1. Conceptual Overview of XBRL-based Digital Financial Report

We provided you with a 50,000 foot view and a 10,000 view of digital financial reporting in previous sections. Here, we drop down to about 1,000 feet. Be patient, we will be at ground level soon enough.

In this section we want to begin to tie the bigger picture with the details and make XBRL-based digital financial reports more tangible for you. We are going to do this step-by-step to help the reader understand the paradigm shift that they need to make.

Undoubtedly, artificial intelligence will bring significant changes. Study after study like *AI for Services*[1] makes this crystal clear. The XBRL-based digital financial report is part of this change. Accountants and auditors will be impacted.

The digital general purpose financial report[2] is an improvement that helps move the institution of accountancy forward, providing a necessary improvement to that tool for the institution of accounting globally. Given today's increasing volume of financial information, complexity of financial information, and importance of financial information; it makes perfect sense to provide such a digital alternative or option to current paper-based reports.

Financial analysis has been digital for many years; first via the electronic spreadsheet and now with a multitude of options.

Perhaps I am stating the obvious. With digital books, maps, photos, films, music, blueprints, etc.; what about the digital financial statement does not make sense?

1.1. Accounting, Reporting, Auditing, and Analysis in a Digital Environment

But it is not just the general purpose financial report that should be machine-readable. Special purpose financial reports should likewise have the option to be created digitally. In fact; accounting, reporting, auditing, and analysis need to be updated for the digital environment.

People talk about things like "continuous accounting" and "continuous auditing" and "finance transformation" and "smart regulation" and "algorithmic regulation". We went over all this in *Computational Professional Services*[3] so you understand that these are all terms for the same idea.

Effectively, XBRL is about enabling the possibility to automate certain tasks and processes involved the flow of information through the systems that relate to financial information. The outputs of one part of the system tends to be the inputs to other system parts.

---

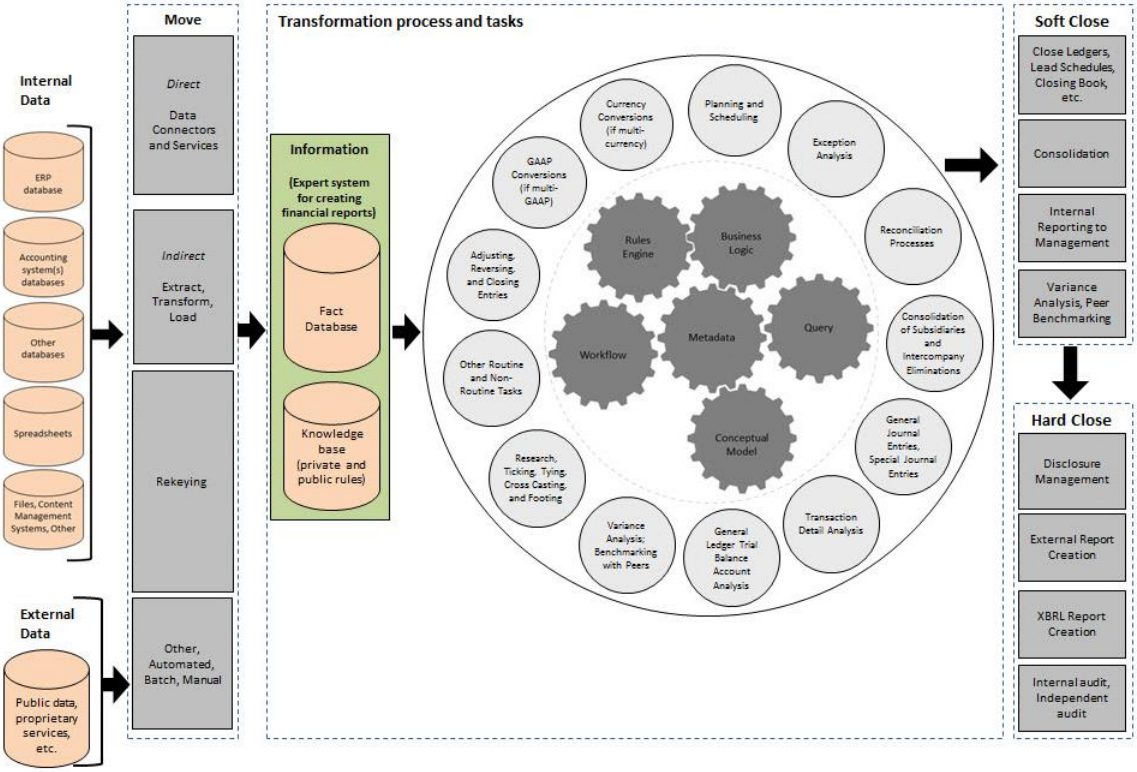
[1] London Economics, *AI for Services (Full Report 2020)*, https://ktn-uk.org/wp-content/uploads/2020/09/AI-for-Services-full-report-2020_KTN-green_Final.pdf

[2] Charles Hoffman, *Step-by-Step Explanation as to How My Automated Reporting Checklist Works*, http://xbrl.squarespace.com/journal/2018/1/5/step-by-step-explanation-as-to-how-my-automated-reporting-ch.html

[3] Charles Hoffman, CPA, *Computational Professional Services*, http://www.xbrlsite.com/mastering/Part00_Chapter01.A1_ComputationalProfessionalServices.pdf

This graphic shows a simplified abstraction that explains the parts that tend to interact with one another and how those parts fit together:

While the above graphic shows the role of XBRL and where the external financial report fits, the graphic is rather abstract and not very tangible. The following graphic is more tangible:



The above graphic shows the tasks and processes that are involved in the creation of a financial report. The graphic was inspired by a similar graphic provided by Blackline4. How many and which tasks and processes can be automated is still an unanswered question. Particularly within a global, multinational corporation who might benefit the most from automation; the complexity of automation can be substantial. Many issues are involved which we will introduce step-by-step. If technology can be employed to improve tasks and processes; performing tasks better which improves quality, performing them faster which improves timeliness, and/or performing them cheaper reducing costs; why would an organization not want to experience such a productivity gain? Alternatively, new products and services can be offered or delivered in new innovative ways. But first, let us truly understand to objective of what we are trying to achieve.

