

How and Where XML is Changing the Markets

Anthony B. Coates <tony.coates@reuters.com>

Abstract

XML has taken root in the financial world, but only the first pieces of the puzzle are in place. This paper presents an overview of the available financial XML specifications, what their scopes are, and how they relate to each other in practice. Reuters is the world's largest supplier of financial data, so this is a practical discussion of what is and is not possible at present, and what is in the pipeline going forward. XBRL, FpML, IRML, ISO15022, and others are covered.

1. Introduction

eXtensible Markup Language (XML) is making clear inroads in engaging the financial world, but the early progress on financial XML specifications [1] [2] has been in particular vertical areas, and no broad horizontal solution for encoding financial information is yet in place. This presentation will examine the available XML specifications for finance, and how they fit into the bigger picture of financial information usage.

2. XML Specifications for Finance

The XML specifications covered in this presentation will be broadly classified as *view* (informational), *do* (transactional), or both.

[1] While one often talks colloquially of XML *standards*, there is only a handful of international bodies which can legally issue a standard. In this presentation, the term *specification* will be used to indicate an XML schema that is an *industry standard* or a *de facto standard*, rather than a legally binding standard.

[2] DTDs, XML Schemas, etc. will be collectively referred to in this presentation as *schemata* (i.e. all in lower case - one schema, two schemata, etc.).

Specifications to be discussed

- Financial Products Markup Language ([FpML](#)) [do]
- eXtensible Business Reporting Language ([XBRL](#)) [view]
- Market Data Definition Language ([MDDL](#)) [view]
- ISO 15022 [view, do]
- swiftML [view, do]
- Financial Information Exchange Markup Language ([FIXML](#)) [do]
- Investment Research Markup Language ([IRML](#)) [view]
- Research Information eXchange Markup Language ([RIXML](#)) [view]
- FinXML [do]
- NewsML [view]
- MarketsML [view, do]
- ebXML [do]

3. [FpML](#)

[FpML](#), the *Financial Products Markup Language*, is a set of financial specifications which is initially focussing on transactions of *over-the-counter financial instruments*. These are financial products which are not traded via an exchange, but directly between two financial institutions, such as banks.

Traditionally, over-the-counter deals are agreed by telephone, where only the most important details are discussed. After this, each party faxes and couriers the full details to the other, and the details are then compared to make sure that both parties were actually agreeing to the same thing. Time pressures mean that the full details cannot be agreed during the telephone phase of the negotiations. As such, the process of *confirmation*, comparing what both parties believed they had agreed

to, is manually tedious, and this makes it a good candidate for automation using XML. FpML messages will each have a known and predefined set of defaults, each of which can then be overridden explicitly by either party as required. This makes the process of *not specifying something explicitly* a well-defined one, and allows the confirmation to be done in a fast semi-automatic fashion where only mismatched information is brought to the attention of humans.

FpML currently covers *interest rate swaps* and *forward rate agreements*, and there are proposals to extend this to *equity derivatives*, *Foreign eXchange (FX) spots*, *FX forwards*, *FX swaps*, *non-deliverable forwards (NDF)*, *simple FX options* and *FX option strategies*. Transactions which are currently exchange-based, such as "sell 1000 Reuters shares on the London Stock Exchange (LSE) for no less than £20 per share", are out of scope at present.

The best way to make progress in defining XML specifications is to separate the issue of how you structure your documents from the issue of the names you use for different quantities or entities. In FpML, there is a separate architecture document that describes the rules for creating FpML DTDs and XML Schemas [3]. This allows the various FpML product DTDs and Schemas to be created with a consistent *look and feel*, without time wasted on fruitless discussions about elements vs. attributes, etc. FpML is interesting in that *elements are used in preference to attributes*. The only attributes in FpML DTDs and Schemas are a few special FpML-specific attributes. In FpML 1.0, only DTDs are formally supported, but both DTDs and XML Schemas will be formally supported in FpML 2.0 if XML Schema becomes a W3C Recommendation in time [4].

At the time of writing, FpML 1.0 was close to release, while work on FpML 2.0 is already well under way, in parallel with the finalisation of the 1.0 specification.

