

Understanding XBRL

Challenges for Software vendors
A roadmap



THE NETHERLANDS

Understanding XBRL

Challenges for Software vendors
A roadmap

This paper is provided to you by XBRL Netherlands on the understanding that it is solely recipient and for the purpose of assisting software vendors in formulating their XBRL strategy.

The paper must not be made available or copied in whole or in part to any other person without the express written permission of XBRL Netherlands.

XBRL Netherlands

XBRL Netherlands promotes the adoption of digital reporting based on XBRL in the Netherlands. XBRL Netherlands is a non-profit organization that is both financially and intellectually supported by a variety of members, ranging from government, businesses, intermediaries, research and development and education.

XBRL (eXtensible Business Reporting Language) is the global standard for digital reporting. It connects the generation, exchange and consumption of business reports with the dynamic and interactive world of the internet. XBRL provides a generic platform for business critical reporting processes and enhances the quality and efficiency of reports, for internal use or for external publication.

XBRL is based on XML and is a royalty free open standard, developed by XBRL International, a non profit consortium with over 400 members.

By:

Aad Bergman - Semansys Technologies

Paul Snijders - Semansys Technologies

Review:

Ineke Jansen - Marco Polo Platform

Dirk Jan Hoekstra - Semansys Technologies

Trevor Pyman - XBRL Australia

1st issue, May 2008

© Copyright XBRL Nederland 2008

TABLE OF CONTENTS

1	Introduction	7
2	XBRL for software vendors	10
3	Functional areas	12
4	About XBRL	14
4.1	Introduction	14
4.2	Data models	15
4.3	Summary	17
5	Data mapping	18
5.1	Introduction	18
5.2	Account mapping	19
5.3	Details or aggregated values	20
5.4	Data mapping solutions	20
5.5	Other data mapping issues	21
5.6	Relational data mapping	21
5.7	Dimensional mapping	22
5.8	XBRL GL	22
5.9	Summary	22
6	Instance documents	24
6.1	Instance document generation	24
6.2	Instance document validation	25
7	Taxonomy related processes	27
7.1	Introduction	27
7.2	Taxonomy development	27
7.3	Publishing the taxonomy	28
7.4	Taxonomy maintenance	29
8	Information systems involved	30
8.1	Accounting systems	30
8.2	Report generators	30
8.3	Fiscal/Tax solutions and Salary/Wage solutions	31
8.4	Accountancy solutions	31
8.5	Information flow examples	31
9	Software development strategies	33
9.1	Introduction	33
9.2	What options are available to software vendors?	34
9.3	The MAKE option	35
9.4	The BUY option	36

9.5	<i>The USE option</i>	36
9.6	<i>What is the best option for your organization?</i>	38
10	Road Map	39
10.1	<i>Introduction</i>	39
10.2	<i>Develop XBRL Publishing capabilities</i>	39
10.3	<i>Develop XBRL import capabilities</i>	39
10.4	<i>Adapt XBRL GL for transactional data exchange</i>	40
10.5	<i>Adopt the semantic and data modeling concepts in the application</i>	40
11	Summary	41

FOREWORD

This paper explores the impact of XBRL on the software industry in the financial reporting area. XBRL (eXtensible Business Reporting Language) is a language that has established itself as the standard for communication of business information between different organizations and information systems. XBRL is also targeted at exchanging financial and other business reports electronically between companies and regulators, banks and stakeholders as well as internally between different subsidiaries and business divisions. It is utilized in the existing product suite of small, mid sized and large software vendors.

Confronted with the emerging XBRL standard independent software vendors) need to answer questions that - amongst others - involve:

- In what way are your existing users confronted with XBRL compliance?
- Does it mean you have to provide XBRL functionality?
- If so, for which applications do you want or need to use XBRL?
- Exactly what XBRL functionality is needed?
- For the identified applications, how much time, resources and investment are required and available to be ready in time?
- Do you make or buy the necessary XBRL (enabling) technology?
- Do you build XBRL components into your (legacy) applications or do you obtain the use of a web based XBRL service provider to provide XBRL functionality?

Dutch businesses have to comply to XBRL due to the governmental choices in the Dutch Taxonomy Project (NTP). Therefore many software vendors already have been confronted with these and other questions. XBRL Netherlands has supported many vendors to cope with strategic decisions at hand. As a result XBRL Netherlands has written this paper to provide independent software vendors with a roadmap for the definition of their own XBRL (enabling) strategy.

The content of this paper is heavily influenced by the market experience by many software vendors for the largest XBRL project in the entire world: the Dutch nationwide implementation based on the successful Netherlands Taxonomy Project (NTP). This project confronted independent software vendors with the necessity to provide XBRL functionality to customers. At the same time other XBRL projects do the same XBRL message justice. The Dutch NTP project defines already around 30 different reporting duties in just as much interrelated reporting taxonomies for Tax filings, Annual reports

and Statistics reporting. Planned and executed properly, the potential of XBRL for business applications is of a highly interesting magnitude.

Before we start it is important to note that XBRL should neither be 'over-hyped' nor should be under-estimated for the challenges that lay ahead. In essence, an XBRL project is just like any other IT project or software development activity.

The difference is that the XBRL knowledge, XBRL core technology and/or XBRL components are required to develop a complete solution that somewhere connects to XBRL. The XBRL parts of this overall solution can range from XBRL software tools that can be built into existing applications to (web) server based XBRL enabling services. All these options are and will be available to facilitate the migration process. A process that, once started, creates opportunities, as well as questions to be answered. This roadmap is your guideline in the decision making process to migrate towards XBRL.

1 INTRODUCTION

The world in general and the sharing of data in particular becomes more and more digital each day. And the need for information increases, as well as the number of people who use data for their processes. These developments require an open, transparent and easy to use mechanism in order to make sure that everybody 'speaks the same language'. XBRL provides in this mechanism.

The importance of XBRL for software vendors is increasing as XBRL is rapidly establishing itself as that language, becoming the standard for communication of information between different organizations and/or information systems.

XBRL is an open, standards-based XML language for defining, preparing, publishing, exchanging, repurposing and analyzing business and financial information, distinguished by robust semantics and significant document content definition modelling capacities. These content models are used to create fact sets or instance documents and translate static data (stored and maintained in distributed and disparate systems) into dynamic and extensible information that can be shared, extended and presented according to each consumer's needs.

With XBRL organizations can dramatically improve the efficiency of collecting, validating, aggregating, analyzing and reporting business information within and between organizations (external) and departments (internal). By reducing manual data entry and inconsistent data definitions, significant improvements in data accuracy can be achieved. Information is entered only once, allowing it to be rendered in many forms, such as a printed financial statement, an HTML document for a website, filing documents with government entities or other.