1.2. Augmented Intelligence

What we are trying to achieve is better human and computer teaming. Another term for this is augmented intelligence[5]. Another term used is intelligence amplification[6]. In his book, *Principles*[7], Ray Dalio advises,

> "By developing a partnership with your computer alter ego in which you teach each other and each do what you do best, you will be much more powerful than if you went about your decision making alone."

Augmented intelligence is about computers and humans working together rather than machines replacing humans. Augmented intelligence is about computers doing what they do best and humans doing what they do best. Augmented intelligence applications combine human and machine intelligence. This is particularly important in systems there is low tolerance for error or where artificial intelligence is not evolved enough to take humans completely out of the loop.

But how exactly do you get computers to perform work effectively? We provided details in the section *Computational Thinking*[8]. Now we want to walk you through those details step-by-step.

1.3. Moving from "Data" to "Data + Knowledge" Mentality

Today, most professional accountants and auditors are very familiar with working with software applications that work with "**data**". You enter *data* into spreadsheets, you might move the spreadsheet *data* into a software application that stores the *data* in a database, you might need to use ETL[9] (extract, transform, load) to move the data from one application to some other application. Everyone's focus is on data, data, data.

But there are two problems with data. First, data is only understandable by the software application that has the data within it's database. For example, what do

---

[4] Blackline, *Finance Controls and Automation Platform*, https://www.blackline.com/finance-controls-and-automation/

[5] Aura Quantic, *What is Augmented Intelligence and why should you know about it?*, https://www.auraquantic.com/what-is-augmented-intelligence/

[6] Wikipedia, *Intelligence Amplification*, https://en.wikipedia.org/wiki/Intelligence_amplification

[7] Ray Dalio, Principles, https://www.principles.com/the-changing-world-order/

[8] Charles Hoffman, CPA, *Computational Thinking*, http://www.xbrlsite.com/mastering/Part00_Chapter01.C_ComputerEmpathy.pdf

[9] Wikipedia, *Extract, Transform, Load*, https://en.wikipedia.org/wiki/Extract,_transform,_load

you think would happen if you took the database and data from one software application and connected that database to some other software application? Would the second software application understand the data in the database of the first software application? Certainly not; that is why you have to be careful when moving data from one software application to another because data is only understandable in context and if you change the context the data will not be understandable.

Second, you put "data" into your databases but you don't put all the necessary "knowledge" into those same software applications.

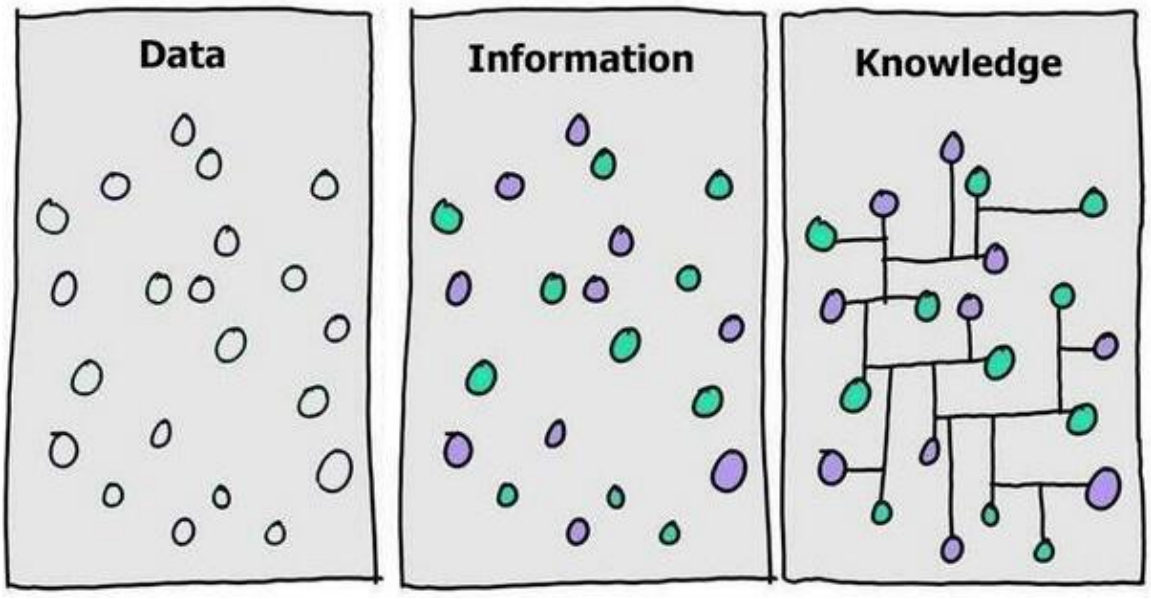
The information age is not about "data"; the information age is about:

Data + Knowledge

This is what we mean. ***Data*** are discrete, objective facts (numbers, words) without context or interpretation. For example, the number "241,086,000,000" is data. Information is data in context.

Information is data that has been arranged, collated, and categorized. "Data" organized within the rows and fields of a database has the characteristic of being information. "Assets for the consolidated legal entity Microsoft Corporation as of June 20, 2017 was $241,086,000,000 expressed in US dollars and rounded to the nearest millions of dollar," is information.

Knowledge is a set of data and information and a combination of skill, know-how, experience which can be used to improve the capacity to take action or support a decision making process. Data in context, with a rich set of associations, structures, models, relations, rules, and other such information is how you create knowledge. Shown graphically[10]:



[10] Random Blather, *Information Isn't Power*, https://random-blather.com/2014/04/28/information-isnt-power/

The point here is that you don't simply take a computer, put it next to a human, and then get human-machine teaming and all the productivity gains that make things better, faster, and cheaper. That is absurd.

The knowledge of professional accountants and auditors is gained through extensive training, experience, and skill. For a machine to "augment" the intelligence of a professional accountant, the machine must have access to that knowledge.

So how do you do that?