3.1. Sample FpML Document

[3] If you are interested in XML architecture, the author highly recommends that you read the FpML 1.0 Architecture Trial Recommendation. It is *not*FpML-specific, and is a good example of how to separate architecture issues from concrete product issues.

[4] How are both DTDs and XML Schemas supported, when there are small but potentially significant differences between the two? For FPML 2.0, the plan is to require that FPML 2.0 documents satisfy *both* the DTD and matching Schema, so that either can be used for validation. This is a transitional strategy, as support DTDs is deprecated, and will be dropped from a future version of FPML.

This sample is taken from the [FpML 1.0 Trial Recommendation \(2001-02-27\)](#).

```
<FpML version = "1-0"
  businessCenterSchemeDefault =
    "http://www.fpml.org/spec/2000/business-center-1-0"
  businessDayConventionSchemeDefault =
"http://www.fpml.org/spec/2000/business-day-convention-1-0"
  currencySchemeDefault =
    "http://www.fpml.org/ext/iso4217"
  dateRelativeToSchemeDefault =
    "http://www.fpml.org/spec/2000/date-relative-to-1-0"
  dayCountFractionSchemeDefault =
    "http://www.fpml.org/spec/2000/day-count-fraction-1-0"
  dayTypeSchemeDefault =
    "http://www.fpml.org/spec/2000/day-type-1-0"
  floatingRateIndexSchemeDefault =
    "http://www.fpml.org/ext/isda-1991-definitions"
  partyIdSchemeDefault =
    "http://www.fpml.org/ext/iso9362"
  payRelativeToSchemeDefault =
    "http://www.fpml.org/spec/2000/pay-relative-to-1-0"
  periodSchemeDefault =
    "http://www.fpml.org/spec/2000/period-1-0"
  resetRelativeToSchemeDefault =
    "http://www.fpml.org/spec/2000/reset-relative-to-1-0"
  rollConventionSchemeDefault =
    "http://www.fpml.org/spec/2000/roll-convention-1-0">
<trade>
  <tradeHeader>
    <partyTradeIdentifier>
      <partyReference href = "#CHASE" />
      <tradeId tradeIdScheme =
"http://www.chase.com/swaps/trade-id">TW9235</tradeId>
    </partyTradeIdentifier>
    <partyTradeIdentifier>
      <partyReference href = "#BARCLAYS" />
      <tradeId tradeIdScheme =
"http://www.barclays.com/swaps/trade-id">SW2000</tradeId>
    </partyTradeIdentifier>
    <tradeDate>1994-12-12</tradeDate>
  </tradeHeader>
  <product>
    <swap>
      <!-- Chase pays the floating rate every 6 months, based
on
      6M DEM-LIBOR-BBA, on an ACT/360 basis -->
      <swapStream>
        <payerPartyReference href = "#CHASE" />
        <receiverPartyReference href = "#BARCLAYS" />
        <calculationPeriodDates id =
"floatingCalcPeriodDates">
          <effectiveDate>
            <unadjustedDate>1994-12-14</unadjustedDate>
```

```

        <dateAdjustments>
<businessDayConvention>NONE</businessDayConvention>
        </dateAdjustments>
    </effectiveDate>
    <terminationDate>
        <unadjustedDate>1999-12-14</unadjustedDate>
        <dateAdjustments>
<businessDayConvention>MODFOLLOWING</businessDayConvention>
        <businessCenters id =
"primaryBusinessCenters">
            <businessCenter>DEFR</businessCenter>
        </businessCenters>
    </dateAdjustments>
    </terminationDate>
    <calculationPeriodDatesAdjustments>
<businessDayConvention>MODFOLLOWING</businessDayConvention>
        <businessCentersReference href =
"#primaryBusinessCenters" />
    </calculationPeriodDatesAdjustments>
    <calculationPeriodFrequency>
        <periodMultiplier>6</periodMultiplier>
        <period>M</period>
        <rollConvention>14</rollConvention>
    </calculationPeriodFrequency>
    </calculationPeriodDates>
    <paymentDates>
        <calculationPeriodDatesReference href =
"#floatingCalcPeriodDates" />
    </paymentFrequency>
        <periodMultiplier>6</periodMultiplier>
        <period>M</period>
    </paymentFrequency>
<payRelativeTo>CalculationPeriodEndDate</payRelativeTo>
    <paymentDatesAdjustments>
<businessDayConvention>MODFOLLOWING</businessDayConvention>
        <businessCentersReference href =
"#primaryBusinessCenters" />
    </paymentDatesAdjustments>
    </paymentDates>
    <resetDates id = "resetDates">
        <calculationPeriodDatesReference href =
"#floatingCalcPeriodDates" />
<resetRelativeTo>CalculationPeriodStartDate</resetRelativeTo>
        <fixingDates>
            <periodMultiplier>-2</periodMultiplier>
            <period>D</period>
            <dayType>Business</dayType>
<businessDayConvention>NONE</businessDayConvention>
        <businessCenters>
            <businessCenter>GBLO</businessCenter>
        </businessCenters>
        <dateRelativeTo href =
"#resetDates">ResetDate</dateRelativeTo>
    </fixingDates>

