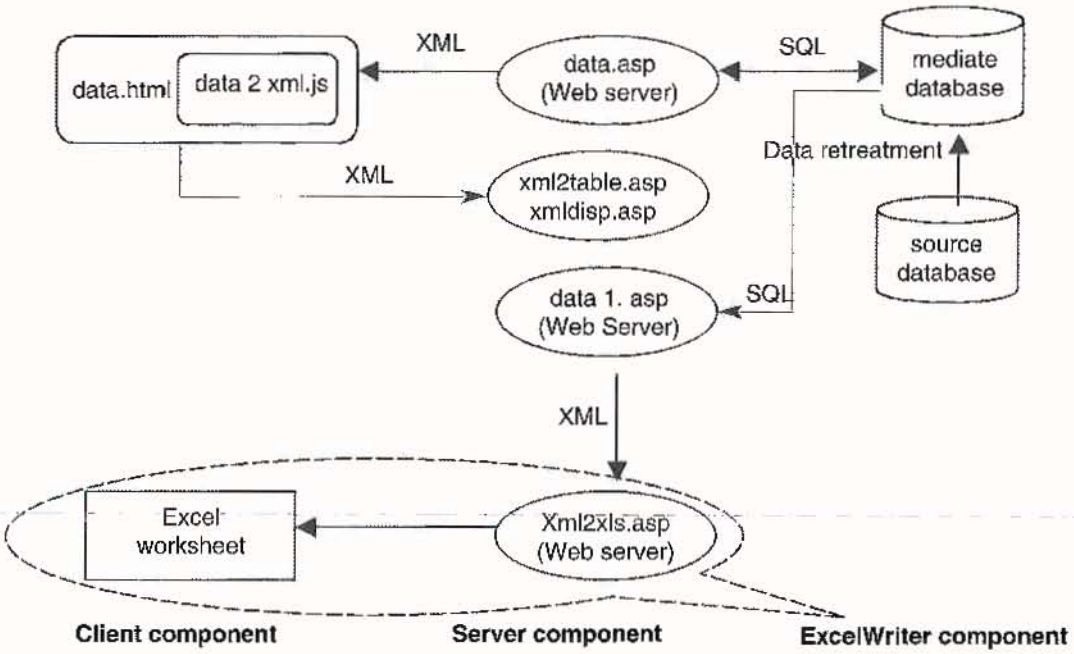



**Figure 4. Server, client, sessions and services communicating with XML**

schema in creating the Web page. As the syntax shown below, the command assigns an XSL stylesheet to an XML document, enabling XSL Transformation or rendering by an XML-compatible browser.

```
<XML ID="recordXML">
<Locking_Table/>
</XML>
```

***ExcelWriter component: Create excel spreadsheet from XML document***

After the transformation of data to XML, some issues are considered on what the interface will be and what kind of format of report will be:

- Easy to view and fast download for clients
- Ability to generate charts real-time online and easy to format the chart appearance
- Server-side component—does not require client to install something on their computer to support the Web page

```
<?xml-stylesheet href="MineralReference.xsl" type="text/xsl"?>
```

- Fully editable data table for client
- Extensible feature allows client-server interaction such as online modelling by user's on-demanding
- Allows client to save the data to disk for their future references.

To achieve that functionality, a third-party component called ExcelWriter (SoftArtisans)[11] is used here to create online Excel spreadsheet. ExcelWriter is a server-side DLL control for generating high-speed Microsoft Excel spreadsheets and charts. In this prototype, presentation-quality Excel reports are generated by COM programming.

### Features

This Web service application keeps the full functionality as the original Windows-based MLA DPP and the familiar interface for users. Users can select the database to be represented from the drop-down list menu on the Web page, and select data report from the tree view control frame which is the driving seat for viewing all of the tables and graphs for these mineralogical data. The data in MineML format is converted and assembled into the table on server side then presented on the Web page as the report. Spreadsheet format working report which contains both data and chart is generated real-time and quickly without requiring clients to install any plug-in.

The expected advantages can be readily implemented in the short-term:

- All the collected MLA mineralogical data can be available in the system
- Up-to-date data can be used at all times
- Interactive feature will be implemented in the system and online modelling tool will be built
- System functionality can be shared between the server site and the user's browser site to work effectively as a whole.