The idea behind XBRL is simple. Instead of treating financial information as a block of text - as in a standard internet page or a printed document (or PDF) - it provides identifying tags for each individual fact item of data. This fact data plus tags are then computer readable. The introduction of XBRL tags enables automated processing of business information by computer software, cutting out costly and labour intensive manual re-entry and comparison processes and improving the data quality dramatically.

With XBRL computers can treat data "intelligently" as they recognise the individual data facts in a XBRL document, select it, analyse it, store it, exchange it with other computers and present it to users in a variety of ways. XBRL greatly increases the

speed of handling financial data, reduces the chance of error and permits automatic checking of information.

There are several reasons for the growing popularity of XBRL including:

1. The intrinsic attractiveness of the XBRL standard itself due its strong design principles
2. The standard provides users with:
 - a clear and commonly used definition of data elements
 - High flexibility
 - powerful and granular dimensional data structures
 - the power of business rules (formulas)
3. Wide spread global adoption of the XBRL standard - it is embraced by international standard setting and governmental bodies such as: - amongst others - the International Accounting Standard Board (IASB) for IFRS, the Security and Exchange Commission (SEC), collective European Central Banks, Dutch and other governments, major accounting firms and major financial services organizations.
4. The cost saving potential of XBRL - companies that use XBRL in their reporting process can save costs and streamline their processes of collecting and reporting financial information. Consumers of financial data, including investors, analysts, financial institutions and regulators, can receive, find, compare and analyse data much more rapidly and efficiently when the XBRL format is used.
5. The potential for universal application of XBRL, as XBRL can handle data in different languages and accounting standards. XBRL is eXtensible to meet different requirements and usages in a variety of environments. Data can be transformed easily into XBRL by suitable mapping tools or it can be generated in XBRL using appropriate software.
6. The proven XBRL track record. Major XBRL projects are already in practical and successful use for specific purposes in several countries and the initiative for projects in other countries.

The subsequent sections of this paper will provide information about XBRL's use, provide more detailed technical explanations and set out how producers and consumers of business information can adopt XBRL.

The opponents of XBRL appear to be fighting a losing battle. The opponents do not disagree with the potential of XBRL to improve information processes, but mainly argue that large investments are required to develop and implement XBRL based solutions. By reading this paper it will become clear that much of the current discussion takes place without a thorough understanding of what XBRL is and what is required to integrate XBRL within software solutions. The XBRL components and web service technology that is already on the market today from vendors like the authors can greatly facilitate XBRL based or XBRL enabled software solutions and significantly reduce required investments.

This paper will present you with a number of possible scenarios to make your software XBRL based or XBRL enabled. It provides the criteria required to make the right investment decisions and to establish a road map for implementation.

2 XBRL FOR SOFTWARE VENDORS

It is the goal of the XBRL standard to provide a common use solution to allow more efficient and better auditable internal and financial reporting. From a software or IT point of view XBRL as a standardized way of using existing open W3C compliant XML based techniques on a global basis. Some see XBRL as the financial internet revolution. This brings XBRL onto the radar of software providers in accounting, reporting, ERP and more,

Over the last decade a range of solutions where created to provide functionality in the financial and business reporting arena. The software market is filled with specific vendors for compliance reporting, a range of business intelligence and data warehouse vendors, report writers as well as application specific management reporting, tax applications, repository vendors and many, many more. As always, each individual vendor solution will solve one particular problem, but non have an interest in solving global problems. In fact point solutions are a hurdle to implement more common approaches and solutions and create a high cost factor for the users.



Each solution consists of a number of layers, including data definition, application logic, business rules, formatting information and more. Each application has its own way to store data definition, meta data, data dictionary or repository. Most applications have a structured way to define and store business rules and calculations and all have application logic. Some also include formatting or layout information.

For all solutions however the core of the application will be the data model. It is interesting that we can see many core data modeling approaches for business and reporting data, however the most common used way for business reporting is the spreadsheet which practically means; no data model at all. Further we see multiple concepts which include the relational database, the hierarchical database, -supported by many vendors- the multidimensional

database (cubes) and also (although not used very much anymore) the network database. These data models are good for the majority of legacy solutions.

It can be concluded that at least there is a wide variety of possible solutions to choose from. The bad news is that there is no common or standard way to describe data and there is no comparability, re-use or easy data exchange.

The main issues are:

- no standard data definition mechanism,
- no common data model,
- proprietary meta data definitions,
- impossibility to share business rule definitions.

This is where a global standard comes in. XBRL is invented to overcome these disadvantages and provide a common solution.

It could be considered that organizations that have one standard chart of account can of course do this exercise once for all reporting entities. However if a subsidiary has to report to local regulators like tax administration or statistics mapping must be done at subsidiary level. Also, if the company acquires a new entity it will take time before the standard chart of accounts is implemented. These complexity and variety problems are solved with the use of XBRL.

The standard chart of accounts may also not be adequate to meet subsidiaries' reporting needs as they are typically designed to meet Head Office requirements. If they constructed to meet all the needs of the subsidiaries, then they may become so large and complex that they eventually become unwieldy and difficult to maintain and understand.

3 FUNCTIONAL AREAS

Software vendors need to have a thorough understanding of XBRL and its impact on their information systems. In order to present the different scenarios available to software vendors, one needs to understand:

- which functional areas are impacted or created with the introduction of XBRL
- where the XBRL functionality can be integrated, does this need to be integrated in existing or to be developed software applications or are there other ways forward

In the next chapters we describe the different functional areas of information systems that are impacted by XBRL:

- Taxonomy creation (data definition and data structure)
- Mapping Management (data structure and creation ability)
- Instance Document Generation (reporting - form, format, publishing)
- XBRL Validation (requirements for data exchange)
- Receiving and processing (requirements for data exchange)
- Display and analysis (user interface and business requirements)

The XBRL parts of current information systems relate to the preparation and validation of messages (instance documents) and often consists of 'only' a portion of the complete application. Depending on the position of an organization in the financial reporting supply chain and the requirements for the used application different XBRL functionalities will be needed. It will be rare that all XBRL capabilities is required in one single organization. Some will be more involved in creating XBRL documents for reporting, others in receiving and processing.

