UBL Planning Subcommittee report

28 October 2001

Summary

The effort to define UBL is marked by the following characteristics (as noted in the aims adopted on 2 August 2001): UBL will be a **synthesis** of existing business document libraries, with a starting point of **xCBL 3.0**. Existing **standards** work, in particular ebXML and Joint Core Components, will be important guideposts for harmonization. Finally, UBL must be legally **unencumbered**.

This report describes the UBL Planning Subcommittee's recommendations for specific deliverables and work breakdowns in order to achieve these aims. Open issues are marked with "**Issue**".

Deliverables

Recommended Deliverables

We recommend that the UBL TC produce the following deliverables in Phase 1. Development dependencies are noted.

#	Name	Dependencies	Comments
1	**Schema design rules and naming conventions**	None (but the decision about an extension vs. restriction design principle must have been made)	There should be some preliminary guidelines developed at the beginning of schema work, but it should also be updated and fleshed out in conjunction with schema development. xCBL does have something like this available for its current state.
2	**Core schemas for business document types**	#1 must be started for this to be started, but need not be finished for this to be finished	We understand the use of "Core" to be synonymous with that used in "Core Components".
3	**Automatically generated documentation**	#2 (doc content must be embedded)	Auto generated content should be semantic (not statistical) in focus and should explain usage properly. This means that this deliverable would have to be "stocked up" in conjunction with schema development. Enforcing the doing of this should be covered in the design rules.
4	**Example instance for each schema**	#2	
5	**Customization methodology**	#2 must have been well fleshed out (and the decision about an extension vs. restriction design principle must have been made)	The need for this is predicated on when people start really using our document types. This has to be part of the first deliverable, because most users will need it and many implementations will be gated on having a good way to do and manage this. The methodology should include guidelines on ways in which to restrict divergence in extensions. This is a "harmonization" concern.

6	**Mappings to ebXML core components**	This can guide #2, or #2 can guide this	We expect the Mapping subcommittee to provide much more information about this.
7	**User documentation for business documents**	We need to know #5 before doing this	
8	**Developer website, newsletter, and other outreach**	None	#7-8 are good ideas, but are categorized as "lower priority". Development takes precedence and has "high priority".

Deferred Recommendations

Issue: The following deliverables were considered, but we needed to defer decisions on these because they require the input of other subcommittees.

#	Name	Rationale	Comments
9	**Mappings to other systems besides ebXML core components**	This depends, first, on the liaisons we form, and second, on what we negotiate with the other organizations.	The list of desired mappings will depend on the liaison activity. We already have a subcommittee to map this to the ebXML core components. The actual mapping work may be done by the other groups, or may be done by joint committees, so it is unclear if these are deliverables yet.
10	**Business document type creation methodology**	We just don't feel ready to decide this yet.	xCBL has this in the form of a "BP spec," and Commerce One has a tool internally that actually creates a new schema based on your input. This area may need to be deferred to Phase 2, in which context drivers are considered.
11	**Standard choreographies**	We're very doubtful that this should be a deliverable, since this depends so heavily on the community, but our liaison with various organizations may suggest that we need to own one or more of these.	Some very high-level "process buckets" (sample choreographies) will probably inform our scope discussion, but for real-world choreographies, we think this should be deferred to Phase 2. Often it has to be negotiated at the community level, so "standard" choreographies are suspect.
12	**Methodology for how to map to other systems**	We need to refer this to the mapping subcommittee.	We can imagine what this is, but we don't have any concrete ideas about it. We suspect that this is out of scope for what we're trying to do right now, but we will defer to Tim McGrath (chair of the mapping subcommittee) on this.

Recommended Non-Deliverables

We recommend that the following not be considered deliverables.

#	Name	Rationale	Comments
13	**Standard customizations**	These depend too heavily on the communities that need them, and so we should recommend that liaison organizations take up the development and maintenance of customizations.	We're concerned about being in this business; the different vertical markets should do this themselves. We might want to do samples as a help to developers. However, if the really small players are off developing a bunch of incompatible customizations, we're defeating the original ebXML vision.
14	**Sample stylesheets for pretty-viewing sample documents**	These have to be non-normative, and are better done on someone's free time.	Concern was raised about the subjectivity of stylesheets, and it would also have a fairly low priority. If we ever get around to them, it would be best done in the context of developer documentation, so they can be accurately characterized. We would be happiest if these were contributed non-normatively (skunkworks). There are also questions about using XForms for the interactive parts of a document.
15	**Automatically generated schemas in other schema languages**	Others can do this with standard tools, and we don't want to be responsible for maintaining a normative version since the transformation can differ depending on goals. Also, it may constrain our design options.	

