Available for free online at https://ojs.hh.se/

Journal of Intelligence Studies in Business 1 (2011) 19-28

Using XBRL Technology to Extract Competitive Information from Financial Statements

Dominik Ditter *, Klaus Henselmann * and Elisabeth Scherr *

* Department of Accounting and Auditing, Friedrich-Alexander University Erlangen-Nuremberg, Lange Gasse 20, D-90403 Nuremberg, Germany

dominik.ditter@wiso.uni-erlangen.de, pruefungswesen@wiso.uni-erlangen.de, elisabeth.scherr@wiso.uni-erlangen.de

Received 1 June 2011; received in revised form 22 November 2011; accepted 26 December 2011

ABSTRACT: The eXtensible Business Reporting Language, or XBRL, is a reporting format for the automatic and electronic exchange of business and financial data. In XBRL every single reported fact is marked with a unique tag, enabling a full computer-based readout of financial data. It has the potential to improve the collection and analysis of financial data for Competitive Intelligence (e.g., the profiling of publicly available financial statements). The article describes how easily information from XBRL reports can be extracted.

Keywords: Financial Intelligence, XBRL, Competitive Intelligence, real-time Business Intelligence

1. Introduction

Competitive Intelligence (CI) can be defined as the process of "*gathering and analyzing information about your competitors' activities and general business trends to further your own company's goals*" (Kahaner 1998, 16). Important sources of competitive information are publicly available financial statements. They provide a lot of valuable information about competitors as their financial performance (e.g., for the calculation of financial key metrics to measure the profitability of competitors) and financial position (e.g., for the evaluation of the capability to survive price wars). However, this information usually cannot be used to the full extent.

The established format types of published financial statements, for example MS Excel, MS Word and Adobe PDF, are unstructured and therefore not computer readable. Software programs simply do not know how to use this information. With no information for further working, data processing systems interpret the information as on-going text. Every item (in approximately 100 to 500 pages) must be manually fed into an analysis software tool or database system. The effort it takes to manually extract the required information from financial statements is time-consuming and error-prone. For this reason, CI managers are forced to acquire adjusted or structured financial data from intermediaries or business data providers. The disadvantages of this approach are the high costs that are incurred and the fact that the data is not obtained directly from the source (i.e., the target company).

The *eXtensible Business Reporting Language*, or XBRL, has the potential to solve these problems. Information within documents that are provided in the XBRL format enable automatic data processing of almost all reported items without time-consuming manual feed of data. The idea behind is that companies have to publish their business reports in a standardized electronic structure, increasing the transparency of the reports for investors. With a little programming effort, everyone (including small investors) can access financial data directly from the source, at low cost and almost in real-time. As a side effect, XBRL also offers opportunities for CI.

Today only a small amount of specific literature for accessing XBRL data is available (except Hoffman 2006). This article is based on Hoffman 2006 and describes how information from XBRL data can be extracted and used for CI. The article shall serve as a technical guide and outlines how to get started and what instruments are required.

This article proceeds as follows. First, we give a short explanation of the XBRL Concept, before explaining an approach to XBRL data extraction in Section 3. In this example, specific financial line items of an actual XBRL document will be extracted. Because the implementation status of XBRL is very sophisticated in the U.S., all explanations for extracting and using XBRL are based on SEC filings. The article closes with a discussion of the effects of XBRL on the development of CI and a short summary.

2. The XBRL Concept

Because XBRL is a derivate of XML technologies, the fundamentals of XML will be illustrated first, followed by an introduction to the fundamentals of the XBRL Concept.

2.1 Fundamentals of XML

The *eXtensible Markup Language*, or XML, is a meta-language for the creation of a self-defined document markup language (Watt 2002, 10).

A popular markup language is the *HyperText Markup Language*, or HTML. With HTML, it is possible to assign a specific look or layout to document content. Therefore, the text or numbers of an HTML-formatted financial statement (file extension **.html)* are tagged (marked) by specific expressions. For example, HTML tags can indicate that the number "*14013000000*" is to be displayed in bold letters or in the color green. This enables computer programs like Mozilla Firefox or Microsoft Internet Explorer to interpret and present the document content in the deposited layout. The World Wide Web Consortium (W3C) lists the applicable markups (vocabulary) and logical structure (grammar) for the creation of an HTML document in the HTML Specification (W3C Recommendation, 1999).

