WBF XML Schemas

David Emerson
Senior Systems Architect
Yokogawa
2155 Chenault Drive Suite 401
Carrollton, TX 75019
USA
Tel: +1.972.417.2753
Dave.Emerson@yca.com

KEY WORDS

ABSTRACT
This paper provides background on, and an overview of, the soon to be released WBF XML batch and enterprise-control system schemas. The schemas consist of two sets, one is intended to provide for the exchange of batch data and is based upon the ANSI/ISA 88 standard. The second is intended to serve as a basis for exchanging data between enterprise and control systems and is based upon the ANSI/ISA 95 standard. The organization of each set is described along with examples on how they can be used.

INTRODUCTION
The World Batch Forum XML working group is developing two sets of XML schemas based upon the ANSI/ISA 88 and 95 families of standards. The intention of the working group in this endeavor is to produce XML schemas that

- Provide a useful and faithful implementation of the respective standards,
- Will be widely used by operating companies, consultants and vendors,
- May be used without requesting permission or concern of copyright infringement,
- Fosters the exchange of data required for manufacturing.
The schemas should be viewed as one implementation of the ANSI/ISA 88 and 95 standards. In the case of ANSI/ISA 88 a relational database implementation is defined in ANSI/ISA 88.00.02 clause 5 and the OPC Foundation has defined DCOM interfaces that based upon the standard. The ANSI/ISA 88 based XML schemas provide an alternative to these other interfaces. In the case of ANSI/ISA 95 the XML schemas may be the first implementation of its type. In either case the XML schemas provide a support for these standards to companies that decide to use XML to exchange data; it is not the intent to require the use of XML.

Without the existence of a core set of XML schemas each company, or even work group, would have to develop their own definitions of XML elements and types based upon the standards. This would inevitably lead to numerous variations with enough structural and nomenclature differences to make the exchange of data using XML more difficult than expected. The creation of the WBF schemas will not make XML based data exchange easy, but it should make it easier. Even when the WBF schemas are used to derive proprietary schemas that extend and constrain the originals when the WBF element names and type definitions are used there will be a common footing that can be used to establish a data mapping between applications.

XML WORKING GROUP

The XML working group was formed in 2000. Membership is open to all WBF members and invited guests. Members join by subscribing to the working groups list server and Yahoo! Group. The Yahoo! Group is used to post and download files such as meeting minutes and schema drafts. As of January 1, 2000 there are over 100 members of the working group. Most of the work has centered around two face-to-face meetings that were attended by 14 individuals representing 10 companies (3 operating companies & 7 vendors).

Once the schemas are released a small group has agreed to work on maintaining the schemas if errors are found. This group does not intend to enhance the schemas, only perform maintenance activities are needed to support users of the schemas. If enhancements are needed then the entire working group will be consulted.

SCHEMA NAMES – BATCHML & B2MML

The sets of WBF XML schemas are named to provide them with distinct identities and to avoid using the copyrighted 88 and 95 numbers. The ANSI/ISA 88-based schemas are collectively called the Batch Markup Language, or BatchML. The ANSI/ISA 95-based schemas are collectively called the Business To Manufacturing Markup Language, or B2MML.

The suffix “ML” for Markup Language is a common practice in naming sets of XML schemas that define a vocabulary. BatchML is used to identify the schemas as targeted to the batch processing industry where knowledge of ANSI/ISA 88 is fairly ubiquitous. The Business To Manufacturing term was chosen to indicate the nature of the schema’s intended use, which is to integrate business systems with manufacturing systems. As with the ANSI/ISA 95 standards this is intended to reach beyond batch processing to the continuous and discrete industries.
SCHEMA ARCHITECTURE

As with the ANSI/ISA 88 and 95 standards, BatchML and B2MML share similarities but differ significantly in their architecture and design. In general BatchML provides for a high degree of flexibility in its use. Where the ANSI/ISA 88 standard models may be collapsed or expanded, individual elements defined in the schema (part of a model in ANSI/ISA 88) may be used independently as desired or high level elements such as master recipes may be used. B2MML offers less flexibility at the lower levels with the clear intention that primary use of the schemas will be for the primary ANSI/ISA 95 models, such as production schedule and production performance.