It is important to note that there is no such thing as 'the' XBRL solution to be implemented. What XBRL parts are needed depends on how current information systems are set up. When we look at successful live XBRL projects around the world, we see information systems that share information between organizations. Systems with user interfaces, user management and (to a certain extent) reporting modules similar to those one would expect with information systems that are not XBRL based. The big difference though is the flexibility and interoperability of XBRL, making information systems dynamic, flexible and interoperable from user's perspective. The XBRL parts of these systems relate to the preparation and validation of messages (instance documents) and often consist of 'only' a portion of the complete application. Depending on the position of an organization in the financial reporting supply chain and the requirements different XBRL functionalities will be needed. It will be rare that all

XBRL capabilities are required in one organization. Some will be more involved in creating XBRL documents for reporting, others in receiving and processing.

4 ABOUT XBRL

4.1 Introduction

In essence XBRL is a global standard for data definition and data exchange based on XML. Key in XBRL is the very powerful, complete and advanced architecture. The core products created in XBRL are:

- Taxonomies - the meta data describing reporting rules
- Instance documents - the physical report in XBRL format

A taxonomy defines:

- Reporting data elements (pieces of information)
- Human language descriptions of the data elements
- Reference to the shared understanding or authoritative definition of the data elements
- Relationships between these data elements
- Business rules for validation across different data elements.

A taxonomy can to a certain extent be regarded as a Data Dictionary or as the collection of meta data. It provides a description of data elements and their interrelationships. Apart from this, it also provides labels to be used in user interface programs and it can hold references to legislation, guidelines and supporting documentation with regard to the data element(s). A taxonomy is highly extensible, which means that anyone is able to add elements or relationships to a taxonomy or supply different, additional labels and or references. A taxonomy is not a replacement for a chart of accounts or reporting scheme! It allows however the global description of accounting rules and therefore creates semantic understanding and similarity, across boundaries of regulators, organizations and departments..

A number of important aspects to consider in investigating the potential of XBRL and taxonomies are the following:

1. Taxonomies describe any type of data: from numeric and monetary data to dates, Booleans, texts and any other XML data types. The X in XBRL brings extensibility so that virtually any data type can be created and added to the taxonomy.
2. Data validation: validation rules can be included in a taxonomy long with definition of data types, for instance to provide rules for min-max values, max occurs, allowed enumerated lists.

3. Aggregation rules: one can describe the aggregation rules in the taxonomy to calculate totals and subtotals.
4. Relationships: XBRL allows the definition of many interrelationship models between ranges of elements. The X in XBRL means - in this perspective - that existing hierarchies and relationships can be extended to specific needs
5. Presentation structures: for presentation purposes a taxonomy can contain the many presentation structures and hierarchies.
6. Data dependencies: dependencies between elements can be described in the taxonomy.

Important to know is that although taxonomies contain a range of complex meta data a large portion of the meta data is contained in the XBRL report or instance document itself. Understanding taxonomies is thus just a part of the job and a thorough understanding of instance documents concepts is also required.

4.2 *Data models*

Software developers and database experts need to pay special attention to both the power and the complexity of the data modelling capabilities of XBRL. As many will know, XML basically represents a hierarchy of data definitions. An XBRL instance document in essence represents the same. However the extensive use of X-Links and additional modules of the XBRL specification allows the abstraction of complex data models from the data itself.

The beauty of this is that XBRL supports different data structures and allows the complete combination and simultaneous use of several data structures in one go. But let's first see what kind of data structures can be defined in XBRL.

- Flat file
 - one can make simple use of XBRL and describe a straight forward list of data in a flat file type approach. This is analogous of the presentation of a csv file with element names and values for each column of data.

- Hierarchy
 - The data can also be modelled and structured in a tree structure hierarchy by making use of XBRL presentation, calculation and/or definition features. Hierarchies can be N-levels deep.
- Cycles
 - the XBRL specification allows the use of cycles (recursive definitions) in the data model. In this case one element can be described as the parent of its own parent.
- Relational data
 - by use of tuple constructions the definition of relational data can be defined in the taxonomy. More complex structures can be defined by the use of nested tuples or relational data in relational data. XBRL allows N-levels deep nesting of tuples.
- Dimensional data
 - using an additional specification module of XBRL, XDT (XBRL Dimensional Taxonomies) any dimensional data structure can be created. For the experts: XBRL allows any granularity in any N-dimensional data structure while preventing sparse data. The dimensional constructions in XBRL can be very dynamic, purpose oriented and provide sufficient modelling capabilities for hyper cubes for business reporting.

In addition it is important to understand that not only XBRL is a language supporting many data structures but all structures can be used in one single taxonomy simultaneously. Relational data can be represented with additional multi dimensional information and displayed in a hierarchy. To keep it simple.

This increases the complexity of XBRL solutions and most of all the application integration and data mapping dramatically. This document only touches on the possible data structures and does not provide a deep technical description of them. For more information or support please refer to authors.

Business Rules are a powerful additional component of XBRL as they make it possible to define validation rules and calculations within the taxonomy. The business rules defined in the XBRL taxonomy allow for simplified application programs. Nobody in the information chain needs to build the business rules into their own (legacy) applications. For using the rules the existing applications do not have to be modified, protecting history and significantly minimizing additional investments in existing application software.

Also interesting for the software business is the fact that XBRL business rules can as easily be used to compare and calculate results from multiple financial statements, compare XBRL data with data from other sources e.g. from internet, internal databases and/or applications and can be used to conditionally create new data.

This overview document is not the place to go into further details but one needs to know that a taxonomy and the data structure contained by it can be extended in any possible direction with extension taxonomies. The data model therefore is not static by default or as fixed as a database model. Based on the data definitions and business rules defined in the taxonomy the validation of an instance document is performed by any generic XBRL processor.

4.3 *Summary*

One could say that with XBRL taxonomies, externalization of concepts from applications take place:

- Externalization of data definitions
- Externalization of data models
- Externalization of validation rules
- Externalization of business logic
- Externalization of presentation

The power and impact of this sets the expectation that there will be opponents of XBRL who will not directly see it as a benefit. Some will consider XBRL as a threat because their business is based on proprietary solutions. Not all vendors for report writers, ETL (extract transform and load) vendors, Business intelligence software and more, will see XBRL directly as an opportunity. The more strategic thinkers however will acknowledge that although XBRL will change business models, the new business models offer great opportunities. The opportunity to enhance application functionalities, to easily step into new markets as well as cheaper software development processes in complying with reporting requirements. These software vendors will embed XBRL solutions to expand their business, both from a geographical and a product portfolio perspective. Vendors can enter into markets and territories they could not reach without the use of XBRL.

5 DATA MAPPING

5.1 *Introduction*

In the previous chapter the description of the data models, the wide range of data validation rules and business rule capabilities are described. For XBRL to be used as the global data exchange language data must be collected or exported from one system and subsequently imported in an other. Producers and receivers of XBRL will be confronted with mapping concepts in an XBRL taxonomy to equivalent concepts in legacy systems and to specific XBRL applications. Thus, software developers will need to facilitate the control and/or audit of the mapping.