## Modelling the mineral data information in MineML

Not like the tables in the database that only contain the dataset information, the converted MineML document contains well separated chunks of information of the sorts such as:

- Dataset
- Project background
- Data source background
- The people involved
- The concerned date and time
- The operation/event for data processing
- More.

A set of experimental MineML documents are developed with considering all of this information.

### Sample XML document syntax for MineML

For better understanding MineML in a straightforward way,

a few instances are presented here to give a snapshot of the new encoding.

In MineML document specified namespaces are used as the prefix, which are preferred to corresponding schemas. That is the way to apply schema in XML document-since element names in XML are not fixed, very often a name conflict will occur when two different documents use the same names describing two different types of elements. (see Panel A)

The namespace attribute is placed in the start tag of an element and has the following syntax: xmlns:namespace-prefix="namespace". In the examples, the namespace itself is defined using an Internet address like "xmlns:mml='http://*** *** ** ***/mml".

The W3C namespace specification states that the namespace itself should be a Uniform Resource Identifier (URI), which is a string of characters which identifies an Internet Resource. When a namespace is defined in the start tag of an element, all child elements with the same prefix are associated with the same namespace.

Another important usage of namespace is its application in XSL. In the XSL document, you will see that most of the tags are HTML tags. The tags that are not HTML tags have the prefix xsl, identified by the namespace "xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance".

In the following is a block (Panel B) which describes the whole feature of 'document properties'. The 'Document Properties' element is defined in corresponding schema dppmd.xsd.

Then further down is the block (Panel C) for the whole dataset with the element name 'table'. The example given shows the type of the 'table' element is defined in schema dppmd.xsd, whose name stands on 'MLA DPP Metadata'.

## Sample schemas

### *The architecture*

The architecture for MineML schemas are designed in multi-layer hierarchy, from top to bottom are the classes covering different scale in mineral industry. The base is a general class for whole industry wide. Although MineML was inspired by JKTech's MLA DPP Web service, I approach this partly from the general online-processing viewpoint, so it has the extensibility to provide support to industry wide applications.

The specification I adopted is W3C XML Schema Definition Language[6], which was released as the W3C Recommendation in May, 2001.

### *Schema design view*

As described in former section, schema is a model describing the structure of information. In MineML schemas are defined in terms of constraints:

- The content model constraint which describes order and sequence of elements
- The data type constraint which defines the valid units of data in the content model.

For more clearly illustrating the structure of schemas' content model, some examples of schema design view are shown in Figure 5 as the graphical representation of the components. The tree structures and annotations present the elements in the schema.

So far the syntax of the sample MineML documents are only for experimental Web service purpose. Many of the constraints defined here are relatively arbitrary and

**Panel A**

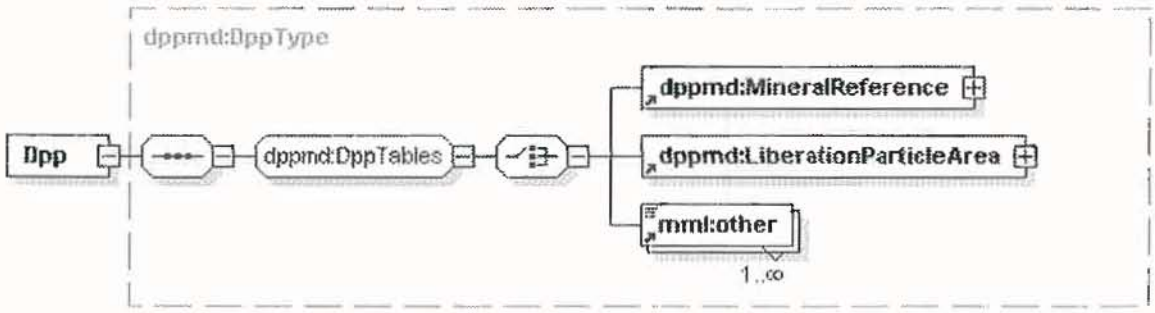
```
<dppmd:MineralReference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mml="http://*** *** ** ***/MML/mml" xmlns:mla="http://*** *** ** ***/MML/mla"
xmlns:dppmd="http://*** *** ** ***/MML/dppmd" xsi:schemaLocation="http://*** *** ** ***/MML/mml mml.xsd
http://*** *** ** ***/MML/mla mla.xsd http://*** *** ** ***/MML/dppmd dppmd.xsd">
```