Scope

We recommend that the following business document types be considered in scope for Phase 1 of the TC's work. The document type categories are shown in priority order. The document types listed in each category are "master types" whose design can largely be reused for additional types in that category, such that the plan of work can be made extensible. It would be a useful exercise to continue listing future "waves" of document types for these categories and prioritize them.

Document types separated with slashes are different types that have especially similar structure, though similarities are all over the place. In some cases, it is unclear yet how many document types there are in a category, as they are combined or separated differently in current usage; there needs to be a design principle about this.

0. **Core Library** category:
    - All the base-level and aggregate core components needed by the other categories
1. **Trade/Procurement** category:
    - Purchase Order/Purchase Order Response/Purchase Order Change
2. **Materials Management** category:
    - Despatch Advice (international) or Advance Ship Notice (U.S.)
    - Planning Schedule/Shipping Schedule
    - Goods Receipt
3. **Trade/Payment** category:
    - Commercial Invoice

  - Remittance Advice

4. **Transport/Logistics** category:
    - Consigment Status Request/Consignment Status Report
      (this is status that's sent to the transport co., not the shipper; maps to X12 214)
    - Transport Contract (international) or bill of lading (U.S.)
5. **Catalog** category:
    - Price Catalog/Product Catalog (these are one thing in X12)
6. **Statistical Reports** category:
    - Accounting Report
      (possibly produced by XBRL, or by us in liaison with them)

Our rationale for this list and a description of how we built it is as follows.

We discussed whether to use a "process-primary" approach, but ultimately felt that it might favor too many document types that aren't used very much in practice. We decided that, in keeping with the pragmatic focus of the committee as a whole, looking at all the documents people actually exchange today would be more practical. We aimed for the 20 percent of document types that would be likely to meet 80 percent of the needs.

We ultimately examined three proposals: Arofan's list, a list that Mike Rawlins created by running an email survey asking about the most frequently exchanged documents among EDI users, and Sue Probert's synthesis of the two (which became our recommendation above).

We considered whether looking at current practice would bias the results towards big players as opposed to SMEs, but concluded that, if anything, smaller organizations will have even less flexibility and capability; the variety of documents actually exchanged frequently in electronic form was surprisingly low.

By choosing master types, we avoided excessive depth in any one area. Covering the master types with high quality will ensure quick development of additional types.

Manner and Schedule of Work

We recommend that individual subcommittees be formed to address each of the scope categories. (It might be practical to combine some of the committees because of scarce resources.) Each subcommittee would be responsible for contributing to not only the schema design but also other deliverables (such as sample instances), as appropriate. Our answer #31 in the project definition questionnaire, provided under separate cover, suggests some useful analysis input to these subcommittees.

To demonstrate progression to the community, the TC should prioritise the production of the purchase order and invoice document types first. The respective subcommittees should coordinate closely with each other because of the similarities between these two document types, but similarities abound; there will need to be close coordination in all cases.

We didn't want to get into an undecidable top-down vs. bottom-up discussion regarding how to approach the core library development vs. the actual document types; we thought that digging in and sharing the results with other subcommittees was the best way to proceed. The experience of the TC members will largely get us through this. The Core Library subcommittee would obviously be the "different" one, and it would benefit from "digging in" the soonest. We believe that the Mapping subcommittee will be suggesting ways to coordinate between core library work and document type work, and between document types.

Issue: We didn't have time to make other specific recommendations about additional subcommittees to create the entire spectrum of recommended deliverables.

Design Principles

Design principles are overarching goals that reflect the agenda of the group. These are different from schema design rules, which are very specific prescriptions for organizing and designing schema code.