Again this paper is not a complete handbook for software vendors or users in implementing XBRL. It can only touch some issues. In this paper, we will concentrate mainly on some of the many aspects related to mapping. We will not cover the area of how data sources are mapped to XBRL taxonomies or the mapping from XBRL to receiving applications. But we are quite confident that the topics addressed will be able to give some insights in your integration tasks.

Like in any other information system we are dealing with data sources in XBRL applications. These data sources can be in any format, such as databases, spreadsheets, XML files and CSV files, but also manual input is regarded as a data source. We can have one or more data sources and one or more taxonomies, which may consist of a number of taxonomy extensions. We have to establish the relationship between these data sources. We need to map the original data sources to the taxonomy.

The result of the mapping process is a mapping file in which the relationship between the data elements at the source and the taxonomy is stored. The mapping file is used for the conversion process (from non-XBRL to XBRL and from XBRL to non-XBRL) and the instance document generation process. In theory mapping is not a difficult task. Using the right tools it is fairly straightforward for an end user to define the mapping structure. In practice however the mapping process needs to be managed with great care. A correct mapping file is essential for the quality of the submitted data.

Mapping requires a significant knowledge and understanding of the business and information requirements. To establish a data and subsequent mapping structure that is both logical and capable of being maintained requires careful analysis and definition of the information requirements in terms of data elements and their structure. Part of the

information requirements are likely to be driven by local and international reporting requirements. In the case of the Dutch NTP taxonomy one needs to have a certain understanding of IFRS as the NTP taxonomy is based on the IFRS taxonomy. The NTP taxonomy however also defines many data elements that are based on the information requirements of the Dutch authorities that are not linked to IFRS requirements. This situation is similar to that in many other countries.

The mapping process will not be a one time activity. There are two dimensions that can be expected to change over time: the original data source and the taxonomy. Changes in the data source and taxonomy are driven by changes in the business and reporting requirements. As the taxonomy is likely to change over time, the mapping process requires taxonomy version control. An organization needs to be able to re-produce reports that have been submitted in the past, based on the mapping that was valid at the time. On the other hand, for comparability, an organization also might need to be able to present last year's data structured in line with this year's taxonomy. For XBRL being a standard this is not a problem, but from organizational and control perspective, this needs to be addressed.

Before using a mapping file the required level of testing and, in the case of a regulatory reporting, the audit and assurance of the mapping file needs to be considered. It is vital that the process of making public taxonomies that are likely to change regularly is carefully established to prevent users using outdated versions of the taxonomy and therefore incorrect mapping.

Below we will provide some mapping situations.

5.2 Account mapping

The most simple mapping consists of a mapping between a chart of accounts and a taxonomy. In this case an account is mapped to a taxonomy element.

11.000.200 cash -> IFRS_cash

In this simple case all accounts will be mapped similarly to taxonomy elements.

The mapping process will lead to a number of questions:

1. How to map one account to 2 or more elements, if required to do so?
2. What to do if one account needs to be mapped to more than one taxonomy?

3. Can several accounts be mapped to a single element?
4. Can an account be mapped to a aggregated element?
5. How to manage the mapping to detail elements for one report and a calculated element in an other report?
6. What to do with contextual differences between the accounts and the taxonomy - e.g. balance (debit or credit) or period definition (instant or duration)?
7. How to control the completeness of the mapping when new accounts are created?
8. What is the sign convention of the application and XBRL, are credit values stored and reported negative or positive. How are corrections represented?

Extending this case we can face the situation where debit values of an account need to be mapped to one element and credit amounts to another element. Other complexity is introduced when beginning balance values need to be mapped to one set of elements, the change values to another set of elements and the closing balance to a third set of elements.

A further case can be when an account has a positive value it needs to be mapped to one element and in case the same account has a negative value it needs to be mapped to another taxonomy element. Like: the account is *net result* and the elements are *net profit* and *net loss*.

5.3 Details or aggregated values

Depending on regulatory requirements it can be that only detailed values need to be reported. In that case not all accounts will need to be imported and thus mapped. In another situation a regulator might require only aggregated values. Then the values of the detailed accounts need to be omitted in the mapping. And this is just the first step of account to element mapping.

5.4 Data mapping solutions

As experienced in many XBRL projects one can see that real life proves that more complex mapping issues arise. It can be that data modification is needed before data can be processed for importing.

When a chart of accounts contains certain hierarchies, or contain cost centre codes it will be needed to transform codes before processing. E.g. the characters that represent the cost centre need to be removed before mapping takes place. This may also be

needed to keep the mapping less prone to changes for each change in cost centre structure.

Also it can be needed to transform the values itself. This is the case when amounts must be reported in thousands or millions. Or when one needs to report in another currency than the accounting system registers. These situations need to be handled in a process before the actual XBRL conversion, but can be part of an XBRL reporting application.

5.5 *Other data mapping issues*

In addition to account mapping, the mapping of other data sources can bring many more issues to the table. Mapping of data from different sources containing non financial data can bring other mapping issues. To mention just some of them:

- mapping of strings,
- source strings can be longer than allowed in the taxonomy,
- string values need to be transformed,
- the notation of dates is different,
- Yes/No fields need to be transformed to True or False,
- code lists need to be converted to allowed enumerated values in XBRL etc.

5.6 *Relational data mapping*

When source data is relational or when the taxonomy makes use of relational structures with tuples the issue of tuple import needs to be considered. In tuple import the source data need to be imported and thus mapped to the tuple structures. And as we know XBRL allows unlimited nesting of tuples. When the source data is relational by nature mapping is quite possible but if the source data is organized differently than is contemplated by the taxonomy it may require more sophisticated mapping.

In tuple mapping issues will arise in relation to field mapping, joins, relational integrity, data base consistency and all the other familiar concepts we have been working on for the last decades in the database environment. Nested tuple import in XBRL also requires import control for constrains on the tuple structures. Constrains like max occurs need to be followed.

As in account mapping in tuple mapping data transformations or modifications can be required.

5.7 *Dimensional mapping*

XBRL is N-dimensional and allows any granularity. This means that dimensional import and mapping needs to be supported. Dimensional mapping defines a number of dimensions related to a certain element value, e.g. revenue by product, region, customer segment and business unit. In XBRL each element can have a different dimensional structure, also called a “hypercube”. Dimensional structures in XBRL are also highly extensible. So if the mapping for one regulatory demand is not sufficient for dimensional mapping to another, the dimension structure can be expanded.