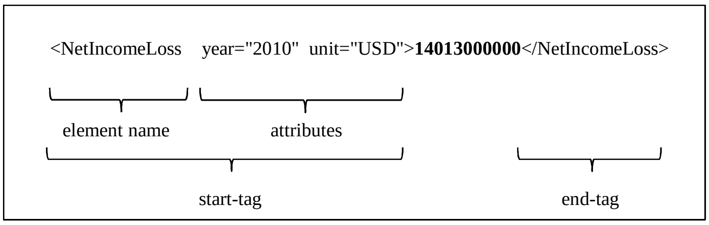

<NetIncomeLoss year="2010" unit="USD">**14013000000**</NetIncomeLoss>
element name
attributes
start-tag
end-tag

Figure 1: Logical structure of markups

XML is similar to HTML in the way it uses tags. However, XML markups define the meaning of document content. For example, in Figure 1 the number *"14013000000"* is encircled by two tags indicating the start and the end of the markup. These tags tell us that the number reported is the net income of a company (and not its turnover, assets, etc). Further, we can see that it is the net income for the year 2010 and that it is measured in US dollars (not euros or pounds). Using this information, a suitable computer program could open the file, read the number and do any computations with it. No human beings are needed to retype the numbers on a keyboard. Access to the data is much faster and less error-prone.

In contrast to HTML, XML is a meta-language. Therefore, the W3C does not regulate the vocabulary, but a set of grammatical rules for creating self-defined computer readable markups (Watt 2002, 10). The name and order of elements and attributes used for the creation of markups can be arbitrarily extended. The XML Specification ensures that XML markups, which consist of a logical structure of elements, attributes and values, are well-formed (W3C Recommendation, 2008). Figure 2 illustrates two simplified examples of well-formed XML documents with the description of identical data content. The examples A and B (see Figure 2) show that there are different possibilities to describe the same issue according to

```
<?xml version="1.0" encoding="US-ASCII" ?>

<company   xmlns="http://www.apple.com/instance/ExampleA"
           schemaLocation="ExampleA.xsd">


  <Apple>
      <NetIncomeLoss    year="2010" unit="USD">14013000000</NetIncomeLoss>
      <NetIncomeLoss    year="2009" unit="USD">8235000000</NetIncomeLoss>
  </Apple>



</company>
```

```
<?xml version="1.0" encoding="US-ASCII" ?>

<company   xmlns="http://www.apple.com/instance/ExampleB"
           schemaLocation="ExampleB.xsd">


  <Apple    year="2010">
      <NetIncomeLoss    unit="USD">14013000000</NetIncomeLoss>
  </Apple>
  <Apple    year="2009">
      <NetIncomeLoss    unit="USD">8235000000</NetIncomeLoss>
  </Apple>
</company>
```

Figure 2: Two well-formed XML documents

XML rules. For XML documents to be automatically exchangeable between different senders and receivers, all participants of the communication chain have to use the same XML markup language. In other words, a uniform and consistent markup of document content must be ensured. Therefore, the usable element names and attributes have to be predefined and deposited in a schema file. Today it is common to do this with the *XML Schema* language (van der Vlist 2002, 2). An XML document is "valid" if the markups conform to the rules of the corresponding schema file. Therefore, this XML document is an instantiation of the schema or a so-called *instance document* (Binstock 2003, 12). XML instance documents (a technical term for a valid document with data content; file extension **.xml*) and *schema documents* (documents in which the declared elements and attributes like *NetIncomeLoss* and *year* are deposited; file extension **.xsd*) are connected by the bold expressions in Figure 2. A so-called validating XML Parser (module of a software program; responsible for the reading in of an XML document) can search for the attribute *schemaLocation* (Harold 2004, 453). If this reference to the schema document is available, the Parser can check the XML document for conformity against the predefined schema. In other words, by rejecting XML documents in the event of inconsistencies or markup errors, the schema can control consistency.