1.4. Approaches to Storing Knowledge and Problem Solving

There are three primary approaches that have been developed over the past 50 or so years for storing knowledge and then using that stored knowledge to solve problems. Those three primary approaches are[11]:

- **Knowledge graphs**: W3C Semantic Web stack of technologies including RDF, SWRL, N3, OWL, SHACL, SPARCL, etc.
- **Graph databases**: Labeled property graphs such as Neo4j and it's query language Cypher as one example, property graphs, ISO standard Graph Query Language (GQL)
- **Logic programming**: Modern Prologs on top of relational databases; ISO standard Structured Query Language (SQL)

Going into the details of each alternative approach is beyond the scope of this discussion. The primary point to understand here is that there are three different alternative approaches.

Two obvious questions here might be: (1) Which is the best approach? (2) How do you get one approach to interoperate with the other two approaches?

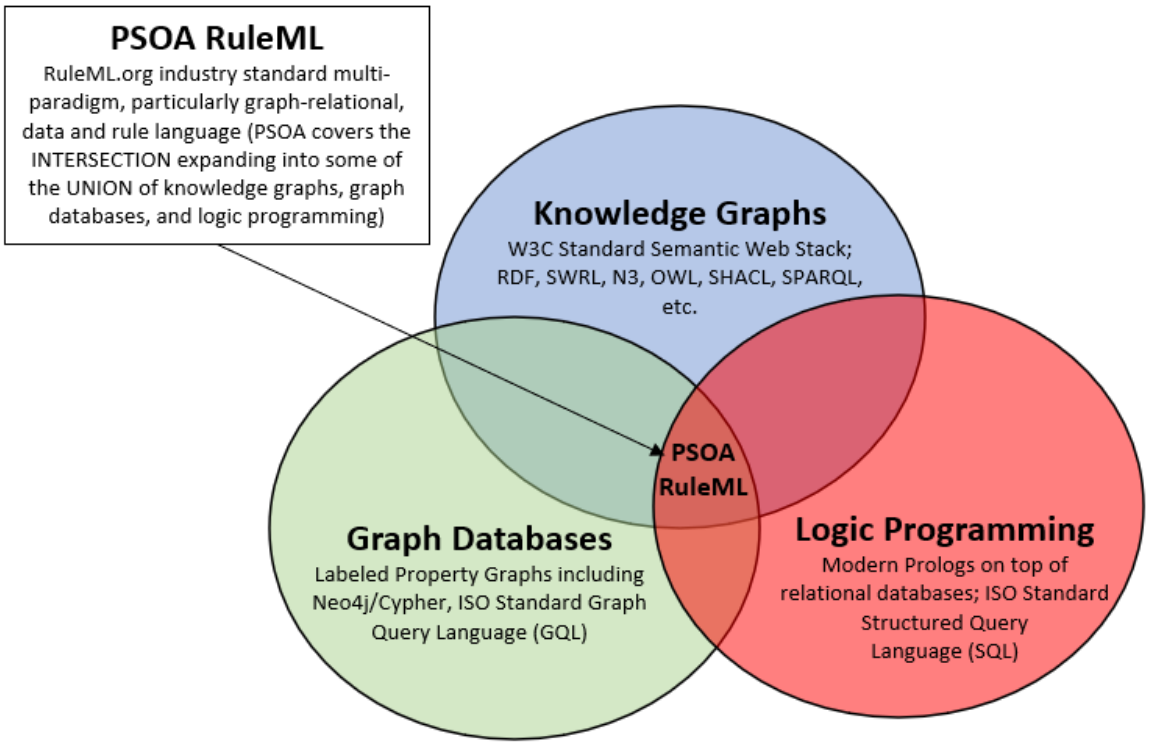
Different organizations do or will use different technical approaches to storing knowledge and problem solving approaches to use that knowledge to create augmented intelligence. The industry standards organization, RuleML.org, which works to create standards for rule interoperability has summarized these three primary paradigms and has created a method to bridge the gap between these three different paradigms.

Again, going any further into the technical details are beyond the scope of what we are trying to communicate to what is likely a very nontechnical business professional who is likely reading this information.

Summarizing this very succinctly: augmented intelligence can in fact work and there are three robust approaches that could be used to implement augmented intelligence.

The following graphic was inspired by a version of this same graphic provided by the RuleML.org paper. I have enhanced that original graphic to summarize the pertinent information that is relevant to business professionals. Here is my enhanced version of that same graphic:

[11] RuleML.org, Harold Boley, *Graph-Relational Data, Ontology, Rules*, http://wiki.ruleml.org/index.php/Graph-Relational_Data,_Ontologies,_and_Rules



Artificial intelligence, structured information, distributed ledgers, machine readable workflow model standards, machine readable decision model standards, and Lean Six Sigma are a match made in heaven and will have a significant impact on accounting, reporting, auditing, and analysis in a digital environment that will prevail during the Fourth Industrial Revolution[12].

1.5. Complexity

Each of the three knowledge storage and problem solving approaches have off-the-shelf products or solutions. However, all three approaches are incredibly complex and technical even for the average software engineer. There is zero probability that the average professional accountant or auditor would ever begin to understand how to use these tools.

The good news is that the average professional accountant or auditor will never be exposed to the complexity.

The ***Law of Conservation of Complexity*** states: "Every application has an inherent amount of irreducible complexity. The only question is: Who will have to deal with it-the user, the application developer, or the platform developer?" Another version of the law of conservation of complexity: "Every application has an inherent amount of complexity that cannot be removed or hidden. Instead, it must be dealt with, either in product development or in user interaction."

[12] *Adapting to Changes Caused by the Fourth Industrial Revolution*, http://xbrl.squarespace.com/journal/2019/8/4/adapting-to-changes-caused-by-the-fourth-industrial-revoluti.html

Irreducible Complexity is explained as follows: A single system which is composed of several interacting parts that contribute to the basic function and where the removal of any one of the parts causes the system to effectively cease functioning.