In XBRL Dimensional Taxonomies (XDT) the allowable set of imported data is defined. These can and must be used in any dimensional import mechanism.

5.8 *XBRL GL*

The XBRL consortium delivered a specific taxonomy for transactions described in XBRL format. This is called XBRL GL (Global Ledger). Where XBRL for reporting (or FR- Financial Reporting) is targeted at summarized data, XBRL GL is about exchanging data at the detailed transactional level.

For mapping and importing this brings a new set of requirements and functionality which will not be described further in this paper.

5.9 *Summary*

XBRL has sophisticated data modelling capabilities. It allows simultaneous use of different data models.

Import tasks for XBRL range from simple mapping for account-to-element to complex mapping of combined relational-dimensional data imports. In importing processes it can be necessary to execute data transformation to make the data usable for the specific regulatory reporting requirement. The import processes need to be accompanied by sufficient control over the mappings as well as control on the import of the data itself. Some controls can be embedded in the mapping file, some are executed by default via the data definitions in taxonomies and some will be coded in software applications. It is required that the software you plan to develop is capable of dealing with all of the issues mentioned above.

If you choose to buy XBRL software it should be one of your selection knock out criteria.

6 INSTANCE DOCUMENTS

6.1 *Instance document generation*

An instance document is in itself nothing more than a file with facts and contexts. A fact describes the value of, for instance, the data element “Cash and cash equivalents” and the context tells us all we need to know about the identification of the reporting entity, the reporting period, the currency used, etc. Examples of instance documents are a tax filing, a statistical report, an annual report, a project report or a departmental budget. Again, these are just business information sets we are already familiar with in many shapes and forms.

An instance document may contain all information that has been defined as either optional or mandatory for that particular reporting obligation. All reporting data that make up an instance document need to be defined in one or more related taxonomies. For example the Dutch NTP taxonomy contains roughly 10.000 data elements, but a VAT declaration contains only about 30 of the data elements within them.

During the process of generating an instance document three components will be brought together:

- the taxonomy
- the mapping, and
- the original source data

The original source data is converted to XBRL and the result of the process is a XBRL instance document.

It may be required that an instance document has to comply with the Financial Reporting Instance Standard (FRIS) that describes how an instance document should be structured. A FRIS rule for example prohibits the presentation of two identical contexts, so one can not represent the opening balance two times.

Note that a large portion of XBRL's meta data and complexity is not in the taxonomy but in the instance document. So software developers should take care to understand instance documents as well.

6.2 *Instance document validation*

After being generated, the XBRL instance document has to be validated. Validation of an instance document can take place at two levels:

- Validation of the structure
- Validation of the content

Validation of the structure is done by using general validation rules that validate whether an instance document complies with the standards defined in the XML specification, as defined in XBRL and possibly in FRIS. A far more powerful use is the ability to validate instances where a taxonomy contains business rules. These business rules, defined in the XBRL formula link base of a taxonomy, allow for a validation of the content of an instance document.

Business rules can also be used for exception reporting, signalling that a certain value varies more than x % compared to the previous reporting period. The power of business rules is that these rules can be defined as part of the taxonomy. This means that neither the submitter nor the receiver of data needs to build these validations into their own software. Using the taxonomy, true generic XBRL processors will take care of the business rule calculations.

The result of the validation process is a signal that the validation was either successful or not successful. If not successful, the results of the validation process need to be picked up by the application that called the XBRL processor and correction functionality needs to be presented to the end user or, in cases of fully automated processes, the process needs to be correctly ended in order to allow for corrective actions in the legacy application.

A significant advantage of using the taxonomy for defining business rules is that the submitter can validate the message (instance document) before sending the message to the receiver. The sooner in the process an error is recognised; the more cost effective the error can be solved. The message receiver can use the same business rules. Although this may seem a redundant process, the submitter has already validated the message, the receiver will want to make 100% sure that the received message is valid and it will take a XBRL processor only a minimal amount of time to perform the business rules validation.

The receiver of a message can also define additional business rules for further validation of the content. An example might be the tax authorities, who will initially

validate a message based on a public taxonomy, but extend this validation by using their taxonomy extension in which additional only internally used business rules are defined.

7 TAXONOMY RELATED PROCESSES

7.1 *Introduction*

This document doesn't cover all aspects of taxonomy development. The reason is that we expect that in first instance the readers will focus on using the existing taxonomies provided by the different authorities. It is expected that taxonomy development will be the playing field of a certain group of taxonomy experts that will have and maintain the knowledge and experience to develop new and extend existing taxonomies at a level of quality that is required.

Does this mean that the "X" in XBRL has only minimal additional value? The answer is: Not at all!

It is just that we don't expect many organizations at this moment in time to invest a lot time in money to get people trained to the level that it can be expected that they develop taxonomies or taxonomy extensions that have the right quality and sustainability.

In this report we will highlight a number issues that come with taxonomy development and maintenance.

Taxonomy management consists of a number of related processes, such as:

- the development of a taxonomy;
- the maintenance of a taxonomy (release management / versioning);
- the publication to the user community.

7.2 *Taxonomy development*

Someone will have to design and create a taxonomy. It is as simple as that. In general, developing a taxonomy is similar to what you are used to when you develop any other information system. You analyse and define the information requirements in terms of data elements and their structure. The additional effort is to understand the additional capabilities and functionality an XBRL taxonomy can bring to software vendors. As with any data modelling approach one can design a simple or more complex architecture. One can make an unfriendly design or a user oriented one. And like other key technologies the XBRL world is going through the same learning curve in what is good and what is bad practice.

Today one can see more straight forward monolithic taxonomy designs as well as sophisticated modular architectures, taxonomies making maximum use of the power of XBRL's modelling capabilities as well as limited ones.

The Dutch taxonomy architecture can be considered as the most sophisticated, modular and easiest to maintain taxonomy in the world. It looks like that this well designed architecture will also be adopted by other governmental projects around the world. The Dutch taxonomy architecture is federated, highly modular, allows easy maintenance by many agencies and reduces the need for end users to understand or even see the complete complexity of XBRL. The architecture provide different reusable layers for ease of maintenance.

After the complete information requirements have been defined, it is therefore advisable to consider taxonomy design options. A good understanding of the do's and don'ts is essential. One should make use of the expertise that has been developed over the last few years in successfully implemented projects in order to avoid unnecessary pitfalls. A taxonomy that is logically structured with an adequate number of layers and dimensions will allow for easier maintenance.

This information analysis and preparation work can be done using the tools that everybody is used to. The result of all this preparation is a set of reporting data definitions and their relationships that can be imported in a taxonomy development tool. The same software can also be used for validation the actual taxonomy including all reporting meta data.