Because it is possible to select a free number of self-defined elements and attributes, two different XML markup languages may use the same name for an element. For example, in Figure 2 "Apple" addresses the company Apple Incorporated. However, in another context "Apple" may mean a kind of fruit. To ensure a clear unique classification, this name conflict can be solved with *XML Namespace* (W3 Schools, 2011). A namespace is an inventory of affiliated elements and attributes that can be identified with a unique name. The namespace name must be an URI (Uniform Resource Identifier) (W3C Recommendation, 2009). Because URIs are absolutely unique, it is not possible for the same URI to exist again. A default namespace is defined in the start tag of the root element by the attribute *xmlns = "URI"* (Evjen 2007, 29). It thereby applies to all other elements that are reported within the document. In the upper example of Figure 2 the elements *Apple* and *NetIncomeLoss* are associated with the namespace *"http://www.apple.com /instance/ExampleA*".

Via the creation of individual, customized tags, XML is a very flexible standard for electronic data exchange. Almost all transmission requirements and communication constellations can be covered with XML. However, for the structured exchange of business information a certain recognized framework has been established: XBRL.

2.2 Fundamentals of XBRL

XBRL was created for the automatic and electronic exchange of business data. The non-profit organisation XBRL International Incorporated (XII) maintains the standard in an own specification. XBRL is a meta-language for creating markup languages for business reporting issues. But in contrast to XML, the XBRL Specification provides both the grammar and core vocabulary (XBRL International Incorporated 2008, 2).

The XBRL syntax is based on several open, globally accepted standard specifications, including XML, XML Schema, XML Namespace, XLink and XPointer. The repertoire of XML technologies selected for XBRL is compiled in the XBRL Specification (SEC Release 33-9002 2009, 11). Furthermore, the XBRL Specification outlines elements and attributes used to define reporting elements and to express relationships among them. Therefore, the unity open-standard XBRL can be understood more precisely as the *“core language”* for the creation of markup languages for business reporting issues. However, with XBRL it is possible to create not only a markup language, rather more a classification system (taxonomy) (Hoffman 2010, 301).

There are many different types of accounting standards around the world, for example, IFRS (International Financial Reporting Standards), US GAAP, German GAAP, Swiss GAAP, etc. Each accounting system demands the reporting of different numbers and data. Sometimes the differences are smaller, sometimes bigger. To make things more complicated, there are different reporting requirements in every country for banks and insurance companies than there are for industrial companies.

For automatic electronic reporting purposes, each reporting standard has to be converted into a standardized structure. In XBRL, this is done with a hierarchical structure (taxonomy) to cope with the complex and extensive accounting rules. Hence, taxonomies consist of XML schema documents and so-called *Linkbases* (see Figure 3). Schema documents and Linkbases are separate files, but they are an entity and together constitute a taxonomy (EDGAR Online, 2011).

The schema documents represent an unsorted list of declared element names and their corresponding attributes (Hoffman 2010, 82). As schema documents contain a predefined list of a business report’s possible contents, taxonomies are often interpreted as “digital dictionaries” for the transmission of financial statements, for instance (Hoffman 2010, 301). It would be theoretically possible to store all declared elements in one single document, but this would be difficult, due to thousands of elements that are needed for the markup of a financial statement (e.g., the US-GAAP Taxonomy contains approximately 19,000 monetary and non-monetary element names).

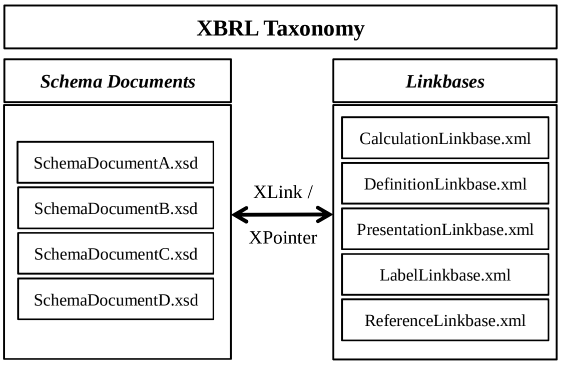


Figure 3: Basic structure of an XBRL Taxonomy

For this reason, the elements and their associated attributes are usually stored according to their purpose in different schema documents. Elements that have been defined in an XBRL Taxonomy are so-called *concepts* (Hoffman 2006, 67).

```
<element
            name="NetIncomeLoss"
            id="us-gaap_NetIncomeLoss"
            nillable="true"
            type="monetaryItemType"
            balance="credit"
            periodType="duration"
 />
```

Figure 4: Concept declaration

Figure 4 illustrates an excerpt of an element declaration from the US-GAAP Taxonomy. In this figure an element with the name *NetIncomeLoss* is declared. Companies can use the element name for transmitting a financial line item: in this context, net income in accordance with US GAAP standards.