In both cases the BatchML and B2MML schemas may be used to derive new corporate, application or system specific schemas. While the derivation of new schemas may seem contradictory to the use of a standard it is a pragmatic recognition that companies, both operating and vendor, have requirements beyond the core requirements of the standards.

Both BatchML and B2MML schemas are implemented using the World Wide Web Consortium’s (W3C) XML Schema recommendations. The W3C schema language is identified by the file extension “xsd”. This schema format provides greater flexibility and additional features over the previous W3C format called DTD (Document Type Definitions) and over the Microsoft XML data reduced (xdr) format.

The working group used the following general rules in creating the schemas:

- Title case has been used (except in the case of common abbreviations such as ID)
- XML simple and complex types have been defined for all elements
- The “##any” element is widely used to permit the WBF elements to be extended with the insertion of elements from other (standard or proprietary) namespaces.

Explanation of these rules is beyond the scope of this paper, further information can be found in the BatchML and B2MML documentation, in the W3C XML Schema Part 0: Primer recommendation and in various books concerning XML schemas.

BATCHML

The Batch Markup Language (BatchML) defines XML elements for

- Master recipes,
- Control recipes,
- Recipe building blocks,
- Equipment elements,
- Batch list, and
- Enumeration sets.

The internal structures for master and control recipes and for equipment elements follows the models and terminology in the ANSI/ISA 88 standards and comply with the data model in clause 4 of Part 2. The batch list element corresponds to the batch schedule defined in Part 2. The term batch schedule was not used in this implementation to reduce confusion with the ANSI/ISA 95 production schedule which is
part of B2MML. The enumeration sets implement the enumeration sets defined in the relational tables in clause 5 of ANSI/ISA 88 Part 2. Since the SP88 committee is currently working on them, general and site recipes as well as batch history are not covered in BatchML.

A single element named BatchInformation is defined to serve as the root element in the schema and as a container for all of the elements listed above. Also every element that is part of one of the elements above is declared as a global element and is based upon a simple or complex type.

There are four BatchML schemas. The reason for this number is to provide the greatest flexibility in how they may be used. The four schemas and their intended uses are:

<table>
<thead>
<tr>
<th>Schema</th>
<th>Intended Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchML-Types</td>
<td>All BatchML elements are based upon a simple or complex type defined in this schema. This schema does not have a namespace so it may be included in its entirety by any other schema.</td>
</tr>
<tr>
<td>BatchML-TypesNamespace</td>
<td>A small schema that includes the BatchML-Types schema and applies a namespace. In effect this provides the same type definitions as BatchML-Types only in a namespace so schemas outside of the WBF namespace may import individual types.</td>
</tr>
<tr>
<td>BatchML-Elements</td>
<td>All BatchML elements are defined in this schema. All of the elements are based upon types defined in the BatchML-Types. This schema does not have a namespace so it may be included in its entirety by any other schema.</td>
</tr>
<tr>
<td>BatchML-ElementsNamespace</td>
<td>A small schema that includes the BatchML-Elements schema and applies a namespace. In effect this provides the same element definitions as BatchML-Elements only in a namespace so XML documents may reference the elements with a namespace prefix.</td>
</tr>
</tbody>
</table>

This architecture permits BatchML to be used in a very flexible manner. For example:

- Schemas defined outside of the WBF working group may include either the non-namespace element or type schema into a non-BatchML namespace
- Schemas defined outside of the WBF working group may import specific elements from the WBF namespace schemas
- Schemas defined outside of the WBF working group may derive new types based upon the BatchML types and in doing so either restrict or expand their content.
- All elements in BatchML may be used as the root element in an XML document
- XML documents may reference the BatchML namespace and use specific elements from the WBF schemas intermixed with elements from other schemas.