Recommended Design Principles

We recommend the following intial design principles. (There may be others that emerge from the design work.) Where principles conflict we have attempted to recommend priorities. These were developed through a process of responding to a "project definition questionnaire", which is provided separately. We hope the questionnaire answers will provide useful fodder for future UBL activities, but does not itself constitute a recommendation from this subcommittee.

Interchange format ("various and sundry")

> Because UBL is purely an interchange format, its design cannot make any assumptions about sophisticated tools for creation, management, storage, or presentation being available. The lowest common denominator for tools is incredibly low (for example, Notepad), and the variety of tools used is staggering. We do not see this situation changing in the near term.

Time constraints

> Making fast design decisions is more important than making the "right" design decisions. Urgency is a key theme in the development of UBL.

Simplicity

> The design of UBL must be as simple as possible (but no simpler).

80/20 rule

> The design of UBL should provide the 20% of features that accommodate 80% of the needs.

Component reuse

> The essential nature of e-commerce transactions is to pass along information that gets incorporated again into the next transaction down the line. For example, a purchase order contains information that will be copied into the purchase order response. This forms the basis for our need for a core library of reusable components. In fact, reuse in this context is important not only for the efficient development of software, but also for keeping audit trails. Thus, the design of document types should share as many common features as possible.

Domain expertise

> It will be critical to leverage expertise in a variety of domains. (The Mapping subcommittee is talking about this as well.)

Customization and maintenance

> The design of UBL must enable customization and maintenance. (What nature of customization hasn't been decided yet.)

Context sensitivity

> The design of UBL must ensure that context-sensitive document types aren't precluded.

Prescriptiveness

Having precise, tight content models and datatypes is a good thing (and for this reason, we might want to advocate the creation of more document type "flavors" rather than less; see below). However, in an interchange format, it is often difficult to get the prescriptiveness that would be desired in any one usage scenario, so this will have to be balanced in UBL.

Content orientation

Most UBL document types should be as "content-oriented" (as opposed to merely structural) as possible. Some document types, such as product catalogs, will likely have a place for structural material such as paragraphs, but these will be rare.

XML technology

We should avail ourselves of standard XML processing technology wherever possible (XML itself, XML Schema, XSLT, XPath, and so on). However, we should be cautious about basing decisions on "standards" (foundational or vocabulary) that are works in progress.

Potential Design Principles

Issue: The following are potential design principles that need more discussion.

Number of document types

Should UBL favor the creation of many different (but similar) document types, using the RosettaNet PIP approach that says that every transmission in a particular process is another document type? To group logical document types together into a single schema leaves less room for prescriptiveness, and may be false economy.

Programming models

What sorts of programming models should be enabled?

Mismatches between schema languages

What should UBL's relationship to the capabilities of various non-XML Schema schema languages be?

Extension or subsetting

Should the schema design favor extension (LCD) or subsetting?

New vs. old

How do we balance redesigning e-business and legacy system requirements?

B2B vs. others

How do we weigh the needs of document types for ERP-to-ERP, enterprise-to-enterprise, and mixed communication?

Non-Principles

We recommend that the following not be design principles.

Relationship to other namespaces

We don't need to reuse existing namespaces wherever possible; in fact, we should be cautious about making dependencies on other namespaces. For example, XHTML might be useful in catalogs and comments, but it brings its own kind of processing overhead, and if its use is not prescribed carefully it could harm our goals for content orientation as opposed to structural markup. (**Issue**: Is this a principle or a non-principle?)

Legacy formats

UBL is not responsible for catering to legacy formats; companies (such as ERP vendors) can compete to come up with good solutions to permanent conversion. This is not to say that mappings to and from other XML dialects or non-XML legacy formats wouldn't be very valuable.

Natural languages

Incorporating multiple natural languages into a single business document instance does not seem to be a goal for now. (However, there may be a need to "link" two instances that are simply two versions -- say, English and French -- of the same purchase order.)

Controlled variant of xCBL

It is not important for UBL to be a strict subset of xCBL, or to be explicitly compatible with it in any way.

Acknowledgments

The UBL Planning subcommittee had the following members:

Mavis Cournane
Adam Cheyer
Mark Crawford
Arofan Gregory
Eve Maler (chair)
Murray Maloney
Sue Probert
Mike Rawlins
Gunther Stuhec
Jim Werner
Lauren Wood