In addition to existing taxonomies, XBRL allows for building extended taxonomies. The same taxonomy development activities need to take place if people decide to develop their own extensions to an existing taxonomy.

Special consideration has to be given to the process of testing the taxonomy. Before publishing a taxonomy every effort has to be made to make sure the taxonomy is robust. Normal practice in the XBRL consortium is to use the taxonomy to produce a representative sample of real live business reports based on the newly created taxonomy.

7.3 ***Publishing the taxonomy***

After developing the taxonomy, the taxonomy needs to be made available to the end users. A taxonomy can be a "public" taxonomy, like the IFRS, US-GAAP or Dutch NTP

taxonomies, but it can also be a private taxonomy, for instance for use within an organization for internal reporting purpose or be the property of an accounting company or institute. A taxonomy can be made available though the internet, as a web service or downloaded onto an internal server or workstation. XML, the fundamental language of XBRL, provides sufficient mechanisms for publishing taxonomies on the internet.

When using an XBRL application end users need to have the facilities to decide which taxonomy they want (or need) to use for reporting (version control). For example different versions of a taxonomy may be used for accounting rules from various years. There has to be version control functionality within an application to manage the storage and usage of more than one taxonomy.

7.4 Taxonomy maintenance

A taxonomy needs to be maintained. In an increasingly dynamic world information requirements will change over time and thus will the taxonomy. There need to be procedures in place that describe the tasks, roles and responsibilities of everybody involved in taxonomy development and maintenance. Especially in “public” taxonomies, used by large numbers of users, it is essential that the quality of the published taxonomy is outstanding.

Taxonomy developers need to have a managed and controlled environment in which they are able to continuously work on newer versions of the taxonomy without affecting other versions. When people make use of extensions to taxonomies developed and maintained elsewhere, it is good business practice to organize the maintenance schedules of the taxonomy extension in line with the maintenance schedules of the core taxonomy.

A new version of a taxonomy has to go through all steps of the original development cycle: analysis, design, build, test, train, make available and support. Just like any other part of an information system.

8 INFORMATION SYSTEMS INVOLVED

There are various information systems and applications that may be impacted by the introduction of XBRL. These applications are impacted by XBRL as they either provide output that has to be converted to XBRL format, either directly or at a later stage in the information processing chain, for further processing (i.e. by a report generator or tax assessment application) or use input in XBRL format that has to be transformed into the application's proprietary format for further processing.

8.1 *Accounting systems*

With respect to accounting systems the following options are available:

- An organization has its own accounting system
- The accounting is done by an external accountant, who has its own accounting system
- The organization and the accountant share the same accounting system in an ASP model

The accounting system holds much of the information required in the information process. Because the NTP taxonomy covers a number of messages which are currently expected to be submitted electronically, it is likely that the accounting systems have all of the data elements available. If new data elements are introduced the software supplier has the option either to amend the database or to allow for manual input of the additionally required elements.

It is possible to introduce XBRL and taxonomies into the accounting system. Software vendors can develop the feature to map account codes to a taxonomy. This introduces XBRL very early into the information supply chain which optimizes the use of XBRL.

8.2 *Report generators*

In the Netherlands a small number of companies is involved in the development of report generators and more specifically report generators for Annual Reports. All of these products work with a proprietary reporting structure. These vendors may have two options, one is to adopt the NTP taxonomy as the reporting structure, or, as a short term alternative, map their own reporting structure to the NTP taxonomy. Over 98% of these report generators are used by intermediaries (accountants/administrators).

8.3 *Fiscal/Tax solutions and Salary/Wage solutions*

We have identified a small number of software vendors that support the preparation of tax declarations (Income and Corporate tax). In a number of cases these applications are built on old technology and all of them have (again) their own structure for organizing data.

Over the last few years interfaces have been developed between the report generators and these fiscal tools. In some cases there are interfaces between accounting systems and these tools. So, we are dealing with a lot of mapping between different structures.

8.4 *Accountancy solutions*

Accounting firms are more and more using work flow and CRM solutions to support their own business processes. These solutions also need to be XBRL-ised, because much of the data needed in (especially) the Annual Report generation process is registered in these systems.

The taxonomy provides a very interesting instrument for the audit process. Having all data in the same structured way, independent from the financial application from which it originates, is very useful for optimising the accounting practice and to ensure that the company meets compliance rules.

8.5 *Information flow examples*

Combining the various 'configurations' of software applications and XBRL infrastructure options result in a continuum of architecture options that may range from:

- On one end of the spectrum an organization that uses its own accounting solution for the preparation of a VAT declaration, that generates and validates the appropriate instance document directly out of its accounting system, that "signs" the instance documents and submits it to the "OTP SOAP 2008", and takes care of the submission follow up.

In this case there can be no confusion about the responsibilities. The organization is responsible for the content of the instance document and for the fact that the message has actually been sent, delivered and accepted.

- On the other end a company that uses its own accounting system, which provides data to the Annual Report generator. This finance (and other) data is used by an

accountant for auditing the data and also for the preparation of the Annual Report. A version of the Annual Report needs to be submitted to the Chamber of Commerce, but the actual submission is left over to a service provider, established as an intermediary between the enterprise/accountant and the OTP SOAP 2008. The role this service provider plays can have many shapes and forms, but let's assume that they have made an arrangement with the OTP SOAP 2008 with regard to security and authorization. They are a "certified" submitter. In this situation we have many responsibilities spread over a number of different parties.

The relationship between the enterprise and the accountant should be absolutely clear, which is (probably) already the case, because XBRL introduces no changes in this area. The relationship between the enterprise/accountant and the service provider needs to be defined, described and agreed in the form of a Service Level Agreement. The use of digital signatures and security certificates needs to be formalized.

9 SOFTWARE DEVELOPMENT STRATEGIES

9.1 *Introduction*

A very special and critical role in the reporting chain is the role of the software vendor. For producing, validating, receiving and analyzing financial and XBRL data appropriate software is needed.

In the previous chapters we have described the XBRL related functional components of information. The XBRL functional components are used with information generated by legacy financial applications. In this chapter we describe some of the applications most likely to generate XBRL information.

Software vendors are preparing to incorporate XBRL output, XBRL input as well as XBRL data management capabilities. The focus in the paper however is on XBRL output. This is more than just a 'save as XBRL' function. In order to serve the users well, applications need to be able to provide a host of functionality. They should manage applicable taxonomies, provide insight in these taxonomies and provide facilities to enhance taxonomies. These enhancements are called extensions and are necessary to communicate facts that are not defined in a taxonomy provided.