```

```

        <resetFrequency>
            <periodMultiplier>6</periodMultiplier>
            <period>M</period>
        </resetFrequency>
        <resetDatesAdjustments>
<businessDayConvention>MODFOLLOWING</businessDayConvention>
            <businessCentersReference href =
"#primaryBusinessCenters" />
        </resetDatesAdjustments>
    </resetDates>
    <calculationPeriodAmount>
        <calculation>
            <notionalSchedule>
                <notionalStepSchedule>
                    <initialValue>50000000.00</initialValue>
                    <currency>DEM</currency>
                </notionalStepSchedule>
            </notionalSchedule>
            <floatingRateCalculation>
<floatingRateIndex>DEM-LIBOR-BBA</floatingRateIndex>
                <indexTenor>
                    <periodMultiplier>6</periodMultiplier>
                    <period>M</period>
                </indexTenor>
            </floatingRateCalculation>
            <dayCountFraction>ACT/360</dayCountFraction>
        </calculation>
    </calculationPeriodAmount>
</swapStream>
    <!-- Barclays pays the 6% fixed rate every year on a
30E/360 basis -->
    <swapStream>
        <payerPartyReference href = "#BARCLAYS" />
        <receiverPartyReference href = "#CHASE" />
        <calculationPeriodDates id =
"#fixedCalcPeriodDates">
            <effectiveDate>
                <unadjustedDate>1994-12-14</unadjustedDate>
            <dateAdjustments>
<businessDayConvention>NONE</businessDayConvention>
            </dateAdjustments>
        </effectiveDate>
        <terminationDate>
            <unadjustedDate>1999-12-14</unadjustedDate>
            <dateAdjustments>
<businessDayConvention>MODFOLLOWING</businessDayConvention>
                <businessCentersReference href =
"#primaryBusinessCenters" />
            </dateAdjustments>
        </terminationDate>
        <calculationPeriodDatesAdjustments>
<businessDayConvention>MODFOLLOWING</businessDayConvention>
            <businessCentersReference href =
"#primaryBusinessCenters" />

```

```

        </calculationPeriodDatesAdjustments>
        <calculationPeriodFrequency>
            <periodMultiplier>1</periodMultiplier>
            <period>Y</period>
            <rollConvention>14</rollConvention>
        </calculationPeriodFrequency>
    </calculationPeriodDates>
    <paymentDates>
        <calculationPeriodDatesReference href =
"#fixedCalcPeriodDates" />
        <paymentFrequency>
            <periodMultiplier>1</periodMultiplier>
            <period>Y</period>
        </paymentFrequency>
    <payRelativeTo>CalculationPeriodEndDate</payRelativeTo>
    <paymentDatesAdjustments>
    <businessDayConvention>MODFOLLOWING</businessDayConvention>
        <businessCentersReference
href="#primaryBusinessCenters" />
    </paymentDatesAdjustments>
    </paymentDates>
    <calculationPeriodAmount>
        <calculation>
            <notionalSchedule>
                <notionalStepSchedule>
                    <initialValue>50000000.00</initialValue>
                    <currency>DEM</currency>
                </notionalStepSchedule>
            </notionalSchedule>
            <fixedRateSchedule>
                <initialValue>0.06</initialValue>
            </fixedRateSchedule>
            <dayCountFraction>30E/360</dayCountFraction>
        </calculation>
    </calculationPeriodAmount>
    </swapStream>
</swap>
</product>
<party id = "CHASE">
    <partyId>CHASUS33</partyId>
</party>
<party id = "BARCLAYS">
    <partyId>BARCGB2L</partyId>
</party>
</trade>
</FpML>

