A COMPARISON OF XBRL AND SDMX-ML[1]
December 2004

This note is aimed to evaluate two technologies, XBRL and SDMX-ML, both as transmission formats and in their perspective ability to potentially play a wider role in the given context[2]. The evaluation context will be defined (paragraph 1). A comparison of the two information models (paragraph 2) and syntaxes (paragraph 3) will be made. A practical comparison will be given, based on an example (loans, deposits and banking branches, classified by municipality) taken from Banca d'Italia supervision practice (paragraph 4). Finally, some conclusions will be outlined (paragraph 5).

1. THE EVALUATION CONTEXT

The business context is represented by the *systematic reporting between banks and national supervisors or central banks*, that can be so characterised (see figure 1-1):

- the system topology is a "hub" one, where there is a central authority that, according to a specified protocol, receives reports from a number of banks and sends some other flows (return flows, maybe remarks) to them. Data are processed and sent to other internal and external users;
- there is a lot of different reporting schemas[3];
- there is a lot of input and output data flows;
- exchanged data are usually quantitative (i.e. numeric) and their identifiers are one- or multi-dimensional;
- the level of automation is not null and refers, to a minimum, to the usage of electronic media (network or other media) for data transmission.

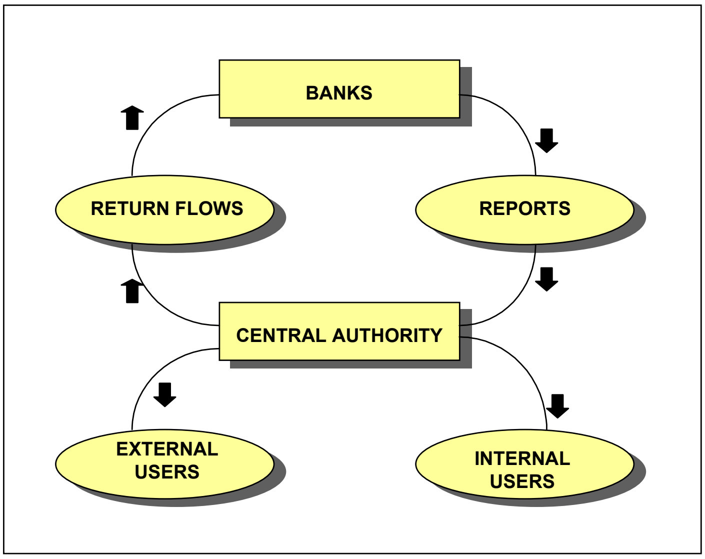

Figure 1-1 – The information circuit

[1] Written by P. Milani and M. Romanelli.
[2] SDMX-ML has been considered together with other SDMX standards.
[3] A reporting schema is the administrative description of some part of the overall reported information, that refers to one or more subjects and has a reporting periodicity or time schedule.

Coming to the technologies under evaluation, it is assumed that the reader is familiar with both XBRL and SDMX initiatives. Information can be found at the corresponding web sites, www.xbrl.org and www.sdmx.org. A short summary will be given anyway.

XBRL stands for eXtensible Business Reporting Language and is a royalty-free, open specification that uses XML data tags to describe and give context and structure to financial information, for its use in the information supply chain. In XBRL words[4], "XBRL defines a syntax in which a fact can be reported as the value of a well defined reporting concept within a particular context. (*omissis*) XBRL instances contain the facts being reported while the taxonomies define the concepts being communicated by the facts". There is an ongoing development of the XBRL specification.

XBRL is being developed by an international non-profit consortium of approximately 250 major companies, organisations and government agencies[5]. XBRL is used in some world-wide countries, essentially for financial reporting. Some usage begins to be seen also in supervision reporting. There is a project, named "XBRL in Europe" and financed by the European Commission within the 6th Framework Programme, to accelerate the awareness about XBRL capabilities.

Market tools for taxonomy and instance documents creation and validation have been developed by some software firms.

SDMX stands for Statistical Data and Metadata Exchange initiative. The initiative was started in 2001 by (alphabetic order) the Bank of International Settlements (BIS), the European Central Bank (ECB), Eurostat, the International Monetary Fund (IMF), the Organisation for Economic Co-operation and Development (OECD), the United Nations Statistical Division, the World Bank, with the aim "to set standards that can facilitate the exchange of statistical data and metadata using modern information technology". Although the "S" of the acronym stands for "statistical", SDMX standards should be intended as able to handle data to be used within a cognitive (i.e. business intelligence) process; thus, they fully apply to the given context.

The "catalyst" of the initiative was GESMES/TS, the EDIFACT format used to exchange statistical data and metadata, in the form of time series, within the European System of Central Banks (ESCB) and BIS circuits, and proposed for use also to the other SDMX Sponsors, i.e. Eurostat, IMF and OECD, involved in the data exchange with BIS and ECB themselves. In fact, GESMES/TS has been the initial core of the SDMX work and the message is maintained under the SDMX auspices.

SDMX has recently added to GESMES/TS-EDIFACT another exchange format, named SDMX-ML (where ML stands for Mark-up Language) and based on a very similar information model[6], but using the XML syntax instead of the EDIFACT one; data can be transported in the form of time series or cross-section. A first application of this format would be in the ESCB context, for improving accessibility of ESCB statistics.

Another SDMX standard is the SDMX information model, that defines the rules with which data are modelled.

Software tools are presently available in the GESMES/TS context only. Other tools could be developed in the other contexts. SDMX tools come from single or co-operation initiatives and they are usually free of charge.