Most every complex type (e.g. Header, Parameter, ApprovalHistory,…) includes the element “##any”. This element is defined by the W3C XML schema recommendation to mean that any well-formed XML
element from any namespace may be included inside an element. In essence this means that an element like Header which is defined to contain:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModificationLog</td>
<td></td>
</tr>
<tr>
<td>ApprovalHistory</td>
<td></td>
</tr>
<tr>
<td>EffectiveDate</td>
<td></td>
</tr>
<tr>
<td>ExpirationDate</td>
<td></td>
</tr>
<tr>
<td>ProductName</td>
<td></td>
</tr>
<tr>
<td>BatchSize</td>
<td></td>
</tr>
<tr>
<td>ActualProductProduced</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>ProductID</td>
<td>##any</td>
</tr>
</tbody>
</table>

may also contain any additional elements as long as they are contained in a reference schema. Therefore an operating company which requires that their header contain the element “CustomerID” may define CustomerID in a corporate schema and use the element in the BatchML Header element.

This flexibility means that an XML document may consist solely of BatchML elements or may use BatchML elements are desired within a non-BatchML document and that a BatchML XML document may reference elements from non-BatchML schemas. This combination should enable the use of BatchML elements and types in the widest possible number of applications.

The high degree of flexibility was provided to meet the intent of the ANSI/ISA 88 standard to permit models to be expanded and collapsed.

**B2MML**

The Business To Manufacturing Markup Language (B2MML) schemas are organized to align with the ANSI/ISA 95 standard’s data models. The basis for each of the schemas including the mapping to the standard’s data models is listed below.

<table>
<thead>
<tr>
<th>B2MML Schema</th>
<th>Schema Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2MML-Common</td>
<td>All elements and types used in more than 1 other schema are defined here</td>
</tr>
<tr>
<td>B2MML-Personnel</td>
<td>ANSI/ISA 95 Data Model Personnel Model</td>
</tr>
<tr>
<td>B2MML- Equipment</td>
<td>ANSI/ISA 95 Data Model Equipment Model (except for the maintenance objects)</td>
</tr>
<tr>
<td>B2MML-Maintenance</td>
<td>ANSI/ISA 95 Data Model Equipment Model (maintenance objects only)</td>
</tr>
<tr>
<td>B2MML-Material</td>
<td>ANSI/ISA 95 Data Model Material Model</td>
</tr>
<tr>
<td>B2MML-ProcessCapability</td>
<td>ANSI/ISA 95 Data Model Production Capability Model</td>
</tr>
<tr>
<td>B2MML-ProcessSegment</td>
<td>ANSI/ISA 95 Data Model Process Segment Model</td>
</tr>
<tr>
<td>B2MML-ProductDefinition</td>
<td>ANSI/ISA 95 Data Model Product Definition Information Model</td>
</tr>
<tr>
<td>B2MML-ProductionSchedule</td>
<td>ANSI/ISA 95 Data Model Production Schedule Model</td>
</tr>
<tr>
<td>B2MML-ProductionPerformance</td>
<td>ANSI/ISA 95 Data Model Production Performance Model</td>
</tr>
</tbody>
</table>
With a separate schema for each model, with the exception of the equipment model which has two schemas, B2MML may be used in a modular fashion with an application only using the schemas required based upon the standard’s models used.

The internal structure of the schemas follows the ANSI/ISA 95 standard’s data model structures. The root element is named after each data model’s root element and each object in a data model is generally represented as an XML element. The standard’s technique of using application specific properties is also implemented in the schemas.

The equipment and maintenance schemas are exceptions in that two schemas have been created for one ANSI/ISA 95 data model. This was done to provide the flexibility of using equipment and maintenance objects separately.