Next the application should be able to map taxonomy elements to data managed in the application or stored in its database. The application will be required to generate a report and facilitate editing, reviewing, auditing, verification and approval. Next the report should be sent out to external or internal stakeholders.

It is an illusion to think that any business software application will only need to be able to handle one single taxonomy. A user always has to deal with a number of taxonomies and taxonomy extensions. Even if a user would only report according to IFRS, he would still need to handle versions and components of the applicable taxonomy and company extensions to that. The notion that a specific taxonomy, such as IFRS, can be 'built into' an application in order to serve the needs of a user, has proven to be incorrect.

Preferably all of this will be automated to reduce the effort of reporting to a simple push of a button. Large corporations can afford a large accounting and control department and may not be very demanding regarding efficiency. However, their subsidiaries and the millions of smaller companies and institutions around the globe will want to see highly effective and cost efficient tools, supporting specific daily reporting tasks.

All functionality of the environment needs to be accessible from other applications and executed automatically through scripting. Vital for adoption and successful deployment of XBRL is the ability to hide complexity and provide an environment that is friendly to casual users. After all most reports are produced only once a year or once per quarter.

Which infrastructures and hence products and services software vendors can and will develop is to a large extent dependent upon the actions of individual software providers and system integrators.

9.2 ***What options are available to software vendors?***

Vendors that chose to treat XBRL as yet another export format, find themselves struggling with their business model. Users do not want to pay for an extra 'save as' format and expect that to be provided under maintenance.

Vendors that are successful in selling XBRL functionality carry an added value proposition. They clearly communicate the current and future value of the provided functionality and define their pricing in accordance with that. This implicates they can ask a decent price for an additional module that provides carefree compliance to reporting requirements at high efficiency rates. If the volumes are considerable, XBRL functionality is sold as a 'no brainer' option at prices equal to a certain percentage of the standard license price of the financial application itself.

In essence individual software vendors have a number of options when considering the impact and implications of XBRL:

- Do nothing;
- Develop XBRL functionality yourself;
- Buy XBRL components and embed them into your own software;
- Provide desktop XBRL functionality as a third party solution;
- Develop integration with a service provider's XBRL capabilities.

"Do nothing" seems to be the worst decision possible. In the foreseeable future XBRL will be the standard for business reporting especially for those who have to report to regulatory authorities (and isn't this the case for all of us?). Even if you think your only reason for doing nothing is the lack of mandatory XBRL reporting, induced by government, you're thinking the wrong way. There is no profitable business case in mandatory (use of) standards, profits will come from the added value software vendors

generate for their customers, by re-use of and simple functionality and decreased costs of software development.

The only real option is to provide XBRL capability. And then the big question pops up:

do you **MAKE** or **BUY** or **USE** XBRL functionality.

Either the software vendor builds the XBRL components/functionality from scratch or the software vendor buys the XBRL components from a XBRL component supplier. The third option is to use parties that provide XBRL capabilities for generating and validating instances as a (web) service.

The make or buy decision is dependant on many criteria, but it is important to stress that the tools available on the market today have taken many years of development and huge investments. Short term rushed and under funded 'make' efforts have a high risk of failure.

9.3 *The MAKE option*

When an software vendor decides to develop an XBRL solution in house, it will bring core XBRL knowledge into the development team, the solution can be more tightly integrated and dependency of thirds parties is minimal. On the other hand it means that a team has to be formed and appropriate XBRL knowledge has to be gained and maintained. This requires a serious investment. XBRL is nothing more than a complex standard with 500-800 pages of specification, best practices, validation rules and more. Together with Financial Reporting Taxonomy Architecture (FRTA) and Financial Reporting Instance Standards (FRIS) the complete specification comes to about 1.200 pages!

Also to consider is that the standardization organization XBRL International is still adding more standards to the set, as such increasing the amount of material. The additions to the current versions of the standard, that are expected in the near future (like dimensions, business rules and versioning) will make the XBRL standard a lot more complex. This puts additional pressure on your development team.

If XBRL is not part of the core business proposition it can be considered a too expensive exercise deflecting the attention from the main solution.

9.4 *The BUY option*

9.4.1 *Integration of XBRL components*

One of the alternatives within the BUY option is to incorporate an XBRL engine in the architecture of your application software. This allows for a more integrated XBRL functionality whereas XBRL reports can be generated from the user interface of the underlying application. The vendor can choose for various levels of sophistication regarding the functionality. It can create specific solutions as well as generic tooling for XBRL reporting. This approach guarantees readiness for all future developments in XBRL. Developments in the language can be accommodated by updating the engine; new reporting developments can be facilitated with new applications based on that engine.

XBRL Processor components can be easily integrated into existing solutions. They are available from XBRL software vendors as .DLL, .EXE, .COM objects.

Special attention has to be drawn to the fact that these components have to be fully XBRL compliant, but also have to comply to the Financial Report Instance Standard (FRIS) that is defined by the authority that is owner of the (part of) taxonomy. It is necessary to check with your XBRL supplier to what level they support NTP FRIS.

9.4.2 *XBRL Desktop solutions*

The other alternative would be to develop a user friendly integration with an external XBRL reporting solution. This solution can either reside on the desktop of an end user or run on a internal server. These solutions are on the market today. The software vendors can provide this functionality to their customers as part of their overall proposition based on a reseller and/or partner agreement.

9.5 *The USE option*

If software vendors opt for the USE option there are two possibilities.

9.5.1 *Proprietary web service*

The software vendors can develop an interface to a “proprietary” (web) service that handles all XBRL related functionality. This way XBRL functionality is completely hiding

for the end user. The software vendor has complete control over the user interaction and the way he wants to present the result of the instance generation process.

Software vendor's (accounting solutions, report generators, fiscal tools) might consider setting up their own web service for XBRL based functionality. This allows them to control the interface between their own application and changing taxonomies. The taxonomies change because of (constant) changes in information requirements.

An interface needs to be developed to be able to submit the required data in any shape or form to the web service. Successful communication requires an upfront definition of the data sources and completion of the mapping of the data source to the taxonomy. After the data source submitted data the web service takes care of the conversion to XBRL, the validation of the data set and (as an option) the submission of the instance document to the information receiver. The web service will provide feedback on the results of the validation and submission process.

The advantage of the interfacing with a proprietary web service solution is the fact that software development is minimized, that no extensive knowledge of XBRL is required and that it offers a very flexible application architecture.

9.5.2 *External open web service*

An alternative to this solution can be a user friendly integration to an external "open" (web) service that handles all XBRL related functionality. This alternative can also be used to add functionality to the current (financial) application. For instance for the generation of Annual Reports.