The whole set of SDMX standards, that are royalty-free and open, has been approved by SDMX Sponsors at the end of September 2004. They would be submitted to ISO as proposed ISO standards. There is a planned development for SDMX standards, both as an implementation of what has already been done and as the provision of further standards, "based on a common information

[4] See XBRL specification 2.1, 31 December 2003, page 14.
[5] Late 2004 data.
[6] The SDMX information model.

model" and focused on "the use of a standard infrastructure, based on the use of a registry framework"[7].

2. A COMPARISON OF XBRL AND SDMX-ML INFORMATION MODELS

An information model is a specific set of rules with which data are defined (i.e. modelled). In the context of the paper this is a crucial aspect to be analysed, because it determines what kind of data can really be handled.

There are a lot of information models all over the world, because every "entity" (organisation, technology and so forth) that handles data adopts one of them, "consciously" or "unconsciously"[8]. XBRL and SDMX-ML technologies are no exception. Both of them transport data and both of them model data someway. SDMX-ML is an example of "conscious" modelling, because its conceptual layer, i.e. the SDMX information model, has been explicitly and formally described, and is even an SDMX standard. XBRL is an example of "unconscious" modelling, because its information model has not been explicitly described and must be inferred by the XBRL specification.

In the following the two information models will be firstly described and then compared. The analysis will be based on XBRL Specification 2.1, 31 December 2003 (SPEC) and on SDMX Information Model version 1.0 (IM), with a glance to envisaged development initiatives.

XBRL information model description

XBRL information model identifies reporting concepts of interest, giving them a context and grouping them in taxonomies.

Concepts are definitions of reporting terms, called also facts, and have a name and a type. Concepts are grouped in taxonomies[9], that seem to refer either to something "logic" (e.g. the description in XBRL terms of International Accounting Standards requirements) or to something that has to be really exchanged, i.e. a reporting schema (e.g. a financial statement in accordance to International Accounting Standards).

Concepts can be "real" (i.e. corresponding to true data) or abstract (i.e. data groupings, useful e.g. for presentation purposes). Facts can be simple or compound and their types can be various (numeric, alphanumeric, monetary and so forth). Monetary facts can have a "balance" attribute, where to specify their "debit/credit" nature. Simple numeric facts have a unit of measurement and, in general, a precision or a #-of-decimals qualification; the precision can also be inferred, according to some very articulated rules.

The "context" contains information about the entity being described, the reporting period and scenario that refer to a fact. Simple facts must have a context, while compound facts, called "tuples", must not. The context is articulated in "entity", "period" and "scenario" elements.

Entity is the subject of XBRL items[10]. In cases where the entity is insufficient to completely identify the subject, the "segment" element can be used; this element is optional and is considered a "container for additional mark-up"[11].

[7] See documents "Framework for SDMX Standards (version 1.0) – Background note for ISO Submission" and "SDMX Information Model (version 1.0)".
[8] In this context, "conscious" modelling means that the information model has been formally described, e.g. using the "entity-relationship" technique or the "Unified Modelling Language" technique, and/or semi-formally described, e.g. using a narrative form, while "unconscious" modelling means that a formal or semi-formal description is not available.
[9] In technical terms, a taxonomy is an XML Schema.
[10] In case of reporting to supervisors or central banks it would usually be the signalling bank.
[11] See SPEC, page 55.

Period is an instant or duration in time[12]. Scenario is an optional element, whose semantics is not clearly defined (i.e. a "container" element), used to specify some additional characteristics of the described fact, that preparers want to associate with items. Using XBRL words, "scenario is used to document the circumstances surrounding the measurement of a set of facts"[13].

Business facts, as described by the SPEC, are deemed to be intrinsically one-dimensional (the fact itself). However, the SPEC (page 56) and Financial Reporting Taxonomies Architecture (FRTA) 1.0, Candidate Recommendation 2, dated 2004-05-15 (paragraph 2.3.2, example 9) seem to suggest a possible way to specify multi-dimensional data:

- the dimensions of analysis, referring to "entity", would be specified in the "segment" element;
- other measurement elements would be specified in the "scenario" element.

Valid values for some XBRL elements (e.g. facts and context elements) are enumerated within the taxonomy description. Valid values for some other XBRL elements (e.g. those elements that qualify the measure) are explicitly defined in the SPEC.

There is also the possibility to provide additional information by expressing relationships[14] between concepts or associating concepts with documentation about their meaning. XBRL defines some standard relationship types but its syntax offers the possibility to define other "custom" relationship types.

Concept documentation is given by: the "label" type, that explains in plain and multilingual text the meaning of the relating concept, giving also the possibility to specify more than one label, each of which can be used in specific circumstances; the "reference" type, that, using the XBRL words[15], "contains relationships between concepts and references to authoritative statements in the published business, financial and accounting literature that give meaning to the concepts".

Relationships between different concepts are permitted by "presentation", "calculation" and "definition" types.

The "presentation" type is intended to describe presentational relationships between concepts, arranging them into a hierarchy and specific ordering, possibly also specifying the preferred label to be used in presentation. The "calculation" type describes additive calculation relationships between concepts. The "definition" type is intended to contain a variety of miscellaneous relationships between concepts. The footnote is another kind of relationship, used to express irregular linkages between facts.