All elements in the schemas are declared using simple and complex types. The common schema contains type definitions that are used in more than one other schema. Each of the other schemas includes the common schema and uses the types as needed. Any type that is used in only one schema is defined in that schema.

Where in BatchML all elements are declared globally, meaning they may be used in other schemas or XML documents, in B2MML only a few elements are declared globally. Generally the objects in the ANSI/ISA 95 standard that represent data to be exchanged between systems are implemented as global elements with the addition of a few container elements for the equipment, personnel and material models. The other objects, which are generally part of the exchanged objects, are defined as local elements. The global elements are:

- Equipment
- EquipmentCapabilityTestSpecification
- EquipmentClass
- EquipmentInformation
- MaintenanceInformation
- MaintenanceRequest
- MaintenanceResponse
- MaintenanceWorkOrder
- MaterialClass
- MaterialInformation
- MaterialLot
- MaterialSubLot
- Person
- PersonnelClass
- PersonnelInformation
- ProcessSegment
- ProcessSegmentInformation
- ProductInformation
- ProductionCapability
- ProductionPerformance
- ProductionRequest
- ProductionResponse
- ProductionSchedule
- ProductProductionRule
- QAMaterialTestSpecification
- QualificationTestSpecification

Most of the elements in the schemas are optional so that XML documents based upon them only need to contain the elements applicable to the application.

While the ANSI/ISA 95 standards are not intended to be expanded and collapsed to the same extent of the ANSI/ISA 88 standard’s models, the B2MML schemas have been designed with the “##any” element as part of most objects. This addition permits corporate, application or product specific data to be added to objects as needed.
CUSTOMIZATION

While the ANSI/ISA 88 and 95 standards provide a firm basis for many products and integration projects, neither standard can satisfy every product or projects requirements. For this reason the BatchML and to a lesser degree the B2MML schemas have been designed so that they may be extended and restricted as required for an application.

While the details on how this can be done are too involved for this paper an example of why this may be done is possible.

A common standard for batch processing companies is the format of their batch ID. This format will vary between companies; some embed the date, a year, or a sequential number, or even include product codes. Also vendor’s systems typically only permit a maximum number of characters in a batch ID and may restrict the characters that can be used, for example no spaces or special characters may be permitted. Any of these examples are valid according to the ANSI/ISA 88 standard. However; when data is to be exchanged between two systems these details can play havoc.

The BatchML common schema declares a simple type called BatchIDType which is an XML string. An XML string may be of any length and contain any Unicode character. This is about all a standard can define for a batch ID since the precise requirements for each application is a business rule.

Operating companies and vendors could each publish their requirements or support for batch ID by deriving a new type from the BatchML BatchIDType. An example of this is shown in the declaration of a new type called CorporateBatchIDType.

```xml
<xsd:simpleType name="CorporateBatchIDType">
  <xsd:restriction base="BatchIDType">
    <xsd:pattern value="B[0-9]{8}[A-Z]{3}"/>
    <xsd:length value="12"/>
  </xsd:restriction>
</xsd:simpleType>
```

This new type is derived from the BatchML BatchIDType so in some cases it could be used interchangeably with the BatchML type. The derivation in this case is used to place two restrictions on the contents of the string. The first restriction is to require a pattern where the first character must be the letter “B”, the next 8 characters must be numbers and then characters 10-12 must be letters A-Z. The second restriction limits the string length to 12 characters. This restriction prevents any other characters from following those in the pattern.

While this format is not obvious to newcomers it is based upon the common programming technique called “regular expressions” and provides for a wide amount of flexibility in limiting string content.

This technique could be used by an operating company to publish its batch ID format to its vendors or supply chain partners or by a vendor to publish a system’s capabilities. In either case if data is to be exchanged a programmer would require this knowledge of both the sending and receiving systems in order to successfully implement the interface.

Customization is not limited to string content, complex types (elements that contain other elements) may have elements added or removed from them, simple types (elements that are based upon a data type such as string or integer) may have limits placed upon their values or have defined lists of values stated.