So, for example, consider a simple mechanism such as a mousetrap. That mousetrap is composed of several different parts each of which is essential to the proper functioning of the mousetrap: a flat wooden base, a spring, a horizontal bar, a catch bar, the catch, and staples that hold the parts to the wooden base. If you have all the parts and the parts are assembled together properly, the mousetrap works as it was designed to work.

But say you remove one of the parts of the mousetrap. The mousetrap will no longer function as it was designed; it will not work. That is **irreducible complexity**: the complexity of the design requires that it can't be reduced any farther without losing functionality.

Anyone can create something that is complex. It is much harder to create something that is sophisticated and simple. Simple is not the same thing as simplistic. "Simple" is not about doing simple things. Simplicity is "dumbing down" a problem to make the problem easier to solve. Simple is about beating down complexity in order to make something simple and elegant; to make sophisticated things simple to use rather than complex to use.

Creating something that is simple takes conscious effort and is hard work. But that is what is necessary to make computational professional services work as desired.

1.6. Utility of High-level Models

Universal tools that are used to solve every possible problem tend to be far more complicated to use that domain specific tools because universal tools tend to be far more flexible and have broader problems that they need to solve than general tools. This is discussed in the next section.

Creating a high-level model which can be used to (a) hide technical complexity and (b) limit a universal tool and turning the universal tool into a domain specific tool that is easier to use.

1.7. Universal vs Domain Specific Applications

Having high-level metamodels such as the forthcoming *Standard Business Report Model*[13] (SBRM) and *Logical Theory Describing Financial Report*[14] (see). Plus, creating the base metadata, such as the US GAAP Financial Reporting Metadata[15] leveraging that high-level metamodel makes all of this extremely technical stuff far less technical to business professionals. How? A few trained professionals create the high-level metadata but every software application and business domain professional benefits from that metadata and high-level models. Further, software creation costs are reduced. How is that possible? Read on.

[13] *SBRM Progress Report*, http://xbrl.squarespace.com/journal/2020/1/30/sbrm-progress-report.html

[14] *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/

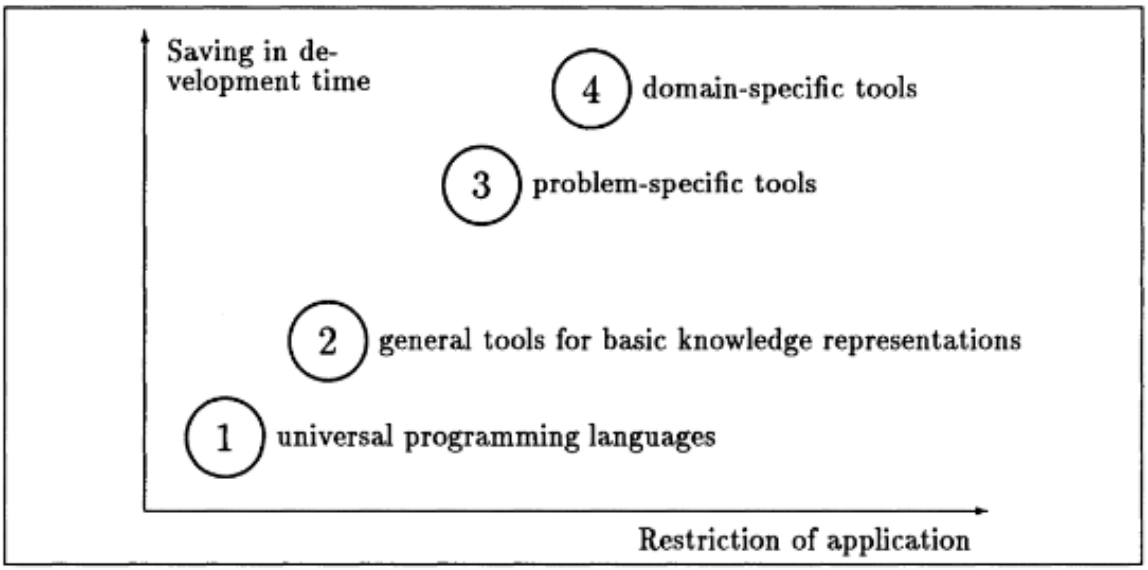
[15] *US GAAP Financial Reporting Scheme*, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/documentation/Home.html

In his book *Systematic Introduction to Expert Systems*[16], Frank Puppe provides the graphic below. The graphic basically points out that **universal, general tools** are less restrictive but cost more to create than **domain-specific tools**. In addition to universal, general tools being more costly to create and more difficult to create; domain specific tools are easier to create and much, much easier for business professionals to use because of the restrictions.

So, a "restriction" is not a flaw. **The restriction is what makes the tool easier to use, cost less, and make software easier to develop**. You don't need the universe of all possible options for a specific domain; you only need to create what that specific domain needs. As **long as you get these restrictions correct**, they really are not "restrictions" of the domain, they are the "boundaries" of the domain. You don't need them.

Technical people don't typically understand these business domain boundaries. Many times, to play it safe technical people add flexibility in order to make certain that business domain user needs are being met. But this flexibility comes at a cost. Additional costs are incurred to create the flexibility and software is harder to use because business professionals need to figure out which option they should use.

Business domain people do understand the boundaries if they think about them. Many business professionals cannot properly articulate the appropriate boundaries or restrictions. This communications problem tends to lead to software that costs more to create than is necessary and harder to use than necessary.



This is not an either-or choice. Sometimes universal tools are very appropriate. Other times domain-specific tools are appropriate. Being conscious of these dynamics will lead to the right software being created and the appropriate level of usability. Universal tools are not a panacea. Unconsciously constricting a domain-specific tool when it would have been better to create a more universally usable tool also can be a mistake one makes.

[16] Frank Puppe, *Systematic Introduction to Expert Systems*, page 11, https://books.google.com/books?id=_kKqCAAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Today, everyone is competing at the "universal tool" level and not one of those universal tools is usable by business professionals. Computational Professional Services is a vertical; but it is an incredibly WIDE (i.e. horizontal) vertical market.