So, XBRL information model is focused on the single data (business fact), to whom a context is given, and on taxonomies, as groupings of business facts. Taxonomies can be intended as "logical", i.e. the representation of some data, of "physical", i.e. the representation of data to be really exchanged. Relationships between business facts are possible. Some model elements have the list of valid values included in the model itself, while some others (e.g. the business fact) have the list included in the taxonomy (i.e. in the XML Schema) as an enumeration. The information model has some optional "container" elements (e.g. "segment", "scenario" and "definition relationship" elements), whose real usage must be actualised within a taxonomy that makes use of them, and has some customisation possibilities. With this respect XBRL information model, as inferred by the SPEC, can be viewed as a "*modelling tool box*", that offers a core set of modelling rules, precisely defined in their semantics and *intrinsically able to handle one-dimensional identifiers*, and an optional and a priori not completely defined set of rules, that can be added to the standard set within

[12] In case of reporting to supervisors or central banks it would usually be the date to whom data refers.
[13] For example, facts can be reported as actual, budgeted, pro forma, etc. For internal reporting purposes, that can be an even greater variety of additional metadata. See SPEC, page 56, 57.
[14] In technical terms, relationships are expressed using XML XLINK.
[15] See SPEC, page 96.

a specific taxonomy by giving a precise semantics and/or structure to the “container” elements and/or to the customisation choices. XBRL documentation gives some indication on how to handle multi-dimensional identifiers, but its level of precision could be ameliorated[16].

In general one could envisage the presence of a *dictionary*, where to insert all definitions[17] and from where to start for the (desirably automatic) generation of all XML “objects” needed by XBRL (i.e. an XML Schema and additional linkbases). The dictionary could also help for definitions (i.e. metadata) reusability in different taxonomies. The dictionary model should be defined by the organisation willing to use SPEC, or by the software vendor willing to enter the XBRL market; it would be compliant with SPEC modelling rules.

SDMX information model description

The IM is a set of rules able to identify and qualify numeric one- and multi-dimensional numeric data, which have time associated with them, and to apply those modelled data to an exchange context. All descriptions are multilingual. The IM has been formally described, using the Unified Modelling Language (UML) technique.

The data modelling rules are the following:

- there are the “concepts”, that is the variables[18] used to identify or qualify (i.e. describe) the numeric one- or multi-dimensional data. Concepts can be “coded” or “uncoded”; these last are plain text. A concept used to identify the data must be a coded one. A specific concept is “time”. Concepts are described by code and description;
- there are the sets of values (“code lists”) each coded concept can assume. Each value is described by code and description;
- there are the data structures (“key families”), i.e. the various sets of different concepts used to identify (“dimensions”) and qualify (“attributes”) the data, together with their code lists. Some concepts, like e.g. “observation status”, are mandatory in every key family. Key families are described by code and description. Within a key family it is possible to define an “attachment level” for its attributes, based on some data grouping criteria[19]. Each attribute is “attached” to a specific level and its value changes synchronously with the level elements[20]. The attachment level is deemed important to reduce “verbosity” of XML files.

All the preceding metadata types are collectively called “structural metadata”.

The exchange context modelling rules are the following:

- there are the “data sets”, that represent the sets of data really exchanged or disseminated in a specific exchange circuit, together with their reporting periodicity. Each data set transports data that refer to a precise key family. Data sets are made up either of a number of time series, or of a view of several related time series’ data at a single point in time (cross-section). Data sets are described by code and description;

[16] For example the indication: (i) does not exclude some usage of “segment” and/or “scenario” elements, other than that of multi-dimensionality; (ii) does not make it clear if all “segment” element components are identifiers, and if all “scenario” element component are qualifiers; (iii) does not clarify the role of “entity” element: if always an identifier, if always a qualifier, if both choices are possible and, in this case, how to specify the role.

[17] Including the actual content of “container” elements, together with the list of their possible values (usually called “code lists”).

[18] Examples of variables are country, unit of measure, methodology, confidentiality and so on.

[19] Data groupings are “slices” of the key family, defined by subsets of the code lists of the key family dimensions. IM defines four types of attachment levels: the “data set”, the “group key”, the “time series”, the “observation”.

[20] For example, the “unit of measure” concept, assumed at constant value for every time series, changes its value only when the time series changes.

- there are the "domain categories", that define, in a hierarchical way, the knowledge domains of interest for which the information is collected or disseminated. Domain categories, together with the data sets referring to them, can be used to define the taxonomy of a specific exchange context. Domain categories are described by code and description.

Footnotes specification is possible, both for structural metadata and for data sets.

So, IM is focused on *numeric* data only and distinguishes between data definitions (i.e. "structural metadata") and exchange context definitions. *IM has a semantics completely defined* and can be considered as both the SDMX-ML conceptual layer and the model of a dictionary where to insert all definitions. From this dictionary one can start for the automatic[21] generation of all XML "objects" needed by SDMX-ML (i.e. one or more XML Schemas, depending on the message types used[22]). The dictionary would ensure definitions (i.e. metadata) reusability in the same or different exchange contexts.

Information models summary

Table 2-1 summarises the major characteristics of XBRL and SDMX information models. The issues that are deemed not to be present in both parts are underlined in the missing part. In the XBRL context, the issues that do not correspond to SPEC modelling rules of precise semantics are in bold.

Issue	XBRL	SDMX
Data definition	Taxonomy and additional linkbases	Concepts, code lists, key families
Exchanged data definition	Taxonomy and additional linkbases	One or more data sets
Classification of the exchanged data	Not contemplated	Specific model element
One-dimensional identifier	Business fact	A defined concept
Multi-dimensional identifier	The identification, besides the business fact, could be provided by "segment" and "scenario" elements	Some defined concepts
Measure	Business fact occurrence	Observation value
Type of measure	Numeric or alpha-numeric	Numeric
Standard qualification of measure	"Unit of measurement, "precision", "#-of-decimals", "balance" elements	Some defined concepts
Data subject	"Entity" element	One or more defined concepts
Other qualification	Some variables, to be specified into the "segment" element if they refer to "entity" or in the "scenario" element if they refer to the measure	Some defined concepts

[21] SDMX documentation seems to ensure the feasibility of this automation.
[22] For the meaning of "message type", see paragraph 3.