For example:

To generate an annual report based on the financial data that is stored in a financial management system a customer can (directly or indirectly) use the services of a service provider. The preparation of the annual report requires additional data input and storage facilities that (at this given moment) may not be provided by the software vendor (examples: share holders information, explanations and disclosures). With indirect use is meant that report generation functionality is supplied by service provider but branded as part of the overall solution of the software supplier.

The end user may not notice that part of the XBRL functionality is included in the legacy software applications whereas another parts of the XBRL functionality is provided by a service provider.

9.6 *What is the best option for your organization?*

Software vendors have to be aware of the fact that professional handling of XBRL and taxonomies requires not only technical IT knowledge, but also managerial, work flow and business knowledge. The introduction of XBRL is more than a technical event. Pioneer companies can help you use XBRL to your strategic advantage. If your systems capabilities of handling XBRL are superior, your customer will regard your solution as strategic.

You can reach that position quickly and easily by using the unique XBRL capabilities, as many vendors chose to do already. These digital reporting solutions allow you to help your customer meet reporting requirements and be compliant quickly and at virtually no risk. Vendor solutions for XBRL reporting can be entirely integrated in software architectures and offer all functionality as described in this paper, and more.

Depending on the nature of an application or custom software solution, one might want to add value by the capability to receive and process XBRL. Vendors of software for financial consolidation, business intelligence, risk management and various others have decided to use available technology to remain on the leading edge. They now are capable of effectively using information available in XBRL, either from internal or external sources, to create a dynamic information environment that is open and flexible.

10 ROAD MAP

10.1 *Introduction*

Regardless of the choice on make, buy or use your XBRL functionality a valid approach is required for the development of XBRL capability into the software vendor's solution.

A Road Map to consider is defined below:

- Develop XBRL Publishing capabilities (generation of instance documents)
- Develop XBRL import functionalities
- Adapt XBRL GL for transactional data exchange
- Adopt the semantic and data modeling concepts in the application

10.2 *Develop XBRL Publishing capabilities*

Development of publishing capabilities allows each software vendor to be part of the XBRL community quickly, easy and at minimal investment cost. Here the company decides to make use of a XBRL product to provide XBRL export options. The main activity for the company will be to connect and integrate the XBRL component. Minimal XBRL knowledge is needed.

When you are still struggling with the make or buy decision, we suggest that you set up a pilot project with one of the XBRL software vendors. This way you will provide your staff and decision makers with practical knowledge and experience about what it means to "XBRLize" your software. This approach will proof to be a small investment compared to the costs of making a wrong decision.

10.3 *Develop XBRL import capabilities*

XBRL import functionality means that the vendor can more easily play a more central role in business reporting chains, collecting XBRL data, convert it into the application data model and storing it in the database.

10.4 *Adapt XBRL GL for transactional data exchange*

XBRL GL brings new business opportunities and cost savings for software vendors and customers. Sharing business transactions between applications using agreed semantics and format dramatically reduces the costs of application integration.

This allows the software application to interact more easily with other applications and reduces development and maintenance costs in maintaining many import and export formats.

10.5 *Adopt the semantic and data modeling concepts in the application*

The most beneficial approach for many vendors is to consider the highly strategic choice to embed the complete well thought through semantic data modeling concepts and rich financial meta data concepts into the core of proprietary application. It enables applications instantly to make maximum use of many possibilities that XBRL brings to the market.

XBRL offers many advantages to be used instantly to deliver smart applications for the future.

11 SUMMARY

In this document we have tried to present you our ideas and suggestions to allow you to make informed and deliberate decisions with regard to your XBRL strategy and implementations. We have argued why XBRL in itself doesn't provide a complete solution. XBRL is "just" a standard and technology which, when used properly, can help you to improve your information process, to safeguard your investments in current applications and significantly reduce development effort and costs for new applications.

XBRL is a fairly new standard that still has to prove itself. A number of very successful implementations have proved that XBRL based solutions to be capable of achieving objectives and meeting expectations and ambitions set by the information reporting community.

The Dutch government is pushing XBRL through the NTP project and by establishing the platform for digital communication, OTP SOAP 2008. By providing a XBRL reporting platform the government expects to decrease the administrative burden for businesses. This only will be effective in tight cooperation with software vendors who implement XBRL in their financial and reporting software.

Confronted with XBRL and the likely request for XBRL solutions by customers each software vendor has to come up with its own XBRL strategy. Based on the emerging infrastructures and the software vendor's current product and service portfolio, each software vendor will have to define relevant XBRL product market combinations and define its role in these respective PMC's. This analysis not only requires a short term market outlook, but also asks for a mid and longer term strategy and vision.

As a software vendor you have to consciously position yourself in the market, on the one hand for developing the short term XBRL solutions based on current governmental and market initiatives and on the other hand for possible fundamental, XBRL induced shifts in the professional and financial service industries.

We feel that for each software vendor XBRL may be a (manageable) threat, for proactive and visionary players it is definitely an unprecedented opportunity, not only in terms of potential new applications and markets but also for future programming and maintenance activities.

As stated in the introduction of this paper software vendors have to answer the following questions:

- Do we invest in this new XBRL standard?
- If so, for which applications do we want or need to use XBRL?
- For the identified applications, how much time, resources and capital are required and available?
- Do we make or buy the necessary XBRL (enabling) technology?
- Do we build XBRL components into our (legacy) applications or do we make use of a web based XBRL service provider for XBRL functionality?
- Do we differentiate between our short and mid term XBRL strategy, buying (service provider based) functionality first and developing our own XBRL enabled applications later?

Regardless of the answers to these questions providers of XBRL functionality have developed sufficient XBRL knowledge, components and server based solutions to cater for any Dutch market demand. This experience should be applicable to any other market following a similar path to The Netherlands.

XBRL Netherlands can provide software vendors amongst others with:

- Strategic and operational XBRL consultancy
- Taxonomy management services, including development of business rules
- Mapping tools
- XBRL components
- Web service provider functionality

We hope this paper provides software vendors with food for thought for a deliberate preparation of your XBRL enabled software solutions and infrastructure.

If not actively developing or pursuing an XBRL strategy, you should come up with a contingency plan that allows you to catch up with competition as soon as XBRL has established itself in the marketplace. Without this plan and not being able to generate and process XBRL messages you may see yourselves confronted with a decrease in your installed customer base and loss of market share.

If you want to be able to generate XBRL messages with your software applications you have to plan how to integrate XBRL functionality into them. XBRL Netherlands is more than happy to provide you with all the XBRL related services and products required to assist with the planning and implementation of XBRL solutions.