1.8. Process Control

Because, as we pointed out, financial reports are not static forms and therefore individual economic entities are allowed to make specific modifications to models; those modifications need to be controlled in order to maintain information quality. Said another way, permissible modifications to the model must be crystal clear to those making such modifications.

Lean Six Sigma[17] philosophies and techniques offer many insights and ideas related to process control. The 1-10-100 Rule is related to what's called "the cost of quality." Essentially, the rule states that prevention is less costly than correction which is less costly than failure. It makes more sense to invest $1 in prevention, than to spend $10 on correction. That in turn makes more sense than to incur the cost of a $100 failure[18].

A kludge (or kluge) is an engineering/computer science term that describes what is best described as a workaround or quick-and-dirty solution that is typically clumsy, inelegant, inefficient, difficult to extend and hard to maintain; but it gets the job done. By contrast, elegance is beauty that shows unusual effectiveness and simplicity.

1.9. Essence of a General Purpose Financial Report

The essence of a general purpose financial report is described within the *Essence of Accounting*[19] in detail. In summary, a general purpose financial report is a high-fidelity, high-resolution, high-quality information exchange mechanism. The report is a compendium of complex logical information required by statutory requirements and regulatory rules plus whatever management of an economic entity wants to voluntarily disclose. The report represents quantitative and qualitative information about the financial condition and financial performance of an economic entity. There are a number of different financial reporting schemes including: US GAAP, IFRS, IPSAS, GAS, FAS, etc.

Financial reports are symbolic systems[20]. A symbolic system is essentially a system built with symbols such as natural language, programming languages, mathematics, or formal logic. An interesting thing is that symbolic systems are understandable by both humans and by computers.

Paper-based, human-readable financial reports have been perfected over hundreds of years. High-quality financial reporting schemes such as US GAAP and IFRS have likewise been perfected over long periods of time.

---

[17] Charles Hoffman, CPA, Lean Six Sigma, http://www.xbrlsite.com/mastering/Part01_Chapter02.K_LeanSixSigma.pdf

[18] Michael Canic, *The Cost of Quality: The 1-10-100 Rule*, https://www.makingstrategyhappen.com/the-cost-of-quality-the-1-10-100-rule/

[19] Charles Hoffman, CPA, *Essence of Accounting*, http://www.xbrlsite.com/mastering/Part00_Chapter01.D_EssenceOfAccounting.pdf

[20] Symbolic Systems, http://xbrl.squarespace.com/journal/2020/8/26/symbolic-systems.html

XBRL-based, machine-readable financial reports need to work as good as or better than the paper-based financial reports that they will work along side of.

Financial reports are not uniform. Financial reports are not forms, they have variability. This consciously allowed variability is an essential, characteristic trait of robust reporting schemes such as US GAAP, IFRS, and others. This allowed variability contributes to the richness, high-fidelity, and high-resolution of reported financial information that is unique to an industry sector, a style of reporting, or an economic entity. This variability is a feature of such reporting schemes. Different reporting styles, different subtotals used to aggregate details, and using some specific approach given a set of allowed alternatives are examples of variability. Variability does not mean "arbitrary" or "random". There are known identifiable patterns.

Rules are used to articulate allowed variability and "channel" creators of reports in the right direction and therefore control variability, keeping the variability within standard limits. That keeps quality where it needs to be. Rules enable things like preventing a user from using a concept meant to represent one thing from unintentionally being used to represent something different. Further, the discipline of describing something in a form a computer algorithm can understand also assists you in understanding the world better; weeding out flaws in your understanding, myths, and misconceptions about accounting and reporting standards.

A **framework**[21] is a set of principles, assumptions, ideas, concepts, values, rules, laws, agreements, and practices that establishes the way something operates. A **theory**[22] is a tool for understanding, explaining, and making predictions about a system. A **system**[23] is a cohesive conglomeration of interrelated and interdependent parts that is either natural or man-made. Having the proper framework, theory, and system helps you understand digital financial reports and work with them effectively. Financial report is a logical system that is based on basic mathematics[24].

What is conspicuously missing from the minds of most professional accountants and auditors are a set of principles, a framework, and a theory relating to how to think about XBRL-based digital financial reports.

1.10. Business Use Case

Consider the following scenario. Two economic entities, A and B, each have information about their financial position and financial performance. They must communicate their information to an investor who is making investment decisions which will make use of the combined information so as to draw some conclusions. All three parties (economic entity A, economic entity B, investor) are using a common set of basic logical principles (facts, statements, deductive reasoning, inductive reasoning, etc.), common financial reporting standard concepts and relations (i.e. US GAAP, IFRS, IPSAS, etc.), and a common world view so they should be able to

[21] *Open Source Framework for Implementing XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2019/Framework/FrameworkEntitiesSummary.html

[22] Charles Hoffman, CPA, and Rene van Egmond, *Financial Report Semantics and Dynamics Theory*, http://xbrl.squarespace.com/fin-report-sem-dyn-theory/

[23] Charles Hoffman, CPA, *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*, http://xbrl.squarespace.com/journal/2019/12/30/special-theory-of-machine-based-automated-communication-of-s.html

[24] *The Mathematics of Double Entry Bookkeeping*, http://xbrl.squarespace.com/journal/2019/11/4/the-mathematics-of-double-entry-bookkeeping.html

communicate this information fully, so that any inferences which, say, the investor draws from economic entity A's information should also be derivable by economic entity A itself using basic logical principles, common financial reporting standards (concepts and relations), and common world view; and vice versa; and similarly for the investor and economic entity B.