Time	"Period" element	Specific concept
Time series	Not contemplated	Specific model element
Attachment level	Not contemplated (each business fact has its own "context")	Specific model element
Code list	Envisaged by the SPEC for some elements only. For all other elements it is defined in the taxonomy	Specific model element
Footnote	SPEC element	Specific model element
Abstract data	Business fact	Not contemplated
Compound data	Tuple	Not contemplated
Links between data	Specific SPEC elements (i.e. relationship types), that link business facts	Not contemplated (envisaged enhancement, applicable to every concept and code list)
Sender and receiver information	Not envisaged by the SPEC	Specific model elements

Table 2-1 – Comparison of XBRL and SDMX information models

Information models evaluation

An evaluation of the two information models will be made taking into account the business context defined in chapter 1. The evaluation will refer to the models as they are now; some further considerations will also be made about possible enhancements that are deemed to eventually enlarge or ameliorate the role that models could play in the given context, taking also into account the already envisaged development initiatives.

Preliminary considerations are the following:

- SDMX information model:
  - ✓ represents the conceptual layer of the other SDMX standards, including the SDMX registry;
  - ✓ has a completely defined semantics;
  - ✓ handles numeric data[23] with one-and multi-dimensional identification.
- XBRL information model:
  - ✓ has been derived by the SPEC;
  - ✓ offers a core set of modelling rules of precisely defined semantics that are deemed to intrinsically represent one-dimensional data identifiers only;
  - ✓ offers some "container" elements, whose semantics is specified at a generic level;
  - ✓ offers some customisation possibilities.

a) the models as they are now

[23] Alpha-numeric information is handled at the level of uncoded attributes only.

XBRL information model first of all should be clearly defined, e.g. using a formal technique and/or a narrative form. In the end, it could become the model of an "official" XBRL dictionary.

More specifically, for XBRL to be applied to the given context a "*profile*" able to model multi-dimensional data should be uniquely identified. A possible assumption could be that suggested by XBRL documentation, specifying that the "segment" element would be used for identifying concepts other than the business fact, and the "scenario" element would be used for the qualifying concepts other than those that refer to the measure in the SPEC.

Moreover, this XBRL "profile" should have a semantics precisely defined in all parts, e.g. uniquely qualifying the use of "container" elements and avoiding any customisation. That would facilitate the software development for the handling of XBRL documents defined accordingly, by giving the possibility to apply generalised solutions. The "profile" could also be defined with the aim of reducing the technical complexity of the SPEC.

With reference to the given context requirements, a formal or semi-formal definition could be limited to the cited "profile".

It would also be worthwhile noting that relationships between SPEC elements are presently focused on business reporting items only and they could have a lesser use in an XBRL "profile" for multi-dimensional data; in fact, in that case links could be useful for all identifiers if one would fully exploit the features offered by relationships.

SDMX information model, to be seen as the conceptual layer of SDMX-ML format, defines numeric data in an exchange context. IM firstly defines the structural metadata (concepts, code lists and key families) that are used in the overall exchange. Previously defined structural metadata can be reused. After having defined such structural metadata, the administrator can define the data sets that form the exchange context, grouped by knowledge domains of interest[24]. All definitions can be included in a dictionary, whose candidate model is IM.

Some possible remarks are the following: (i) IM handles numeric data only; (ii) relationships between model elements are not possible (but it is an envisaged enhancement); (iii) the admitted roles for concepts are "dimension" and "attribute" only, while other roles could be desirable (e.g. what concept plays the role of business reporting item[25], unit of measure, data subject and so forth); (iv) it could also be desirable to have the possibility to specify code list subsets to be taken into consideration within data sets, in order to reduce, in certain cases, the number of key families to be defined.

<u>b) the models as they could be</u>

XBRL and SDMX-ML, as exchange formats, transport data modelled according to specified rules. In the end, they have defined an information model that, at the moment, suits their "core business" needs.

As XBRL and SDMX-ML are both respectful technologies, one could consider to be desirable if the two information models would enlarge their scope and evolve in a way that they could eventually become the base for the conceptual layer of a *metadata-driven software platform*, able to support many parts of the information supply chain.

An example of the scope of such a software platform, that applies to external and internal reports of banks, is depicted in figure 2-1, that roughly reproduces a slide presented some months ago by PricewaterhouseCoopers; in this figure:

[24] This meaningful grouping is usually called "taxonomy" in the literature.
[25] This role is deemed particularly important to facilitate the mapping between IM and the "star schema", and also the possible mapping with XBRL.

- there is an extraction from a bank's internal information system of all elementary data that are needed to generate the reports;
- there is a transformation and calculation of the extracted data, in order to produce the output data;
- there is an output data formatting, according to the reporting format(s);
- all processes are driven by metadata, i.e. by the set of definitions stored in a dictionary, whose data model is likely to be the information model.

The automation of such a context can also be implemented making use of data warehouse products and solutions, of course. What could distinguish the envisaged solution by a warehouse-based solution is that in our case the software platform could be *unique*[26], were the dictionary and the software able to handle definitions variability over time (i.e. metadata historicity).

Looking at some SPEC extensions[27], that are presently at the requirements stage, XBRL could have the intention to do some steps in that direction[28][29].

Assuming an evolution of XBRL in this context, its information model should first of all be clearly (i.e. formally or semi-formally) defined, not only for "conceptual" reasons but essentially for a guarantee of:

- a "*level playing field*" between users that want to buy an XBRL-based solution and users that want to follow an in-house development for such a solution without running excessive risks or without bearing perhaps relevant consultancy costs;
- a possible *interoperability* between XBRL products of different vendors[30], that should all ensure compatibility with an "official" XBRL information model.

---

[26] Present ETL tools are software generators, i.e. they produce specific software for each reporting schema.

[27] See "XBRL Formula Requirements", public working draft of 20 April 2004, and "XBRL Taxonomy Versioning Requirements", public working draft of 1 October 2002, at the XBRL web site.