The XBRL Specification provides several elements and attributes (vocabulary) that can be used to describe the declared elements in more detail. The attribute *nillable* (possible value: true/false) determines if there is an obligation to report this item in the instance document (SEC, 2010). This concept does not need to be included in the report if the value is true. The attribute *type* expresses if the concept is a monetary item, a string item, a date item and so on. The taxonomy developer may add an optional *balance* attribute (possible value: debit/credit) to the concept definition if it is a monetary item type (XBRL International Incorporated 2008, 80). For example, it will indicate if the reported fact is an asset or a liability in the Statement of Financial Position. The attribute *periodtype* indicates if the concept is an instant or duration type. The net income is a duration type because it is part of the Statement of Income (Hoffman 2010, 89).

A special feature of XBRL is to describe complex relationships (links) between different concepts (concept-to-concept link) or to add auxiliary information to concepts (one-way link). The different links are stored in separate files according to their purpose, the so-called Linkbases (e.g., label links are generally stored in a separate document, the so-called *LabelLinkbase*). The supported Linkbases according to the XBRL Specification are shown in Figure 3. The Calculation-, Definition- and PresentationLinkbase contain concept-to-concept links, whereas the Label- and ReferenceLinkbase contain one-way links.

The links are built with the help of the W3C specifications *XML Linking Language (XLink)* and *XML Pointer Language (XPointer)*. Every concept has the attribute *id* that serves as unique identifier (Hoffman 2010, 88). In Figure 4, the identifier of the declared concept is *"us-gaap_NetIncomeLoss"*. With the help of the identifier, XPointer can locate (point to) concepts in the schema document. XLink is used to describe the relationships (links) between two located concepts or from one located concept to auxiliary information. The concrete XLink and XPointer rules can be looked up in the XBRL Specification (http://www.xbrl.org/SpecRecommendations).

A calculation link between two monetary item type concepts enables them to be linked mathematically, but with the limitation it only allows the description of the summation or subtraction between them (EDGAR Online, 2011). For example, the use of calculation links enables the description of net income as total earnings minus expenses. All specified calculation links between concepts are aggregated to a Linkbase, in this case the *CalculationLinkbase*. The function of the calculation links is important because it makes it possible to control if the reported monetary statements are mathematically complete and correct (XBRL Spain 2005, 21).

The *DefinitionLinkbase* serves to express different kinds of (inter)relationships between concepts (Hoffman 2006, 67). For example, it can be deposited that an explanation to the impairment must be disclosed in the notes in the case of asset impairment.

The main function of the *PresentationLinkbase* is to display the list of unsorted concepts in a hierarchical structure according to the presentation rules of the accounting standards. Additionally, for each hierarchical level the order of the concepts can be deposited according to the particular formal requirements (IASCF 2010, 23). For example, within the Statement of Financial Position the assets are comprised of current assets and non-current assets. Furthermore, US GAAP requires current assets to be displayed before non-current assets. This can be implemented with the use of presentation links. All in all, the presentation links offer the possibility to group and sort the unsorted list of declared schema elements for the human eye.

The *LabelLinkbase* offers the possibility to add a human-readable name (e.g., net income) for a concept (e.g., <NetIncomeLoss/>). If several links with human-readable names in different languages have been defined, XBRL reports can be prepared and read in different languages (van der Heiden 2006, 15). For example, the company Apple could provide the numbers of its balance sheet. Analysts from Germany could choose the German language and they would receive a report with lines like "Sachanlagen", "Vorräte" and so on. An English-speaking analyst would see "Property, Plant & Equipment" and "Inventories" on his report. By overcoming the language barrier in this way, information about foreign competitors is easier to understand.

The aim of the *ReferenceLinkbase* is to reference the underlying legal background of the concept and descriptive literature in commentaries. Reference links may also provide documentations about the correct usage of the special concept.

In summary, taxonomies consist of schema documents and Linkbases. Schema documents only represent a container of unsorted concepts. They will be structured with the individual Linkbases.