1.11. Principles

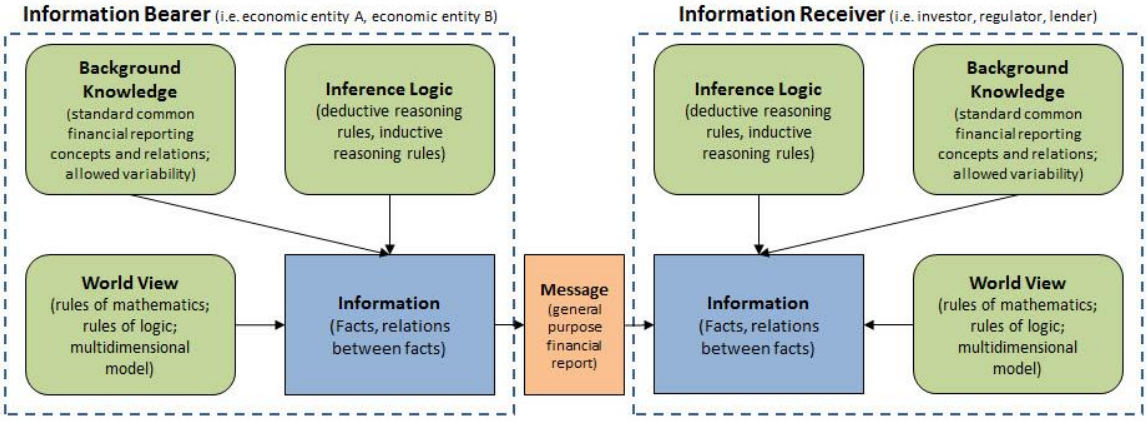
The following is a summary of fundamental principles which contribute to the framework of a general purpose financial report which are explained in more detail in *Principles*[25]:

1. A general purpose financial report is a high-fidelity, high-resolution, high-quality information exchange mechanism.
2. Creators of general purpose financial reports are information bearers.
3. Consumers of information from a general purpose financial report are information receivers.
4. Prudence dictates that using information from a general purpose financial report should not be a guessing game.
5. All general purpose financial report formats conveying information should convey the exact same meaning be that format paper, e-paper, or some machine readable format.
6. Explicitly stated information from information bearers or reliably derived information is preferable to requiring information receivers to make assumptions.
7. Double entry accounting enables processes that allow for the detection of information errors and to distinguish errors (unintentional) from fraud (intentional).
8. Catastrophic logical failures are to be avoided at all cost as they cause systems to completely fail.

1.12. Exchanging Information

Fundamentally, a general purpose financial report is a means of exchanging financial information as is explained in *Exchanging Complex Financial Information*[26]:

[25] Charles Hoffman, CPA, *Principles*, http://www.xbrlsite.com/mastering/Part01_Chapter02.B_Principles.pdf
[26] Charles Hoffman, CPA, *Exchanging Complex Financial Information*, http://www.xbrlsite.com/mastering/Part02_Chapter05.A_ExchangingComplexFinancialInformation.pdf

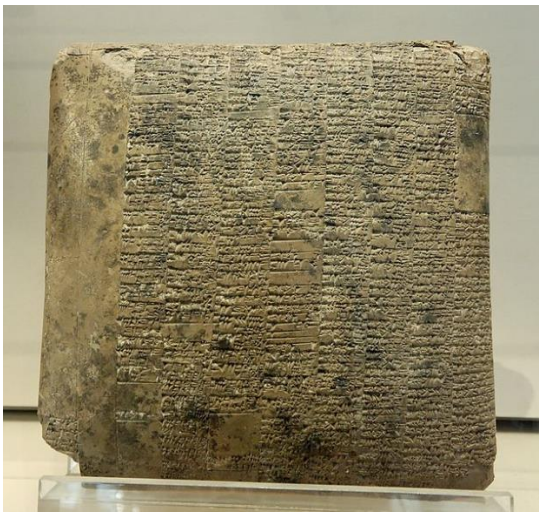


Special purpose financial reports work identically to general purpose financial reports except that they use a different financial reporting scheme[27].

1.13. Historical Financial Report

The role of general purpose financial report is to provide information about the financial position, financial condition, and changes in financial position that is useful to management and other stakeholders of an economic entity for economic resource allocation decisions. For information in a general purpose financial report to be useful, the report should be timely and free from material errors, omissions, and fraud.

General purpose financial reporting has existed for thousands of years in different forms. Below is an annual balance sheet of a State-owned farm which was drawn up by a scribe which details the account of materials and workdays for a basketry shop in 2040 BC[28]:



[27] *Modern Approach to Creating a Financial Reporting Scheme*, http://xbrl.squarespace.com/journal/2019/12/19/modern-approach-to-creating-a-financial-reporting-scheme.html

[28] Wikimedia, Annual balance sheet of a State-owned farm, drawn up by the scribe responsible for artisans: detailed account of materials and workdays for a basketry workshop. Clay, ca. 2040 BC., retrieved October 28, 2015, https://commons.wikimedia.org/wiki/File:Balance_sheet_Mesopotamia_Louvre_AO6036.jpg

Accounting existed before the invention of writing. Between 5,000 and 10,000 years ago farmers in Mesopotamia, where agriculture was born, used physical object to count crops and animals[29]. The distinction between types of crops or animals was made by using different types and shapes of objects. Then, in about 3200 BC, around 5,000 years ago, the first spreadsheet was invented.

These farmers began documenting information using clay tablets in the earliest form of human writing ever discovered called Cuneiform. They partitioned their clay tablet into rows, columns, and cells. These farmers used single-entry accounting. The spreadsheet below documents an account of barley distribution[30]:

In 1211 AD a bank in Florence was the first documented use of double-entry accounting[31]. Between 1299 AD and 1300 AD double-entry accounting came of age. In 1494 AD during the Renaissance, Venetian mathematician and Franciscan friar Luca Pacioli[32] published a book, *Summa de arithmetica, geometria. Proportioni et proportionalita* (Sum of Arithmetic, Geometry, Proportion and Proportionality)[33]. That book documented an approach to accounting now called double-entry bookkeeping and recommended that others use this approach. The approach allowed for better error detection and the ability to differentiate unintended errors from fraud. Accountants adopted that new approach.