These customization abilities are the basis for tailoring BatchML and B2MML for specific corporate and application uses.
INDUSTRY ACCEPTANCE

While both BatchML and B2MML are designed for the exchange of manufacturing data they will likely be used with significant differences.

The batch processing industry and marketplace is relatively mature with a great deal of installed base of vendor’s systems and operating company’s methods of operation. While ANSI/ISA 88 has been widely adopted in the U.S. and Europe there are still many situations where legacy concerns limit its use or where proprietary data, both vendor’s and operating company’s, require additional data be exchanged.

BatchML is designed to fit into this type of marketplace. The BatchML schemas may be used as published to exchange core sets of data, they may be customized by adding elements and constraining or removing elements to fit corporate, application or product specific requirements, or individual elements may be used in third party applications.

On the other hand the enterprise-control system market is still evolving. While there are large enterprise system vendors many operating companies are still developing their standards and evolving their approaches to integration. B2MML provides the first technology independent implementation of the ANSI/ISA 95 standard. The model based schemas in B2MML may be used as needed in specific applications or products. The schemas provide a framework for XML documents that can be processed by different systems with the property structure permitting application specific details to be incorporated within the framework. The B2MML schemas will likely see first use as part of custom integration projects where systems on both sides are customized to process the XML document contents.

Industry acceptance of BatchML and B2MML does not only depend upon the acceptance of the standards they are based upon. Also critical is the acceptance and use of XML in operating companies. Based upon the widespread use of B2B marketplaces and commerce sites and other XML based industry initiatives it appears that there is strong evidence of the use of XML by IT groups in operating companies. Some of the reasons for this is the use of XML in mainstream information technology. After all it is seldom that Microsoft, Oracle, IBM and Sun Microsystems all agree on something as they have all adopted XML and provide wide support for it among their products. Some of the reasons for this wide acceptance is XML’s independence from any one company, its ability to be used on many platforms (Unix, Linux, Windows, MAC) and languages (Visual Basic, C++, Java,…), its ability to cross firewalls, its ability to be used in program to program connections or as a file format, and its strong support for structured data.

The acceptance by IT groups should make the use of BatchML and B2MML easier to accept than previous process industry defacto standards such as the DCOM based interfaces from the OPC Foundation.

USAGE SCENARIOS

While it may seem that the logical uses of BatchML and B2MML are in developing product interfaces there are many other potential use scenarios, especially when considering operating companies.

**Internal Integration**

Automation and enterprise integration projects that are internally managed and require the coordination of multiple vendors could use BatchML and B2MML as the basis for data definitions.
While due to the wide variability of data in ANSI/ISA 88 based products BatchML will not offer plug and play data exchange it is possible to use the BatchML elements and type definitions as core requirements for various vendors. Data requirements beyond the core schemas can be handled by creating project specific extensions to the schemas to document vendor and company specific data.

The flexible framework of the ANSI/ISA 95 data models implemented in B2MML are ideal for integration projects where the data to be exchanged is highly application specific.

The use of BatchML and B2MML can be used to avoid recreating the data structures on for each system and application thereby reducing the cost of integration projects.

Specifications

An area that is currently underserved by standards is the publication of a company’s specifications for a project during bid solicitation, proposal evaluation and project execution. It is commonplace for vendors to receive different formats for items such as recipes, formula data, schedules, material data, and equipment data from operating companies during the bid process. Likewise during proposal evaluation operating companies invariably receive the same types of information in each vendor’s format. In all cases this data may well conform to the ANSI/ISA 88 or 95 standard, but it’s depiction and packaging may cause it to appear very different from one another. During project execution the exchange of large amounts of data can require time consuming manipulation and sometime re-entry, again this is a result of variances in how data is package and organized.