In the area of accounting, taxonomies are primarily developed and published by such standard-setters as the IFRS Foundation or the Financial Accounting Standards Board (FASB). Depending on the particular legal situation or XBRL adoption degree in the respective countries, the reporting companies may or are required to use the taxonomies to create and file reports in XBRL format (instance documents). Due to the standardized markup structure, XBRL reports can be automatically readout and processed by computer programs. To fulfill this aim, it is important that all participants of the reporting chain use the same standardized taxonomy. How easily information from XBRL reports can be extracted shall be illustrated with the help of SEC Filings according to the US-GAAP Taxonomy in the following.

3. Extracting competitive information from XBRL Financial Statements

This section describes how competitive information can be extracted from US GAAP XBRL reports. An actual annual report from the company Apple Incorporated serves as basis for the illustration.

3.1 Financial data provided by the SEC

XBRL can be implemented for different business reporting issues (e.g., banking supervision, tax and other regulatory reporting as well as internal management reporting). However, XBRL originally has been created to improve the data exchange of financial statements. With different taxonomies, it is possible to represent the specific national accounting standards like US GAAP, IFRS or German GAAP.

In the U.S., companies have to use the US-GAAP Taxonomy when they are obligated to prepare their financial statements according to US GAAP and SEC regulations (XBRL US, 2008). In 2006, the non-profit jurisdiction XBRL US was commissioned by the U.S. Securities and Exchange Commission (SEC) to develop a taxonomy that is consistent with US GAAP requirements and the Commission's regulations (SEC Release 33-9002 2009, 12). In 2010, the on-going development and maintenance responsibilities for the US-GAAP Taxonomy devolved to the FASB (FASB, 2011). The taxonomies supported by the SEC XBRL mandate are listed on the Web site http://www.sec.gov/info/edgar/edgartaxonomies.shtml.

Because of the SEC XBRL mandate (or Interactive Data Program), many XBRL filings of listed companies are available for analysis online. Beginning with fiscal periods ending on or after June 15, 2009, domestic and foreign large accelerated filers that prepare their financial statements in accordance to US GAAP and have a public equity float above $5 billion were required to provide their financial statements to the SEC and on their Web sites in XBRL format (SEC Release 33-9002 2009, 42). All other public companies that fell under the definition of large accelerated filers using US GAAP were required to submit their financial statements in XBRL format for fiscal periods ending on or after June 15, 2010. Finally, all remaining US GAAP filers and all foreign private issuers using IFRS had to comply with the XBRL requirements in year three of the phase-in (SEC Release 33-9002 2009, 43). For foreign private issuers using IFRS, the requirement to file XBRL reports was postponed until SEC approval of the IFRS-Taxonomy (SEC, 2011). It was estimated that about 500 companies in year one, 1,800 companies in year two and about 12,000 companies in year three of the phase-in were required to submit their filings in XBRL (Hoffman 2010, 219) to the SEC Electronic Data-Gathering, Analysis, and Retrieval system (EDGAR). Anyone can access this data pool and download the XBRL filings (Forms 10-K, 10-Q, etc) free of charge. By providing several types of RSS Feeds, all XBRL filings can be downloaded to and integrated into a database or an analysis tool. In combination with the EDGAR system, XBRL enables competitive information from thousands of companies to be downloaded and analysed almost in real-time.

3.2 Extracting Apple's XBRL Data

For extracting all information that an XBRL report provides, a special XBRL Processor is needed. The reason is that an XML Processor has no knowledge of XBRL and thus is not able to understand and handle the structure and relationships among the different XBRL documents (Hoffman 2006, 494). An XBRL Processor can follow the XLink and XPointer expressions and is able to put the different information together. It can read, write, control, handle or otherwise process XBRL data (Hoffman 2010, 232). An XML Processor can also be used to extract information; however, it is not possible to use all information (e.g., to mathematically check for correctness and completeness) XBRL documents provide (Hoffman 2010, 24).

An XML Processor is a software program that can read, change, delete or transform XML documents. The module of the XML Processor responsible for the reading-in of an XML document is called XML Parser. An XML Parser facilitates access to the content of an XML document by converting it into an Application Programming Interface (API). Afterward, this API can be accessed with programming languages for further processes (Maruyama 2002, 21). One possible programming language is Visual Basic for Applications (VBA), which can be directly embedded in MS Excel (Hoffman 2006, 495).