[28] Reading the "XBRL Taxonomy Versioning Requirements" public working draft, 1 October 2002, one can find the following statements:

- "the purpose of taxonomy versioning is to allow comparison, analysis and aggregation of data represented in XBRL instance documents to be performed even when the source instances refer to different versions of the same base or extension taxonomies";
- "taxonomy versioning maintains information about how successive versions of a taxonomy differ from each other".

[29] Reading the "XBRL Formula Requirements" public working draft, 20 April 2004, one can find the following statements:

- "to effectively exchange business reporting information (*omissis*) often requires the applications to perform validation on data types and values, apply consistency checks, test data quality, augment the data with calculated values and possibly corrections, and to provide feedback to the producing application that indicates the nature and severity of problems encountered";
- "XML Schema is not sufficient to this task because it allows the validation of individual data elements and structural relationships but does not express constraints between elements. (*omissis*) Besides, the nature of rules is that they generally compute a whole series of results, some of which may be considered fatal errors, others as warnings, others as merely informational; an XML Schema validator mainly detects fatal errors";
- "there has now been sufficient experimentation and implementation experience with XBRL (*omissis*) to have illustrated both the need of an extension to the language to meet this need, as well as illustrating deficiencies in schemes used to date and the typical patterns of usage and the limitations on what that language actually needs to cover".

[30] Interoperability between products of different vendors is a serious argument. Limiting the analysis to the data warehouse field, all products adopt a common data model, the "star schema". Moreover, a reference model (Common Warehouse Metamodel: CWM) has been defined by Object Management Group (OMG) in order to standardise metadata exchange between data warehouse and business intelligence products of different vendors, thus ensuring interoperability between them.

The presence of a formal or semi-formal definition of XBRL information model would be a prerequisite to open a discussion about the feasibility of such a model to be used as the base for a metadata-driven software platform.

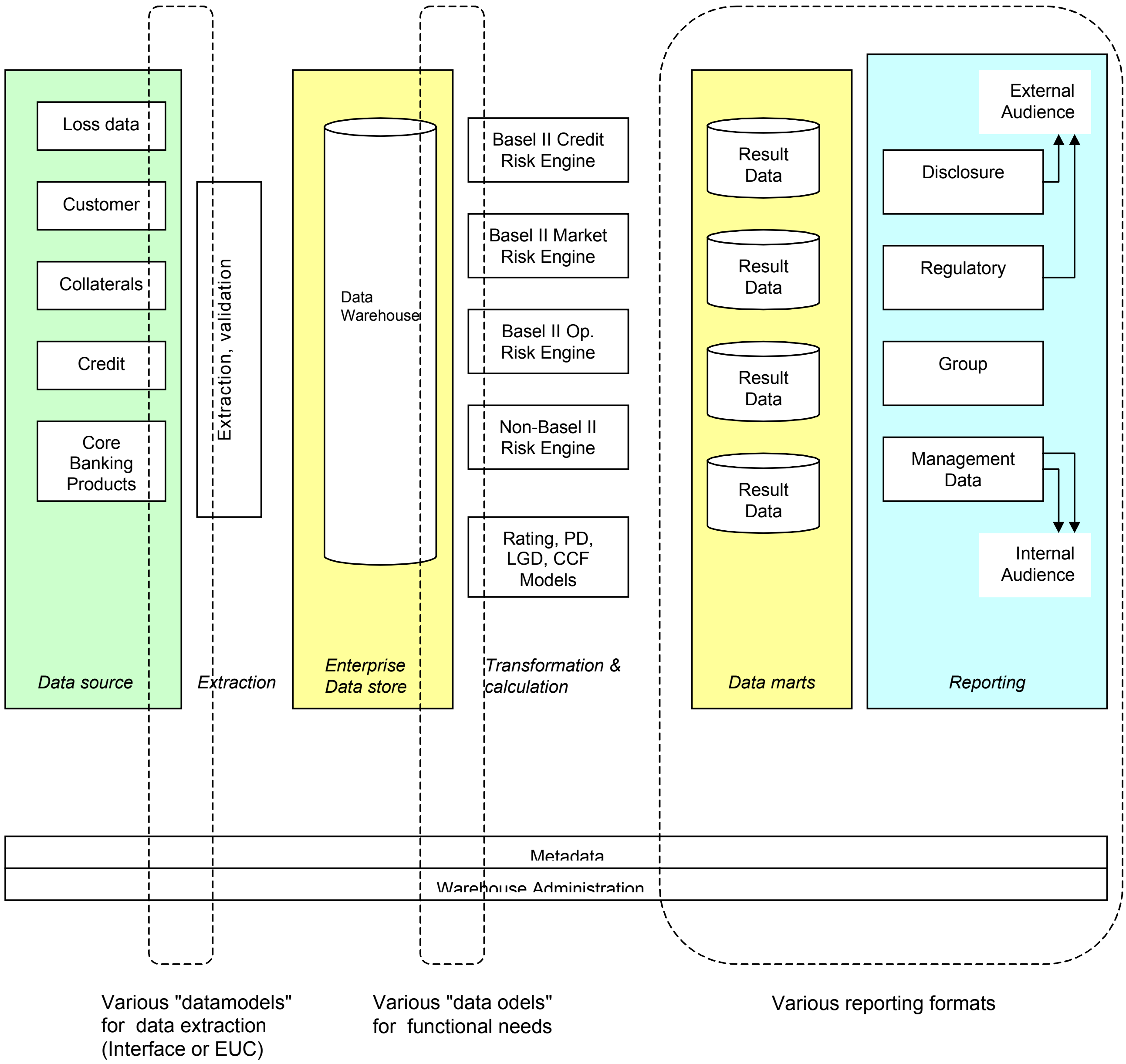

Figure 2-1 – The scope of a software platform in a bank environment

SDMX information model envisaged enhancements refer to hierarchies within concepts and code lists, that seem to be oriented toward the definition of links of various types. Were SDMX willing to pursue the depicted scenario, further enhancements to its present information model are deemed useful:

- the handling of other than numeric data types;
- a wider concepts modelling, like e.g. synonyms and events (e.g. mergers and acquisitions) handling[31];

[31] These cases are not infrequent in a context like that defined.