```

4. XBRL

XBRL, the *eXtensible Business Reporting Language*, is a financial specification

which is initially focussing on end-of-year company filings and reports. On an international level, the major complexity with company reports at present is that each country has its own accounting standard, each requiring a different usage of XBRL (though this will change when International Accounting Standards (IAS) becomes widely used). In the United States of America (USA), the United States of America (US) Generally Accepted Accounting Principles (GAAP) (*Generally Accepted Accounting Principles*) is used, while in the United Kingdom (UK) it is the UK GAAP, Australia has an Australian GAAP, etc. What differs in each case are:

- the list of defined accounting items;
- the rules on how lower level items are added/subtracted/multiplied to give higher level items.

XBRL allows companies to add their own items by extending their local XBRL *taxonomy*. This allows those items which are important to a company understanding its own business to be directly related to the standard accounting terms required in its annual filing and report.

XBRL has strong support from international accounting bodies and firms. Like FpML, XBRL separates architecture from product definition. However, the XBRL architecture is the antithesis of FpML's. XBRL is designed so that it can be *easily embedded* into general XML document formats, and so defines only 2 elements for general usage: `<item>` for accounting items, and `<group>` for grouping accounting items with common attributes. It is the *type* attribute of an `<item>` which identifies what kind an item is (examples taken from the XBRL 0.9 Specification [2000-07-31]):

```
<item  
type="ci:shortTermInvestments.marketableSecurities">2</item>
```

Note the period "." in the *type* value. XBRL uses namespaced hierarchical attribute values instead of the more usual XML paradigm of hierarchical element structures. The period in the *type* is an extra namespacing mechanism, to reduce the possibility of naming clashes. XBRL *taxonomies* have a format very similar to that of XML Schemas, the *important difference* being that the `<element>` tag in an XBRL taxonomy defines a hierarchical *type* value, not an element name:

```

<schema
  xmlns:xbrl="http://www.xbrl.org/core/2000-07-31/metamodel"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:ci="http://www.xbrl.org/us/2000-07-31/us-gaap-ci-2000-07-31"
  targetNamespace="http://www.fabulous.com/brandEquity">
  <import
    namespace='http://www.xbrl.org/core/2000-07-31/metamodel'
    schemaLocation='http://www.xbrl.org/core/2000-07-31/xbrl-meta-2000-07-31.'
    <element name="intangibleAssetsGross.brandEquity"
      type="xbrl:monetary">
      <annotation>
        <appinfo>
          <xbrl:rollup
            to="ci:intangibleAssetsNet.intangibleAssetsGross" weight="1"
            order="4.5"/> <xbrl:label xml:lang="en">Brand
            Equity</xbrl:label>
          </appinfo> </annotation> </element> </schema>

```

Note the `<xbrl:rollup>` tag, used to define the weighting with which a child item should be added to its parent item (the parent is defined by the value of the `to` attribute). The `order` attribute is a hint on the display order in which items with the same parent should be presented. Multiple `<xbrl:label>` tags can be used to add multilingual display headings for each item.

The fact that [XBRL](#) has effectively repurposed the core of [XML Schema](#), using it to structure attribute values rather than elements, is a likely source of confusion for those already familiar with [XML Schemas](#). It will lead to the creation of a parallel set of [XBRL-specific](#) tools, which do with [XBRL taxonomies](#) and [XBRL documents](#) what standard tools do with [XML Schemas](#) and instances of those Schemas. In many cases, such deviance from the [XML norms](#) could be expected to severely limit the acceptance of an [XML specification](#), but [XBRL](#) has sufficiently strong support from the accounting profession that it appears certain to succeed in spite of its idiosyncrasies.