MS Excel is a well-known and widely used analysis tool. Furthermore, one important component is already integrated into it: an XML Parser. As a result, MS Excel can be a useful tool for extracting competitive information from XBRL financial statements. With only a little technical expertise, XBRL data can be extracted without the help of special software. Because the built-in XML Parser is used, only a stand-alone instance document and not the (extension) schema and the different (extension) Linkbases can be used (note: XBRL supports creating own individual concept-extensions if the taxonomy structure does not provide the adequate concept for transmission. However, when the taxonomy structure is extended or adjusted, it is necessary to publish the corresponding extension schema and extension

Linkbases.). Nevertheless, this simple approach can generate huge benefits for CI.

Apple’s annual report for the fiscal year 2010 can be downloaded from the SEC EDGAR database in the data formats HTML/ASCII and XBRL. Figure 5 shows a simplified excerpt of the XBRL report (instance document). Among other data, it contains all information needed for the automatic extraction and calculation of the key metric *Return on Sales (after interests and taxes)* that is defined as the ratio between net income and sales (Tracy 2009, 132). It is one way of measuring a company’s profitability (here the Return on Sales after interests and taxes). Therefore, it is a useful key performance indicator for many Competitive Intelligence purposes. However, there are an infinite number of other calculations that could be automated as well.

In accordance to US GAAP, companies have to report their sales revenues as net value, that is as revenues earned from selling products minus sales returns, sales allowances and sales discounts. Therefore, for the calculation of this key metric, Apple’s net sales is inserted into the formula for the term sales. The net income is calculated after subtracting the expenses from earnings and represents the profit for the year attributable to shareholders. In Apple’s instance document, the values for the numerator and denominator of the ratio Return on Sales are transmitted by the predefined element names *SalesRevenueNet* and *NetIncomeLoss* of the US-GAAP Taxonomy 2009.

```
<?xml version="1.0" encoding="US-ASCII" standalone="yes" ?>

<xbrl
        xmlns="http://www.xbrl.org/2003/instance"
        xmlns:us-gaap="http://xbrl.us/us-gaap/2009-01-31"
        xmlns:dei="http://xbrl.us/dei/2009-01-31">

        <schemaRef  href="aapl-20100925.xsd"/>

        <dei:EntityRegistrantName contextRef="2010">APPLE INC</dei:EntityRegistrantName>
        <dei:DocumentFiscalYearFocus contextRef="2010">2010</dei:DocumentFiscalYearFocus>
        <dei:DocumentFiscalPeriodFocus contextRef="2010">FY</dei:DocumentFiscalPeriodFocus>
        <us-gaap:SalesRevenueNet contextRef="2010" decimals="-6" unitRef="iso4217_USD">65225000000
        </us-gaap:SalesRevenueNet>
        <us-gaap:NetIncomeLoss contextRef="2010" decimals="-6" unitRef="iso4217_USD">14013000000
        </us-gaap:NetIncomeLoss>

        <context id="2010">
                        <period><startDate>2009-09-27</startDate><endDate>2010-09-25</endDate></period>
        </context>

        <unit id="iso4217_USD">
                        <measure>iso4217:USD</measure>
        </unit>

</xbrl>
```

Figure 5: Simplified excerpt of Apple's instance document

In order to distinguish between GAAP (prefix: us-gaap) and Non-GAAP element names (prefix: dei), a so-called *prefix* is used. A prefix in the start-tag of an element associates a specific namespace to single element names instead of assigning a default namespace for all element names within an instance document (see section 2, Figure 2). Each element name prefix is associated with an own URI (Harold 2004, 65).

For human beings the instance document in Figure 5 might look a bit confusing. But computer programs can find a path through this “data jungle”, finding and extracting the information needed. The standardized structure enables the selective and automatic analysis of financial statements.

In our approach, a few lines of VBA code will need to be written (see Figure 6) and the code will have to be inputted into the Visual Basic Editor in MS Excel. First, it is necessary to convert the instance document to an API so that the document content can be accessed. Afterward we can search for the element names *SalesRevenueNet* and *NetIncomeLoss* and import the contained values into an MS Excel spreadsheet. In Figure 6 the VBA code for the extraction of the net sales (see the bold expression) is illustrated. If we feed the storage location of the instance document into column A in the MS Excel spreadsheet (see Figure 7) and execute the VBA program (or VBA Macro), this specific fact value will automatically be imported into the denoted column E.