- a wider code lists modelling, like e.g. code lists with properties[32] and multi-dimensional identifiers[33];
- a generic modelling of data groupings (i.e. generic data "cubes");
- the definition of the code list subset to be taken into consideration within a "cube"[34];
- the representation of calculation expressions;
- the modelling of other parts of the given business context, like e.g. remarks and reporting schemas as data set groupings.

Metadata historicity should also be taken into account.

3. A COMPARISON OF XBRL AND SDMX-ML SYNTAXES

The comparison aims to point out:

- the XBRL and SDMX-ML characteristics that refer to data exchange;
- some specific aspects of the two syntaxes;
- the ability of the two technologies to delegate to an XML standard software the integrity constraints validation, i.e. how much part of the validation process can be performed using standard XML Schema validation software.

It is assumed that the envisaged XBRL "profile" for multi-dimensional data has been identified.

An **XBRL** taxonomy is comprised of an XML Schema and of all the XML linkbases contained in, or directly referenced by, that schema. An XBRL instance is an XML document that contains the facts being reported; it can make use of more than one taxonomy. The combination of an XBRL instance and its supporting taxonomies and additional linkbases constitute an XBRL business report and the set of related taxonomies is called a Discoverable Taxonomy Set (DTS).

Information about the exchange process[35] is *not considered* by the SPEC; it could be specified in the taxonomy as any other data, in a properly defined "envelope" structure, in a dictionary. The choice to specify this information in the taxonomy should be avoided, because it could force organisations toward the development of specific software for every incoming message (so called "stovepipe" approach).

While XML makes use of the "unique" construct in order to identify duplicate items, this task is performed by SPEC, for single and compound items, making use of some equality predicates, that seem very articulated.

Each XBRL simple fact has its own context, i.e. its own entity, period, scenario, and the context has to be explicitly declared for each fact. That means e.g. that an XML Schema validation software

---

[32] A code list structure made up of a code and a description seems insufficient for all cases, because some code lists are really registries (e.g. banks' code list).
[33] An example of this type is the data set code list, whose identifier is usually multi-dimensional. These code lists are important for compatibility rules validation.
[34] For example: a specific business fact, classified by country, could be valid for some countries only within the general code list; or some combinations only of the dimension code lists could be valid for a data set. Although the handling of code list subsets (i.e. "use domains" in relational terms) is demanding in terms of administration effort, it nevertheless facilitates the validation of some integrity constraints (i.e. compatibility rules).
[35] E.g. sender and receiver information, type of exchange, type of message, reporting periodicity and so forth.

cannot check that more than one fact must have the same context or part of it[36]; moreover, this could imply some "verbosity" in the corresponding XML document.

As a general feeling, SPEC seems more oriented toward data definition than to data exchange. It seems to have a significant level of complexity, that has been ascertained also by independent sources[37], and to imply a sophisticated use of XML and related languages (XLINK, XPATH, XPOINTER and so forth). This could have some impact on organisations, in terms of needed skills, consultancy costs, acquisition of market tools. The envisaged XBRL "profile" for multi-dimensional data handling could and should reduce someway this complexity. In the given context, the level of complexity could be different for central institution and banks; in fact, a central institution would also define reporting schemas and send to banks the corresponding XML "objects".

SDMX proposal admits the presence of different exchange types. Examples of exchange types, derived from the GESMES/TS experience, are "update and revision"[38], "delete"[39], "full dissemination"[40]. SDMX proposal defines some different XML document types, called messages, because of the combined requirements from the supported processes[41]. Each document type is described according to an XML Schema. All of these document types share a common "envelope" at the message level, as well as a set of common low-level components, so that header information and basic structure will always be the same. More specifically, there are document types for the transmission of structural metadata only, for the transmission of significant amounts of data in the form of time series or in the form of cross-section[42], for the generic dissemination. Each document type can refer to more than one key family definition; it is then able to transport more than one data set (e.g. a whole reporting schema). There is also a document type able to send queries to information sources (so called "data request message"). The "attachment level" concept should give some efficiency to SDMX-ML messages, because it avoids useless repetitions.

According to SDMX-ML documentation, all messages share the same IM and then it is always possible to map one message into another one; moreover, there is the possibility to automatically create key-family-specific schemas from structural metadata.

An XML Schema validation software is deemed to be able to check all what forms part of the IM, i.e. data structures, the list of values specified for concepts, equal attribute values between different data groupings. However, were a code list subset applicable to a data set this constraint should be verified by specific software.

As a general feeling, SDMX-ML seems to be quite simple and its XML usage seems "light" and likely neither to require sophisticated skills nor to imply consultancy costs. Other exchange information could be defined[43].

Table 3-1 compares the XBRL and SDMX-ML exchange features. The issues that seem not to be present in both parts are underlined in the missing part.

Issue	XBRL	SDMX-ML

[36] For example it would not be possible to specify, at the XML Schema level, that the signalling bank is always the same, as it is likely to be, or that data refer to the same time.
[37] See article "XBRL – A Case Study in Complexity" (www.infoworld.com/article/04/04/30/18OPstrategic_1.html). See also some brochures of XBRL products.
[38] Messages contain data variations only.
[39] Messages contain indication of some deletions to be made.
[40] Messages contain a whole data bank.
[41] See the document "SDMX Schema Design (version 1.0) and Documentation".
[42] These formats are key-family-specific.
[43] For example, there seems not to be a clear link between a reporting schema and the list of data sets that transport that information. Moreover, IM does not model some aspects of the exchange process, like e.g. data quality remarks.