The [US GAAP](#) (Commercial & Industrial) was the first completed [XBRL taxonomy](#). It contains thousands of items, and at the time of writing only one [US corporation](#) has published its company report using [XBRL](#). The [UK GAAP](#) is taking a different approach, and will be limited initially to hundreds of items. It is hoped that this will significantly reduce the learning curve, and thus speed the uptake of [XBRL](#) in the [UK](#). Reuters is planning to publish its 2001 company report in [XBRL \(UK GAAP\)](#) to demonstrate its support. Note that Germany is also quite advanced in its

development of an XBRL taxonomy. A draft [IAS GAAP](#) taxonomy has been published, and this is of particular importance since European Union ([EU](#)) corporations will be required to submit company reports in [IAS](#) form from 2005. The [IAS GAAP](#) seems the best placed to become a global standard taxonomy for company reports in [XBRL](#).

5. MDDL

Market Data Definition Language ([MDDL](#)) is a financial information specification being produced by the Financial Information Services Division ([FISD](#)), part of the Software and Information Industry Association ([SIIA](#)). To quote from the [MDDL](#) mission statement:

The mission of the activity is to define a publicly available standard that provides a generic XML-based interchange format on the fields needed to describe financial instruments (including identifiers and current and historical values), corporate events (including specific corporate and instrument information affecting value and tradability), and market-related information (including economic and industrial indicators). The goal is to promote data interoperability.

[MDDL](#) has only just started its work (so no details are available at the time of writing), and is initially focussing on end-of-day and snap information for financial instruments (share prices, etc.). Some of the corporations involved have donated their early [DTDs](#) for financial information as inputs, but the final specification will not necessarily resemble any of these. The [MDDL](#) work has been split into technical (architectural) and vocabulary threads, following a similar pattern to [FpML](#) and [XBRL](#).

6. ISO 15022

A major non-[XML](#) specification which will influence [MDDL](#) and other financial [XML](#) specifications is ISO 15022. This provides a standard set of (>10k) data fields for financial information and (~100) messages for financial transactions. The data dictionary and catalogue of messages are maintained on ISO's behalf by Society for Worldwide Interbank Financial Telecommunication ([SWIFT](#)), a banking industry co-operative. An effort has been started to define a direct [XML](#) version of ISO 15022, one which supports not only ISO 15022 but also the existing (non-[XML](#)) [SWIFT](#) and

Financial Information Exchange ([FIX](#)) transaction protocols. However, this work has not yet been made public, and so it is difficult to speculate whether ISO 15022 will influence financial [XML](#) usage more via its direct [XML](#) incarnation or via its integration into other [XML](#) specifications.

7. swiftML

[SWIFT](#) is an existing electronic messaging system used by major banks. The messages are being converted to [XML](#) under the name "swiftML". [SWIFT](#) are closely involved with ISO 15022, on which the latest set of [SWIFT](#) messages is based, so swiftML is also closely aligned to ISO 15022.

Architecturally, swiftML [DTDs](#) ([XML](#) Schemas eventually) are generated from UML models via a swiftML-specific set of mapping rules. The generated [DTDs](#) contain many fixed attributes with opaque ID codes of various kinds. It is not clear whether the explicit inclusion of these IDs will be continued when swiftML eventually moves to [XML](#) Schemas. The author suggests that assignment of fixed IDs to elements would probably be better done using Schema Adjunct Framework ([SAF](#)), the *Schema Adjunct Framework*, when this is standardised in future.

7.1. swiftML Sample

```
<Payment elementID="ABCD0003" type="ABCD0003">
  <CreditAcct elementID="ABCD0001" roleID="ABCD0005"
type="ABCD0004">
  <Balance elementID="ABCD0002" type="float">1000</Balance>
  <AcctID elementID="ABCD0008"
type="string">124-56789-1</AcctID> </CreditAcct>
  <DebitAcct elementID="ABCD0001" roleID="ABCD0006"
type="ABCD0004">
  <Balance elementID="ABCD0002" type="float">1550</Balance>
  <AcctID elementID="ABCD0008"
type="string">125-56789-1</AcctID>
</DebitAcct>
</Payment>
```

8. FIXML

[FIX](#) is a non-[XML](#) financial transaction protocol which aims to be vendor-neutral. The

consortium is composed of a group of banking and financial institutions who view themselves as clients rather than vendors. [FIXML](#) has been announced as the [XML](#)-isation of the existing [FIX](#) protocol (messages), though no details are available at the time of writing. The intention is that, for an interim period, both traditional [FIX](#) messages and [FIXML](#) will be supported in parallel until [FIX](#) is eventually deprecated in favour of [FIXML](#).