**Panel B**

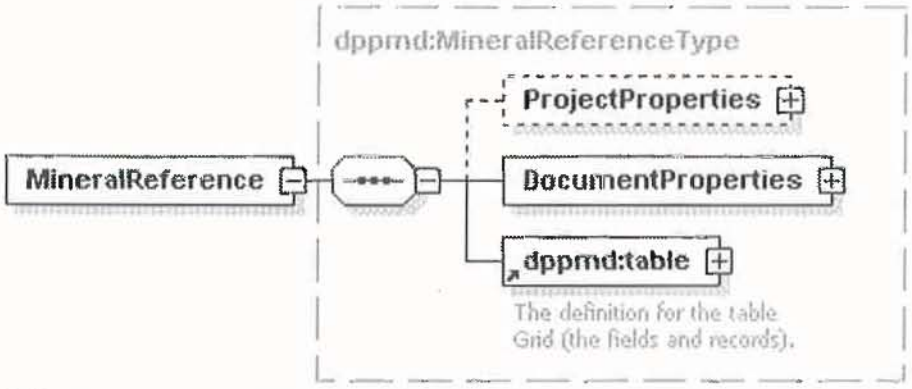
```
<dppmd:DocumentProperties documentID="MLA-DPP-ZincRougherConc._C5-001">
                <mla:description>An xml instance for Mineral ReferenceTable of MLA DPP result using DPP Matadata schema.
Copyright (c) MLA, JKMRC, UQ, Australia. All rights reserved</mla: description>
                <mla:editor>Wally Xu</mla:editor>
                <mla:created>2001-07-29</mla:created>
                <mla:event>
                        <mla:operation>DataAnalysis</mla:operation>
                        <mla:agent>
                                <mla:program>MLA DPP</mla:program><mla:person/>
                        </mla:agent>
                        <mla:DbStatus phase="source">Jan01ZincRougherConc._C5.mdb</mla:DbStatus>
                        <mla:DbStatus phase="mediate">MLA_DPP_Result.mdb</mla:DbStatus>
                        <mla:EventDate>2001-07-01</mla:EventDate>
                </mla:event>
                <mla:event>
                        <mla: operation>DataTransfer</mla:operation>
                        <mla:agent>
                                <mla:program>ASP</mla:program><mla:person/>
                        </mla:agent>
                        <mla:DbStatus phase="mediate">MLA_DPP_Result.mdb</mla:DbStatus>
                        <mla:DbStatus phase="XML">MineralReference.xml</mla:DbStatus>
                        <mla:EventDate>2001-07-01</mla:EventDate>
                </mla:event>
        </dppmd:DocumentProperties>
```

**Panel C**

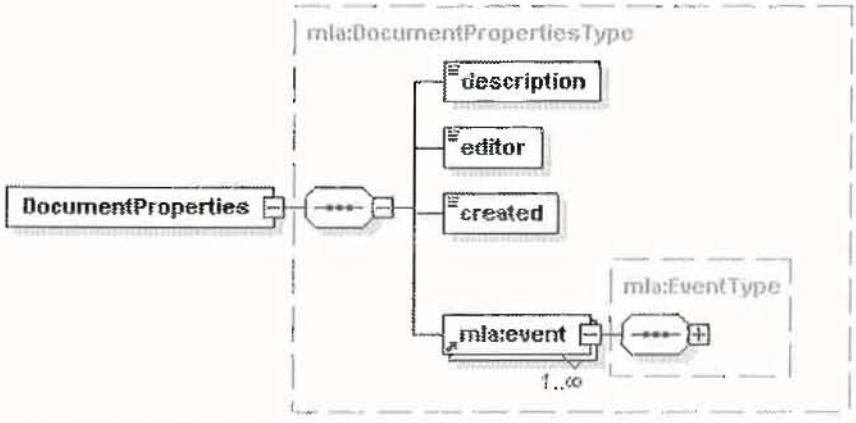
```
<dppmd:table dppmd:TableName="Mineral Reference" dppmd:TableLocation="Mineral Set" dppmd:TableSort="TableGrid">
        <dppmd:description>The Mineral Reference Table under the Mineral Set category lists the minerals which the MLA
system has measured in the sample, the mineral abbreviation code, the mineral density, the mean atomic number (z*) and the elemental
composition for each mineral.</dppmd:description>
        <dppmd:MineralReferenceTableGrid>
            <dppmd:MineralClass>
                <dppmd:MineralName>Apatite</dppmd:MineralName>
                <dppmd:AbbreviationCode>Ap</dppmd:AbbreviationCode>
                <dppmd:Density>3.2</dppmd:Density>
                <dppmd:Mean_z>14.1</dppmd:Mean_z>
                <dppmd:Al>0.0</dppmd:Al>
                <dppmd:As>0</dppmd:As>
                <dppmd:Ba>0</dppmd:Ba>
                ...
                <dppmd:P>18.4</dppmd:P>
                ...
                <dppmd:Zn>0</dppmd:Zn>
            </dppmd:MineralClass>
            ...
            <dppmd:MineralClass/>
        </dppmd:MineralReferenceTableGrid>
    </dppmd:table>
```