Reporting scheme	One or more taxonomies and additional linkbases	One or more data sets
Exchange types	Not contemplated	Update and revision, delete, full dissemination. Other types are possible
Reporting periodicity	Not contemplated	Specific information
Exchange file types	Not contemplated	"Compact", "cross-section", "generic", "utility", "structure", "request" messages
Data exchange (i.e. business report)	An XBRL instance, supported by its XML "objects"	A data exchange file type ("compact" or "cross-section" or "generic" or "utility") plus its supporting XML Schema
Metadata exchange	One or more XML "objects" (XML Schema and additional linkbases)	Specific ("structure") message plus its supporting XML Schema
Data and metadata exchange	Combination of an XBRL instance and its supporting XML "objects"	Specific ("generic") message plus its supporting XML Schema
Sender and receiver information	Not contemplated	Specific section of the messages

Table 3-1 – Comparison of XBRL and SDMX exchange features

4. A PRACTICAL COMPARISON

The comparison aims to give readers a flavour of what the exchange of multi-dimensional data with SDMX-ML and XBRL looks like. It is based on an example (loans, deposits and banking branches, classified by geographic location at the municipality level) taken from Banca d'Italia supervision practice and on the envisaged XBRL "profile".

For sake of rapidity, the example: refers to just one bank; uses invented values for the quantities and for the information about sender and receiver; takes into account only a portion of the whole data set, that is municipalities belonging to the province of Turin; simplifies somehow the data structure ("residence" and "maturity" concepts have not been handled; "observation status" concept has been handled in SDMX solution only).

In the attached zip file one can find:

- the whole data set (Tdb10194_euro_eng_txt.zip);
- data and metadata of the subset used in the example (Example-SDMX-Tavola-TDB10194.xls);
- the SDMX-ML instance document for structural metadata (Example-SDMX-TDB10194-Structure.xml);
- the SDMX-ML document type and instance for data in the form of time series (Example-SDMX-TDB10194-Compact.xsd, Example-SDMX-TDB10194-Compact-sample.xml);

- the SDMX-ML document type and instance for data in the form of cross-section (Example-SDMX-TDB10194-CrossSectional.xsd, Example-SDMX-TDB10194-CrossSectional-sample.xml);
- the XBRL instance document (Example-XBRL-TDB10194-Instance-segment.xml).

In the example and using XBRL terminology:

- loans, deposits and branches are the business reporting concepts, together with the information about sender and receiver;
- the signalling bank is the "entity" element;
- municipality is specified in the "segment" element, while municipalities of the province of Turin are its corresponding data types;
- information about sender and receiver has been considered as another data.

In the example and using SDMX terminology:

- economic phenomenon and municipality are the identification concepts, while signalling bank, frequency, unit of measure, unit multiplier, number of decimals are the qualification concepts;
- the code list of the economic phenomenon contains loans, deposits and branches, while the code list of the municipality contains municipalities of the province of Turin.
- referring to the attachment level: frequency is constant for all data; observation status can vary from data to data; unit of measure, unit multiplier and number of decimals are constant for each economic phenomenon;
- two message types have been used, able to transport data in the form of time series and cross-section.

The size of SDMX-ML Compact is some 260Kb (kilobytes), the size of SDMX-ML CrossSectional is some 212Kb, the size of XBRL file is some 630 Kb. So, in this example XBRL has demonstrated itself some 3 times more verbose of SDMX-ML CrossSectional and some 2,5 times more verbose of SDMX-ML Compact. SDMX-ML better efficiency is deemed to depend on the presence of the attachment level, that gives qualification attributes the possibility to be specified only when their value changes.

Observing the aforementioned files, one could say that SDMX-ML examples seem more "readable" than XBRL one, perhaps because SDMX-ML XML statements can be opened or closed in ways that one can find significant[44].

5. CONCLUSIONS

This note has briefly described XBRL and SDMX-ML specifications, both as transmission formats and in their perspective ability to potentially play a wider role in the given context.

The XBRL specification is deemed by some market observers as something that could and should eventually become the global standard for "financial or business reporting". This opinion is due to increasing disclosure demand on companies by regulators and to the possible role of XBRL as an enabler for augmenting companies automation in this field. Moreover, XBRL International seems active in the definition of "market oriented" taxonomies[45]. The XBRL specification seems also to

[44] These ways fit with the SDMX modelling rules.

[45] For example, a joint project between International Accounting Standards Committee Foundation (IASCF) and XBRL International has defined a framework for preparing financial statements in accordance with International Accounting Standards, expressed using XBRL.

be considered by some national supervisors as a technology useful also for primary reporting purposes. There is a project, named "XBRL in Europe" and financed by the European Commission within the 6th Framework Programme, to accelerate the implementation and the adoption of XBRL in Europe.

The SDMX-ML specification is young and yet unused, but it is based on a conceptual layer that is very close to that of GESMES/TS; so, it capitalises on the success of GESMES/TS as the ESCB and BIS statistical information exchange format and its effectiveness is warranted, at least at the conceptual level.

Table 5-1 resumes the general characteristics of XBRL and SDMX-ML that apply to whatever context, while Table 5-2 points out the characteristics that specifically refer to the given context.