9. [IRML](#)

[IRML](#), the *Investment Research Markup Language*, is an effort to develop a specification for tabular investment research data, such as that supplied in *morning notes*, or in the front matter of full investment reports. Standardisation of this data will make it possible generate automated comparisons and summarisations of investment research from multiple firms, which will impact the way investment research is used in future. Where currently individual users of investment research are likely to read only a small number of reports from selected analysts, [IRML](#) should make it worthwhile to take information from a wider range of analysts and combine them with a user's preferred weightings to get a broader and more balanced view.

[IRML](#) is at an early stage in its development, and is currently being reorganised to better separate architectural issues from product issues, so no samples are available at the time of writing.

10. [RIXML](#)

[RIXML](#) is another investment research specification, but one which focusses on metadata rather than on the way that reports are structured. Some press reports have suggested that [RIXML](#) and [IRML](#) are direct competitors, one of which must wither, but this is not the case. [RIXML](#) is being developed by a closed group of companies who produce or consume investment research, and some of those companies are also involved in [IRML](#), so the two efforts are most likely to remain complementary rather than competing.

Early drafts of [RIXML](#) are not being made public, so no samples are available at the time of writing.

11. FinXML

FinXML is mentioned here for completeness. It is an [XML](#) specification for financial transactions from Integral, but is not open (and indeed is *patent pending*). Their licensing model has recently been loosened. Previously, the author's reading of the license agreement was that you had to agree to not to work on or with any competing specification before you could even look at the FinXML DTD or related materials. Now they can be downloaded for a trial period, though you must delete all materials after the expiry date (including all FinXML instance documents that you have created) if you decide not to join the "FinXML.org" consortium.

12. NewsML

NewsML is a multimedia news packaging and distribution format from the International Press Telecommunications Council ([IPTC](#)). While it is not specific to finance, it is important to remember that news is as important as statistics in shaping the decisions of brokers on what to buy and sell. So, when considering [XML](#) formats for financial information, NewsML is important as the [XML](#) way to provide the news side of the equation. While NewsML provides rich facilities for packaging news and adding metadata, it does not define the formats of the actual content. Textual content in NewsML are most likely to be done using eXtensible HyperText Markup Language ([XHTML](#)) or News Industry Text Format ([NITF](#)). Reuters will be using [XHTML](#) (automatically converted to [NITF](#) where required), while some [US](#)-based news providers will be using [NITF](#) directly, as this is the preferred format of some [US](#) newspapers. When Reuters moves its full news production to a NewsML architecture at the end of 2001, it will more than likely become the world's largest publisher of [XHTML](#).

13. MarketsML

Reuters has announced that it will be producing an [XML](#) specification for financial information and transactions, named "MarketsML". Reuters is the largest financial information supplier in the world, and the intention with MarketsML is to build a comprehensive set of [XML](#) Schemas which cover the full range of financial

information and messages which Reuters deals with, both now and into the future. MarketsML will interoperate with NewsML to allow complex linking between news and financial statistics. Further details have not been released at the time of writing, but check the latest version of this talk for more information (see [Section 17.](#)).

14. ebXML

While ebXML is a general e-business [XML](#) specification, and not specific to the financial industry, it would be wrong to leave it unmentioned in this context. The *do* half of financial [XML](#) is about financial transactions, while part of ebXML's scope is general business transactions. Due to the sums of money and the risks involved, financial transactions typically have more demanding requirements for speed, validation, authentication, and security than do general business transactions. However, once the global ebXML infrastructure is in place, it would be very surprising if that infrastructure were not suitable for at least some financial transactions, and the migration could well continue from there if the needs of the financial world drive the performance and functionality of ebXML implementations. So, expect to see ebXML take a not insignificant role in financial transactions in the future, though be aware that it is too early in ebXML's life to say where the first uses of ebXML for finance might occur.