It is the basic structure of the DPP metadata element. There are 2 instance elements for description of the table 'Mineral Reference' and 'Liberation Particle Area'




Each instance table element has the features of 'Project Properties' and 'Document Properties', which are defined in corresponding schemas




It gives the information about the properties/background of the XML document like people, date, sponsor, etc.

**Figure 5. Some examples of schema design view**

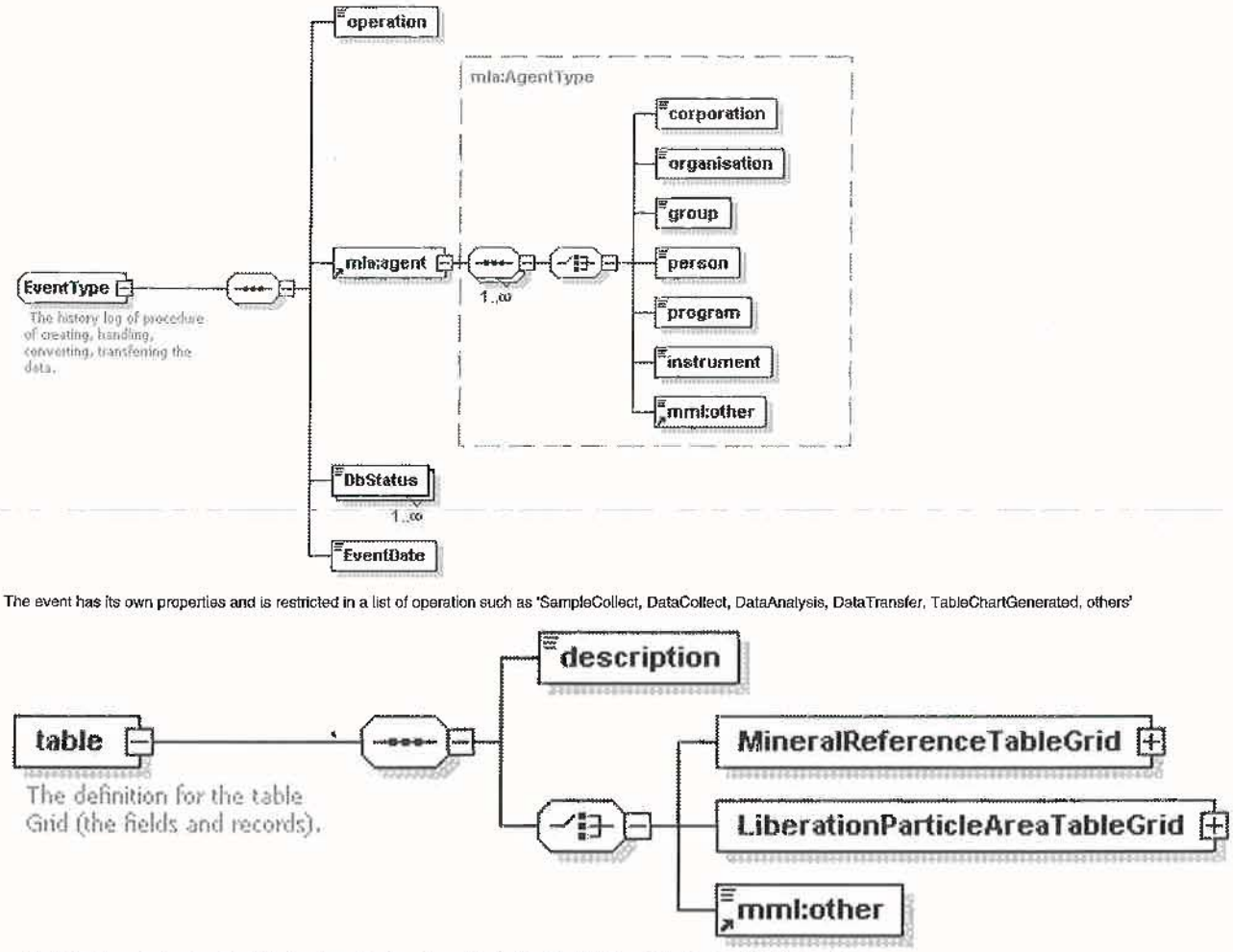



**Figure 5. Two more examples of schema design view**

expected to be revised and extended, but the content model presented here are flexible and rigorous for future development.

## Conclusions and further work

In this project a experimental Markup Language—MineML is proposed as a new data encoding and the protocol between independent services in the mineral industry. A proof-of-concept prototype of Web service is developed to validate and demonstrate the experimental framework of MineML specification.

The primary deliverable Web application will be the demonstration of MineML based Web application used for archiving and transferring the mineralogical data information. However, this prototype is just the preliminary work for MineML project.

Following the previous work, multi Web solutions can be tested such as XML/XSD/XSLT, DirectXML/XSLT, and XML/.NET to approach the best performance. The consistent framework will be carried on to approach the project's final goal: keeping extension and consummating of MineML and developing MineML-based Web service solution, software and API for import/export to legacy applications.

The interoperability between the application's functionality and the data encoding structure will run through the project. However, MineML itself is not designed with delivering any finished, commercial-grade software–MineML need the widespread supporting by mineral software vendors to develop MineML-aware software applications.