---

[29] Denise Schmandt-Bessersat, *On the origins of writing*, YouTube.com, https://www.youtube.com/watch?v=kidWY-pJFb0

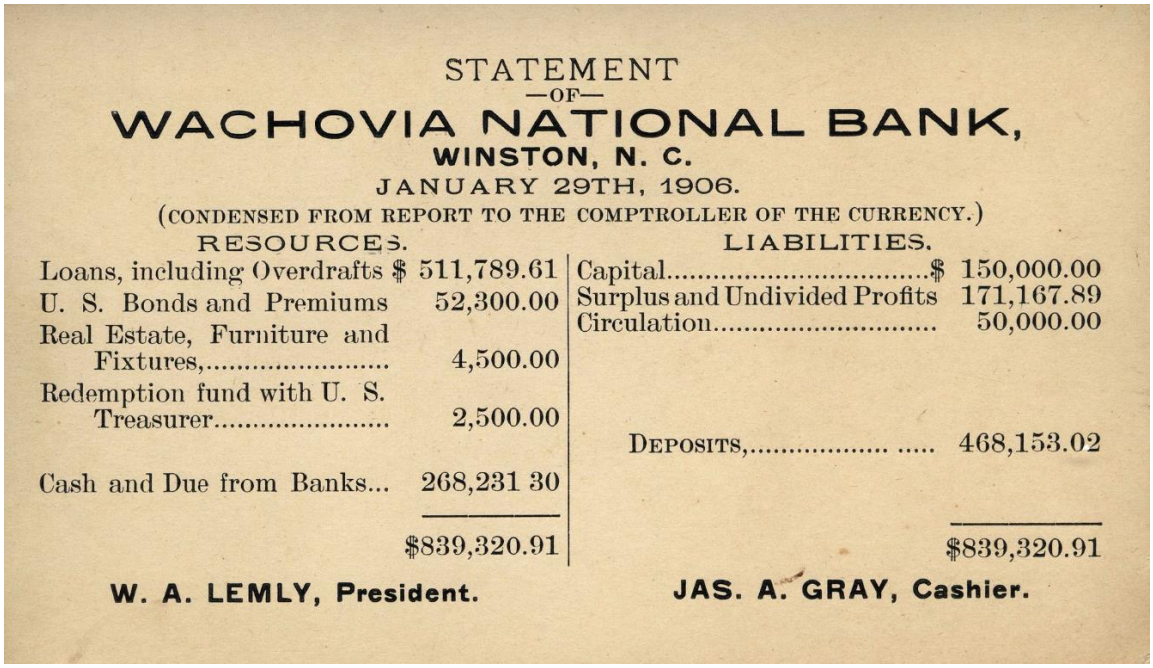
[30] Metropolitan Museum, Proto-Cuneiform tablet with seal impressions: administrative account of barley distribution with cylinder seal impression of a male figure, hunting dogs, and boars, https://www.metmuseum.org/art/collection/search/329081

[31] Geoffrky Alan Lee, *The Development of Italian Bookkeeping 1211–1300*, Wiley, https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-6281.1973.tb00183.x

[32] Wikipedia, Luca Pacioli, https://en.wikipedia.org/wiki/Luca_Pacioli

[33] Wikipedia, Summa de arithmetica, https://en.wikipedia.org/wiki/Summa_de_arithmetica

A significant advancement the general purpose financial statement[34] was the move from clay tablets to paper. Another significant advancement in financial reporting included the invention of the printing press, the copy machine, and word processing which made distributing information easier. Yet another advancement was the internet which enabled the broad distribution of financial information for literally pennies using e-paper formats such as PDF, HTML, and word processing document formats.



STATEMENT
—OF—
WACHOVIA NATIONAL BANK,
WINSTON, N. C.
JANUARY 29TH, 1906.
(CONDENSED FROM REPORT TO THE COMPTROLLER OF THE CURRENCY.)

RESOURCES.		LIABILITIES.	
Loans, including Overdrafts	$ 511,789.61	Capital	$ 150,000.00
U. S. Bonds and Premiums	52,300.00	Surplus and Undivided Profits	171,167.89
Real Estate, Furniture and Fixtures	4,500.00	Circulation	50,000.00
Redemption fund with U. S. Treasurer	2,500.00		
		DEPOSITS	468,153.02
Cash and Due from Banks	268,231 30		
	$839,320.91		$839,320.91

W. A. LEMLY, President. JAS. A. GRAY, Cashier.

For the past 100 years or so financial reporting has been mainly paper based. Only in the last 25-30 years have reports been created electronically in a word processor and then printed or saved to an electronic format or "e-paper" such as PDF or HTML and broadly distributed simultaneously anywhere on the planet.

But the information contained in PDF and HTML reports can still only be read by humans. Digital financial reporting, in contrast, makes much of this information readable by computers, vastly expanding the potential for automating processes for creating financial reports and analyzing information communicated by those financial reports. XBRL enables new modern approaches to creating a financial reporting schemes and new modern approaches to creating general purpose and special purpose financial statements[35].

Help from machines can reduce many mechanical tasks and therefore the time and also the costs of creating and consuming financial report information and improve information quality at the same time. Automation results in increased productivity.

[34] Wikipedia, *Financial Statement*, retrieved October 28, 2015, https://en.wikipedia.org/wiki/Financial_statement

[35] *Modern Approach to Creating a Financial Reporting Scheme*, http://xbrl.squarespace.com/journal/2019/12/19/modern-approach-to-creating-a-financial-reporting-scheme.html

1.14. The High Cost of Errors

Errors in financial reports are embarrassing if not detected and corrected, time-consuming to detect and fix, and are a waste of the finance team's resources. These errors are generally caused by mentalities that the manufacturing industry have solved using Lean Six Sigma[36] techniques which have not been adopted by accounting and financial reporting departments. The 1-10-100 rule points out that it costs $1 to prevent an error, $10 to correct an error after the error has been made, and $100 to deal with the consequences of errors.

An AccountingToday article, *Are the numbers right?*[37], points out that nearly 70% of respondents to a survey said that their organization has made a significant business decision based on inaccurate financial information. The reasons for the errors include human error, lack of automated controls and checks, and clunky spreadsheets and outdated processes.

Further, a Blackline study, *Mistrust in the Numbers*[38], points out a significant rift between how accurate CEOs and CFOs believe the numbers are as contrast to the quality of the numbers based on those who actually maintain the accounting information. Basically, higher level executives believe the numbers are far more accurate than they really are. The Blackline study also points out that human error is one of the biggest challenges in terms of attaining high quality and that errors in the information add weeks to processes.