15. Some Notes on XML Validation

Mention validation in an XML context, and people's minds usually jump to thoughts of Document Type Definition ([DTD](#))s or XML Schemas. Some people may also connect validation to Schematron, REgular LAnguage description for XML ([RELAX](#)), or Tree Regular Expressions for XML ([TREX](#))[2]. However, each of these choices is simply one kind of validation, and rarely does any of them provide everything that is needed to genuinely validate a document. It can be enormously difficult to design an [XML](#) specification so that all validation can be done using a single validation tool, yet there are no tools (of which the author is aware) which make it easy to use multiple validation strategies on a single [XML](#) document (or parts thereof) to get the desired effect.

What does *validation* even mean? People who write validation tools think of it as checking whether you have the allowed set of tags in the correct structure, or maybe

the allowed datatypes, or perhaps the required set of relationships between certain elements, attributes, or content. For someone using a particular XML specification, validation is whatever you need to do to make sure that a particular XML document *makes sense* semantically. Due to the limitations of existing validation tools, some schema-specific validation is typically embedded into applications that process XML documents, but a better architectural principle is to build a separate schema-specific validator that performs those checks which are not done by the standard tools. This is cleaner and easier to maintain than having validation code spread throughout an application.

An unfortunate trend in numerous XML specifications is to use attributes in preference to elements for typing (*what kind of*) information. This is seen as making the schemata easier to manage, because new types can be added without changing the schema. This simply pushes the versioning problem off into some other document (converting the problem into a meta-problem [5]), and that other document's format is commonly not well specified, understood, nor supported. Unless a special tool has been developed to validate the attribute values, this approach simply converts the XML architect's problem into a problem for each and every application developer, which makes it a poor strategy in the bigger picture.

It is important to be aware of these issues in understanding what the architectural differences between the different financial schemata are, and what the consequences of those differences will be in implementing applications which process XML documents which conform to those schemata.

16. Some Notes on Starting Your Own Multivendor Consortium

Most of the XML specifications mentioned in this presentation have been created by multivendor consortia. Should you be interested in creating your own financial XML specification, and think that you will need a consortium to match, be aware that most new consortia seem to take from 6 months to 1 year to put in place their management and operating structures, before any technical work seriously begins.

[5] As eloquently pointed out by C. Michael Sperberg-McQueen in his closing to the *Extreme Markup Languages 2000* conference.

This is a very serious delay in an XML world driven by *Internet time*. If time is of the essence, the fastest way to approach development of a new public specification is to work via an existing financial consortium, or otherwise via Organization for the Advancement of Structured Information Standards (OASIS), which can provide a prefabricated committee process and infrastructure to get you up and running.

17. Conclusion

The financial area is a large and complicated one, and XML is still only a recent technology. The people who understand financial information and transactions the best are not the ones who understand XML the best, so it is only to be expected that much of the XML work in finance is still at the level of bringing the right people together. FpML and XBRL particularly have made good progress in creating credible and open financial XML specifications, but the architectural mismatch between these two may cause problems in the future, should they ever start to cover the same areas of the financial landscape. There is not yet an XML specification that covers the bulk of financial information requirements, nor an XML specification which covers the bulk of financial transaction requirements. However, the financial community is clearly driving towards these, while at the same time trying not to over-divide the available development resources by pursuing simultaneously too many holy grails.

Vertical approaches to creating financial specifications are a good way to prove the viability of XML-based solutions to the business managers whose support is needed if XML is to cover financial needs comprehensively. It remains to be seen how a consortium with experience in a particular vertical area will be able to expand its membership and/or skills base in order to extend the scope of its specification(s). There remains a lot of committee work to be done between where we are now and where we want to be. An *important point* that now has been proven by experience is that separating the XML structural decisions from the product/vocabulary decisions is a very good way to allow the people who know XML to do XML, and the people who are domain experts to worry about domain issues rather than angle-brackets.

It is against this background that Reuters is developing its MarketsML family of XML Schemas, with the aim of uniting the representation of financial information via a consistent architectural approach. MarketsML will interoperate with other XML specifications via transformation, and has the express aim of being able to fully

represent the data models of major financial XML specifications. As the XML Schema Working Group (WG) has been trying to get across of late, it is the data model (Information Set) that is important, not the particular element names nor the particular attribute names.