An example of this would be including an XML document containing equipment data, sample recipes and sample production schedules in an RFQ. This would provide a precise method to transmit requirements and sample data for a new project. Likewise during project execution operating companies could transmit formulation data and detailed equipment data in XML documents to be loaded into automation systems.

During proposal evaluation an operating company could more easily compare the data capabilities of different vendor’s systems by comparing their extensions and restrictions to BatchML and B2MML.

One solution may be to utilize BatchML and B2MML based XML documents as the format for exchanging this type of data. If the data is exchanged in XML documents then mainstream tools such as Office XP and mapping languages such as XML StyleSheet Transformations (XSLT) may be used to manipulate the data. If the core data represented by BatchML and B2MML are in a consistent format then the data will be easier to use.

Corporate Standards

Taking the use of the markup languages in specifications to a higher level, operating companies could create corporate standards for data items such as master recipes, schedules, production performance, and material information. These corporate standards could be used to create mappings to supply chain partners and preferred vendor’s systems. Corporate standards can be created by deriving new schemas from BatchML and B2MML. This will not only save time in their development but also make it easier for supply chain partners and vendors to map the data to their own information and automation systems.

Web Publishing

The ability to export data from systems in XML that complies with publish schemas will make the publication of data via the web, whether for internal uses or as part of supply chain collaboration, easier. There are many web technologies that support the display of XML data on web pages in table and list
formats. The advantage of using XML to provide data for web pages is that as the layout of the web page changes the data feed can remain the same.

**Office Automation**

The extraction of manufacturing data into common office automation products such as Microsoft Office is a common activity. In today’s environment accessing discrete values or series of values such as possible in CSV (Comma Separated Values) files limits the usefulness of the office automation products. As office automation products start to support XML data then more structured data such as B2MML’s production schedule and performance elements and BatchML’s master recipe and equipment elements can be imported directly to office automation products for manipulation. While this ability is not fully developed yet as corporate standards are developed it will become possible to develop personal database, spreadsheet and word processing templates based upon them.

**BENEFITS**

The use of BatchML and B2MML as the basis of integration projects in operating companies should be able to provide real savings in life cycle costs when compared to existing alternatives.

In the early stages of a project BatchML and B2MML XML documents can be used to exchange sample data and documentation of existing data to benefit both operating companies and vendors. During project execution XML documents can be used to exchange configuration data that can be mapped into corporate standard and system specific formats. This use of XML can reduce the manpower costs in the front end of projects.

When automation and information systems must exchange manufacturing data, designing the data exchanges using BatchML and B2MML can provide savings in avoiding costly custom designs and their maintenance and permit reuse of software between projects.

B2MML is unique in that it is the first published implementation ready format for using the ANSI/ISA 95 standard. If companies had considered using the standard on projects prior to this they would have had to bear the cost of developing a set of proprietary data structures to enable its use. With B2MML the published schemas may be used as the basis for using the standard, thereby reducing costs associated with using the standard and increasing the ability to interoperate between companies and vendor’s systems.

**CONCLUSION**

The Batch and Business to Manufacturing Markup Languages are XML based implementations of two standards that are very critical to the batch processing industry, ANSI/ISA 88 and 95. These industry markup languages will enable the use of mainstream information technology in automation and information integration projects. While there will not be an immediate step change when the markup languages are released there will be a clear opportunity for operating companies to benefit from their use.
One note of caution, this paper was written in early January, 2002 to meet the deadlines for this conference, at that time the working group was preparing the first release candidates for BatchML and B2MML. While this paper reflects the status of the schemas at that time by the time the paper is published details may have changed. The latest schemas and documentation should be referenced to understand their current status.

The author would like to thank all the members of the WBF XML working group and especially those that have attended meetings, participated on-line with the list server, or privately between members for their contributions to the schemas and the topics discussed in this paper. The views and opinions expressed in this paper are the author’s interpretation of the working group’s intentions and work to date and not necessarily those of the working group.

REFERENCES