The software developers in the industry will have the priority to access the project results and all source code, and will be assisted and encouraged to begin development of implementations based on a widely acceptable MineML specification.

## References


1. eXtensible Markup Language 1.0 (Second Edition). W3C Recommendation 6 October 2000 http://www.w3.org/XML/
2. Chemical Markup Language, http://www.xml-cml.org
3. Mathematical Markup Language 1.01 Specification, W3C Recommendation, http://www.w3.org/Math, http://www.webcq.com
4. The bioinformatics sequence markup language, http://www.visualgenomics.com/products
5. XML: Emmanuel Barillot and Frederic Achard, Aug, 2000. A lingua franca for science, *TIBTECH*, vol. 18.
6. XML Schema http://www.w3.org/XML/Schema/

7. XML Query Language, http:// www.w3.org/XML/Query/
8. XSL Transformations (XSLT), W3C Recommendation http://www.w3.org/TR/xslt
9. XML Path Language, http:// www.w3.org/TR/xpath
10. Scalable Vector Graphics (SVG) 1.0 Specification W3C Candidate Recommendation 02 November 2000. http://www.w3.org/TR/SVG/ ; SVGview: Adobe SVG Viewer plugin, http://www.adobe.com/ svg/viewer/install/main.html
11. ExcelWriter, http://www.softartisans.com/
12. Microsoft ActiveX Data Objects, http:// www.microsoft.com/data/ado/
13. Document Object Model , HTTP:// www.w3.org/ DOM/
14. Cascading Style Sheets, HTTP:// www.w3.org/ Style/CSS/
15. The bioinformatics sequence markup language, http://www.visualgenomics.com/products
16. Geography Markup Language, OGC Recommendation Paper, 20 February http://opengis.net/gml/01-029/GML2.html
17. BOSAK, J. XML, Java, and the future of the Web, 1997. http://sunsite.unc.edu/pub/sin-info/standard/xml
18. BARILLOT, E. and ACHARD, A. A lingua franca for science, TIBTECH, vol. 18. Aug, 2000.