The latest version of this presentation will be available at <http://about.reuters.com/researchandstandards/events/2001/05/xml-europe/>. If you find that this URL is not in place when you check it, the author can be contacted directly by e-mail as tony.coates@reuters.com.

Bibliography

[ebXML] ebXML - <http://www.ebxml.org/>

[FpML] FpML - Financial Products Markup Language, <http://www.fpml.org/>

[FinXML] FinXML - <http://www.finxml.org/>

[FIXML] FIXML - <http://www.FIXprotocol.org/>

[IRML] IRML - Investment Research Markup Language, <http://www.irml.org/>

[ISO15022] ISO 15022 - <http://www.iso15022.org/>,
http://groups.yahoo.com/group/XML_Init_Main

[MDDL] MDDL - Market Data Definition Language,
http://www.fisd.net/reports/xml_default.html

[NewsML] NewsML - <http://www.iptc.org/>, <http://newsshowcase.reuters.com/>

[NITF] NITF - News Industry Text Format, <http://www.nitf.org/>

[OASIS] OASIS - Organization for the Advancement of Structured Information Standards, <http://www.oasis-open.org/>

[RELAX] RELAX - REgular LAnguage description for XML, <http://www.xml.gr.jp/relax/>

[RIXML] RIXML - Research Information eXchange Markup Language,
<http://www.rixml.org/>

[SAF] SAF - Schema Adjunct Framework,
http://www.extensibility.com/tibco/resources/saf_dec2000.htm

[Schematron] Schematron -
<http://www.ascc.net/xml/resource/schematron/schematron.html>

[swiftML] swiftML - http://www.swift.com/index.cfm?item_id=2642

[TREG] TREG - Tree Regular Expressions for XML,
<http://www.thaiopensource.com/trex/>, <http://www.oasis-open.org/committees/trex/index.shtml>

[XBRL] XBRL - eXtensible Business Reporting Language, <http://www.xbrl.org/>

[XHTML] XHTML - eXtensible HyperText Markup Language,
<http://www.w3.org/MarkUp/>

Glossary

DTD	Document Type Definition
EU	European Union
FISD	Financial Information Services Division
FIX	Financial Information Exchange
FIXML	Financial Information Exchange Markup Language
FpML	Financial Products Markup Language
FX	Foreign eXchange
GAAP	Generally Accepted Accounting Principles
IAS	International Accounting Standards
IPTC	International Press Telecommunications Council
IRML	Investment Research Markup Language

LSE	London Stock Exchange
MDDL	Market Data Definition Language
NITF	News Industry Text Format
OASIS	Organization for the Advancement of Structured Information Standards
RELAX	REgular LAnguage description for XML
RIXML	Research Information eXchange Markup Language
SAF	Schema Adjunct Framework
SIIA	Software and Information Industry Association
SWIFT	Society for Worldwide Interbank Financial Telecommunication
TREX	Tree Regular Expressions for XML
UK	United Kingdom
US	United States of America
USA	United States of America
WG	Working Group
XBRL	eXtensible Business Reporting Language
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

Biography

Anthony B. **Coates**

Leader of XML Architecture & Design

Reuters
Chief Technology Office
London
United Kingdom
Email: tony.coates@reuters.com

Anthony B. Coates - Anthony B. Coates is Leader of XML Architecture & Design in Reuters Chief Technology Office, and represents Reuters on technical working groups for FpML (Financial Products Markup Language), Market Data Definition Language (MDDL), and IRML (Investment Research Markup Language). He is an architect of Reuters' MarketsML specification. As Leader of XML Architecture & Design, Tony recommends XML technologies and design strategies to product groups within Reuters to co-ordinate the business usage of XML and leverage the best available solutions. This includes both general XML technologies and specific vertical market XML languages. His background includes developing software for technical analysis & financial graphics, developing multimedia & Web applications, and theoretical & experimental physics. He has worked with XML since 1998. Prior to joining Reuters CTO, Tony worked as a software engineer for Equity Systems Pty Ltd, where he designed and implemented an XML architecture for financial graphics solutions.