General characteristics	XBRL	SDMX-ML
Mission	Designed for business document formatting	Designed for the description of a systematic exchange, based on structured data
Handled data types	Numeric and alpha-numeric	Numeric
Level of usage	Used in some world-wide countries, essentially for financial reporting. Some tentative usage begins to be seen also in supervision reporting	Approved at the end of September 2004. A first application planned in the ESCB context
Data modelling rules	Inferred by SPEC. Not completely specified semantics (modelling toolbox)	Explicitly specified (SDMX standard). Completely specified semantics
Admitted exchange types	Aspect not contemplated	Update and revision, delete, full dissemination. Other types are possible
Admitted XML document types	Aspect not contemplated	Many types (i.e. messages), that can automatically be transformed one into another, because all of them share the same modelling rules
Efficiency of XML documents	More verbose than SDMX-ML	Less verbose than XBRL
Tools support	Some market tools support the taxonomy and instance document creation and validation	No tools presently available
Perceived complexity	Complex. Sophisticated use of XML	Simple. "Light" use of XML
Resources required	Could require consultancy or sophisticated in-house skills, especially in case of in-house development. Could require the purchase of some software	It is deemed not to require consultancy or sophisticated in-house skills. Data definition activity could leverage the

	tools, particularly for taxonomy definition	existence of a dictionary model

Table 5-1 – General characteristics

XBRL seems focused on business document formatting, while SDMX-ML seems focused on structured data exchange. SDMX-ML handles numeric data only.

XBRL has some world-wide usage, essentially in financial reporting. SDMX standards, except GESMES/TS, are very young and yet unused (although a first application of SDMX-ML has been planned in the ESCB context). However, it should not be undervalued that GESMES/TS is widely used world-wide since 1998 and it makes use of modelling rules that are very close to IM[46].

There are some market tools that support the use of XBRL, while there is none for SDMX-ML. At the same time, SDMX-ML seems able to delegate to an XML standard software a bigger part of the validation process.

XBRL modelling rules are not explicitly defined, while SDMX modelling rules constitute one of the SDMX standards. XBRL could be considered as a "modelling tool box".

XBRL-aware software development seems more complex than SDMX-ML-aware software development, especially for the central institution[47]. An XBRL-based solution could require consultancy and/or sophisticated in-house skills, particularly in case of custom development. It could also require the purchase of some software tools; this seems especially true for the central institution, that has to develop taxonomies.

Context related characteristics	XBRL	SDMX-ML
Multi-dimensional representation	Core set of modelling rules intrinsically one-dimensional. Needs a specific "profile"	Intrinsically one- and multi-dimensional
Data and exchange description	Dictionary model to be defined, according to XBRL modelling requirements	Dictionary model completely defined
Software development	An XBRL "profile" with completely specified semantics and reduced technical complexity is recommended	Semantics completely specified. Modelling rules clearly defined

Table 5-2 – Context related characteristics

An XBRL "profile" able to handle multi-dimensional data should be precisely defined and its semantics and modelling rules should be completely specified; this could facilitate software development for the handling of XBRL documents defined accordingly, by giving the possibility to apply generalised solutions. This "profile" should also reduce the overall complexity of XML usage

[46] IM is someway more generic than GESMES/TS model.
[47] A central institution should also define reporting schemas and send to banks the corresponding XML "objects".

made by SPEC. It is worthwhile noting that the "profile" could make a limited use of relationship types, because they are presently defined on business facts only.

The definition of mapping rules between the two technologies could be advisable, to facilitate interoperability between different organisations that have made different choices. This initiative could require some enhancements or clarifications in the data modelling rules defined up to now and would be facilitated by the formal or semi-formal definition of the XBRL information model, or at least of the XBRL "profile" information model. This formal or semi-formal description could also help institutions in taking decisions about a possible XBRL adoption and/or in software development.

To conclude, the two technologies have been evaluated in the given context taking into account the following two profiles: as exchange formats and in their perspective ability to potentially play a wider role.

The two technologies compete as exchange formats, where SDMX-ML seems designed for structured numeric data exchange, while XBRL seems designed for business document formatting.

That would be a scenario where a central institution – that has its own information model, dictionary and exchange process model – decides to add XBRL and/or SDMX-ML to its portfolio of handled formats, possibly proposing one of them as its reference format. In this context, the institution would continue to do the job as before and would develop some software that creates XBRL and/or SDMX-ML XML "objects" and handles XBRL and/or SDMX-ML messages. The mentioned XBRL "profile" should be defined. Mapping rules between XBRL and/or SDMX-ML information models and the dictionary data model should be defined (in this sense, a formal or semi-formal description of XBRL information model could help). SDMX-ML exchange information should be taken into account.

With reference to the perspective ability of the two technologies to potentially play a wider role in the given context, the data definition case will be firstly outlined.

That would be a scenario where a central institution decides to adopt XBRL or SDMX-ML modelling rules as its own information model. That would mean the development of a dictionary compliant with the new modelling rules, together with some software that creates XBRL or SDMX-ML XML "objects", handles XBRL or SDMX-ML messages and possibly connects the new dictionary to other software components.

In case of XBRL adoption, the mentioned "profile" should be defined, a formal or semi-formal description of its information model is deemed necessary[48], exchange information should be properly handled.

In case of SDMX-ML adoption, its exchange information should be taken into account.

In a broader perspective, the two information models could further enlarge their scope and evolve in a way that they would eventually become the base for the conceptual layer of a metadata-driven software platform, able to support many parts of the information supply chain. In this scenario, it might be desirable a convergence between the two.

[48] The presence of an "official" XBRL information model, defined e.g. by XBRL International, would allow a "level playing field" in terms of costs and risks between a "buy" and a "make" approach. More specifically, the "level playing field" would be between users that want to buy an XBRL-based solution and users that want to follow an in-house development for such a solution without running excessive risks or without bearing perhaps relevant consultancy costs.