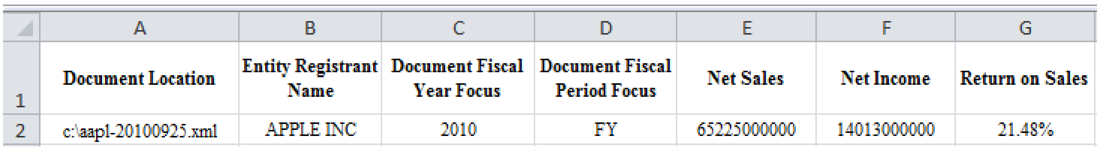
```
Sub ExtractXBRLforCI()

Dim Row As Range
Set Row = Sheet1.Range("A2")

'XML Parser ------------------------------------------------------------------------------------
   Dim instanceDocument As MSXML2.DOMDocument
   Set instanceDocument = New MSXML2.DOMDocument
   instanceDocument.async = False
   instanceDocument.validateOnParse = False
   instanceDocument.Load (Row)

'XBRL Extraction -------------------------------------------------------------------------------
   Dim Nodelist As MSXML2.IXMLDOMNodeList
   Set Nodelist = instanceDocument.getElementsByTagName("us-gaap:SalesRevenueNet")
   Dim Node As Object

   For Each Node In Nodelist
   Cells(Cells(Rows.Count, "E").End(xlUp).Row + 1, "E").Value = Node.Text
   Next Node

End Sub
```

Figure 6: Sample VBA code to extract an XBRL fact value

3.3 Results

By extending the VBA code (or replacing the bold expression), the remaining columns in the MS Excel spreadsheet (columns B, C, D and F; see Figure 7) can be filled.

After the import of the needed information into the spreadsheet, normal MS Excel formulas can be applied to the values (column E and F) in order to calculate the requested key metric. For the company Apple we calculate a Return on Sales of 21.48 % for the fiscal year 2010 with the aid of Apple's XBRL data. The result is displayed in column G. Therefore, with XBRL no manual work for the calculation of the Return on Sales is needed anymore. If we do this calculation only once and for one company, the benefits of this approach seem to be limited. The true potential appears if we imagine that the procedure will be applied to many companies. By extending the VBA Macro with a few more lines of code, it would be possible to calculate a ratio (or dozens of them) for thousands of competitors in a fully automated process. It would be possible to compare Apple's performance measure with all other examined companies (or the industry average) by a pivot table (benchmarking) or further graphical analysis, for example. Often used analytical CI techniques like benchmarking and competitor profiling (e.g., the profiling of financial statements) (Bouthillier 2003, 54) therewith can be supported.



	A	B	C	D	E	F	G
1	Document Location	Entity Registrant Name	Document Fiscal Year Focus	Document Fiscal Period Focus	Net Sales	Net Income	Return on Sales
2	c:\aapl-20100925.xml	APPLE INC	2010	FY	65225000000	14013000000	21.48%

Figure 7: Extraction results

4. Effects of XBRL on the Development of Competitive Intelligence

The ultimate goal of CI is to gather and analyse as much (external) information as possible in order to guide strategy by understanding a company's marketplace competitiveness and its adaptability to future changes in the competitive environment. In the literature the CI process is often divided into the following four steps: (1) Direction, (2) Collection, (3) Analysis and (4) Dissemination (Vrien 2004, 3). For the collection of competitive information (step (2)) there are several different sources possible. Studies found that the systematic screening of the internet is among the most important and widely-used instruments of CI (Vrien 2004, 11 and 17). The "internet" technology XBRL provides a lot of opportunities for CI. When all participants in the reporting chain (sender and receiver) use the same XBRL taxonomy, an automatic selection of individual desired data is possible. A time-consuming manual search through online available financial statements will not be needed anymore. In combination with other internet technologies like RSS, the financial data can be extracted almost in real-time directly from the source and it doesn't have to be acquired at high cost. Besides the analysis getting faster and cheaper, a broader data basis can be examined. Mass data can easily be analysed as well as textual or qualitative data (e.g., information about the company's strategy and the managers' forecast to the future performance) with the use of XBRL. With taxonomies (esp. LabelLinkbases) available in different languages, the collection of data can be driven independent from language hurdles. This will become more and more important for CI due to globalized markets.

All in all, XBRL contributes to a quantitative better collection of data without reducing the data quality. The data quality rather increases. The fact that step (2) in the CI process improves, has also positive consequences for the steps (3) and (4). On the basis of better data, qualitatively and quantitatively, more reliable decisions are possible.

5. Summary

The article illustrates a simple approach to automate the extraction and further processing of financial statement information (e.g., for profiling of financial statements) using publicly available XBRL reports and MS Excel. With the creation of a simple VBA Macro, XBRL data enables calculating not only one stand-alone key metric, but whole MS Excel templates (e.g., scoring systems or benchmarking models) can be fed with financial data.

The XBRL technology provides a lot of opportunities for CI. Competitive information from financial statements can be collected and analysed independent of former limitations (e.g., data volume, language or qualitative data). Designed as an open-standard, it is possible to customize the use of XBRL to own individual needs so that it can greatly simplify and speed up the analysing of financial data.

References

Binstock C et al. (2003) *The XML Schema Complete Reference*, Pearson Education Inc., Boston, Massachusetts

Bouthillier F and Shearer K (2003) *Assessing competitive intelligence software: a guide to evaluating CI technology*, Information Today Inc., Medford, New Jersey

EDGAR Online (2011) *Try XBRL Glossary*, Available online on URL: http://www.tryxbrl.com/Learn/Glossary/tabid/58/Default.aspx

Evjen B et al. (2007) *Professional XML*, Wiley Publishing, Indianapolis, Indiana

FASB (2011) *FAF/FASB XBRL Taxonomy Role,* Available online on URL: http://www.fasb.org/jsp/FASB/Page/

Harold E and Means S (2004) *XML in a Nutshell*, Third Edition, O'Reilly Media Inc., Sebastopol, California

Hoffman C (2006) *Financial Reporting Using XBRL*, Available online on URL: http://frux.wikispaces.com/

Hoffman C (2010) *XBRL for Dummies*, Wiley Publishing, Indianapolis, Indiana

IASCF (2010) *The IFRS Taxonomy Guide 2010*, Available online on URL: http://www.ifrs.org/NR/rdonlyres/38EAB57A7264A7491ECEEEF29BBE8A6/0/ITG201020100702.pdf

Kahaner L (1998) *Competitive Intelligence: How to Gather, Analyse and Use Information to Move Your Business to the Top*, Touchstone, New York

Maruyama H (2002) *XML and Java*, Second Edition, Pearson Education Inc., Boston, Massachusetts

SEC Release 33-9002 (2009) *Interactive Data to Improve Financial Reporting*, Available online on URL: http://www.sec.gov/rules/final/2009/339002.pdf

SEC (2010) *XBRL Glossary*, Available online on URL: http://www.sec.gov/spotlight/xbrl/glossary.shtml

SEC (2011) *No-Action Letter*, Available online on URL: http://www.sec.gov/divisions/corpfin/cfnoaction/2011/caq040811.htm

Tracy J (2009) *How to read a financial report*, Seventh Edition, John Wiley & Sons, Hoboken, New Jersey

van der Heiden J (2006) *XBRL in Plain English*, Available online on URL: http://www.bataviaxbrl.com/downloads/XBRLinPlainEnglishv1.1.pdf

van der Vlist E (2002) *XML Schema*, O'Reilly Media Inc., Sebastopol, California

Vrien D (2004) *Information and Communication Technology for Competitive Intelligence*, Idea Group Publishing, Hershey, Pennsylvania

W3C Recommendation (1999) *HTML 4.01 Specification*, Available online on URL: http://www.w3.org/TR/html401/

W3C Recommendation (2008) *Extensible Markup Language (XML) 1.0*, Available online on URL: http://www.w3.org/TR/REC-xml/

W3C Recommendation (2009) *Namespaces in XML 1.0*, Available online on URL: http://www.w3.org/TR/REC-xml-names/

W3 Schools (2011), *XML Namespaces*, Available online on URL: http://www.w3schools.com/xml/xml_namespaces.asp

Watt A (2002), *XML in 10 minutes*, Sams Publishing, Indianapolis, Indiana

XBRL International Incorporated (2008) *XBRL 2.1 Specification*, Available online on URL: http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-1231+Corrected-Errata-2008-07-02-redlined.doc

XBRL Spain (2005) *White Paper on XBRL Technology*, Available online on URL: http://www.xbrl.es/downloads/libros/White_Paper.pdf

XBRL US (2008) *US GAAP Taxonomy Preparers Guide*, Available online on URL: http://xbrl.us/Documents/PreparersGuide.pdf