With the volume of information increasing at increasing rates and the complexity of the information increasing at the same time; throwing more overworked humans at this situation will not solve the problem, it will only make the problem even worse.

1.15. Changing Old School Financial Report Creation Processes

Have the stars aligned, creating an opportunity for reinventing the outdated old school financial report creation processes? In an article, *Surety Data Standards: Is Manual Data Entry Dead*?[39], NASBP says "We're hammering the final nail in the coffin of manual data entry... more to come." Surety insurance companies use an XBRL taxonomy[40] to standardize work-in-progress reporting and reduced the process from 20 minutes to 3 seconds. NASBP says, "The gruesome (and grueling) days of painful re-keying of data may be coming to an end. Could data standards be the magic bullet?"

---

[36] Charles Hoffman, CPA, *Comprehensive Introduction to Lean Six Sigma*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.72_LeanSixSigma.pdf

[37] AccountingToday, Ranica Arrowsmith, *Are the numbers right?*, https://www.accountingtoday.com/news/finance-executives-worry-about-inaccurate-numbers

[38] Blackline, *Mistrust in the Numbers*, https://www.blackline.com/resources/whitepapers/mistrust-in-the-numbers

[39] NASBP, *Surety Data Standards: Is Manual Data Entry Dead?*, http://xbrl.squarespace.com/journal/2018/10/31/nasbp-surety-data-standards-is-manual-data-entry-dead.html

[40] *Gaining an Appreciation of XBRL's Power to Express Business Rules*, http://xbrl.squarespace.com/journal/2016/1/17/gaining-an-appreciation-of-xbrls-power-to-express-business-r.html

Data standards are one important piece of the puzzle to making the painful, monotonous, onerous, grueling, gruesome, and downright barbaric old-school practices use in accounting, reporting, auditing, and analysis more modern[41].

No one really disputes the fact that old school processes, practices, techniques, and procedures for creating external financial reports contain inefficiencies. For example, consider these four sources:

- **CFA Institute**[42]: calls for "...greater efficiencies within the current inefficient system" [of creating financial reports].
- **Gartner**[43]: "...average Fortune 1000 company used more than 800 spreadsheets to prepare its financial statements"
- **Ventana Research**[44]: "...for larger companies, assembling the periodic external reports typically is an inefficient and error-prone process."
- **PriceWaterhouseCoopers**[45]: "...old school manual processes..." and "commonly cut and pasted, rekeyed, or manually transferred into word processing and spreadsheet applications used for report assembly and review process steps"

What has changed?

1.16. Role of Structured Information as an Enabler

The answer is that one thing has changed which has enabled another thing. Each of the four organizations above hails XBRL or "structured data" as the way to make financial reporting processes more efficient.

That is not quite right. XBRL or structure data is not the change that will make processes more efficient; structured data enables the change to occur. If you don't understand the difference between structured and unstructured data, the video, *How XBRL Works*[46], helps you see what structured information is as contrast to unstructured information.

So, XBRL or structured data, is the enabler of a change, it is not the change itself. Again, then what changed? Well, two things changed.

First, the structured information lets a computer effectively address the individual pieces of a financial report. Because of the structure, software applications can do things with the individual pieces of the report. Basically, you can take measurements of structured information; that was impossible when financial reports were unstructured information.

---

[41] *Financial Transformation and the Modern Finance Platform*, http://xbrl.squarespace.com/journal/2018/11/2/financial-transformation-and-the-modern-finance-platform.html

[42] CFA Institute, *DATA AND TECHNOLOGY: TRANSFORMING THE FINANCIAL INFORMATION LANDSCAPE*, June 2016, http://www.cfapubs.org/doi/pdf/10.2469/ccb.v2016.n7.1

[43] Nigel Rayner, Neil Chandler, *XBRL Will Enhance Corporate Disclosure and Corporate Performance Management*, April 23, 2008, http://unstats.un.org/unsd/nationalaccount/workshops/2008/newyork/IG22.PDF

[44] Robert Kugel, Ventana Research, *Making XBRL Reporting Easy*, February 13, 2009, http://businessfinancemag.com/technology/ventana-research-making-xbrl-reporting-easy

[45] Mike Willis, PriceWaterhouseCoopers, *Disclosure management: Streamlining the Last Mile*, March 2012, https://www.pwc.com/gx/en/xbrl/pdf/pwc-streamlining-last-mile-report.pdf

[46] *How XBRL Works*, https://www.youtube.com/watch?v=nATJBPOiTxM

Second, because you can address or measure or otherwise work with the individual pieces that make up a financial report; more processes, procedures, and other tasks used in the report creation process can be automated using machine-based processes because the individual report pieces are identifiable.

Old school review processes are almost 100% manual. This is because old paper and e-paper financial report formats were unstructured. If information is structured, it does not have to be this way. Tasks can be automated leveraging the structured nature of the information. On the other hand, there is ZERO probability that 100% of the financial report creation process will be automated. To think that would be absurd.

What percentage can effectively be automated though? Certainly, it is some percentage. That percentage is greater than 1%. Is it 10%? Is it 20%? Is it 50%? More than 50%? Time will reveal the answer to that question.

Further, there will no doubt be quality improvements will also be achievable. There is no way that a process that is nearly 100% manual can be of perfect quality. While current report creation processes throw many hours of high-quality and expensive effort at processes to detect and correct errors; humans make mistakes. So, there is some level of quality problems that exist in the current old school processes that cause errors. But you cannot see those problems or measure the problems because, you guessed it, the current financial reports are unstructured and you cannot address the pieces of a report. Just because you cannot measure quality problems does not mean that quality problems do not exist. They exist.

Interestingly, the discipline of describing something in a form a computer algorithm can understand assists you in understanding the world better, weeding out myths and misconceptions. This process will help economic entities creating reports improve report quality but it will also help standards setters and regulators create clearer financial reporting rules.

1.17. Benefits of Digital Financial Reporting

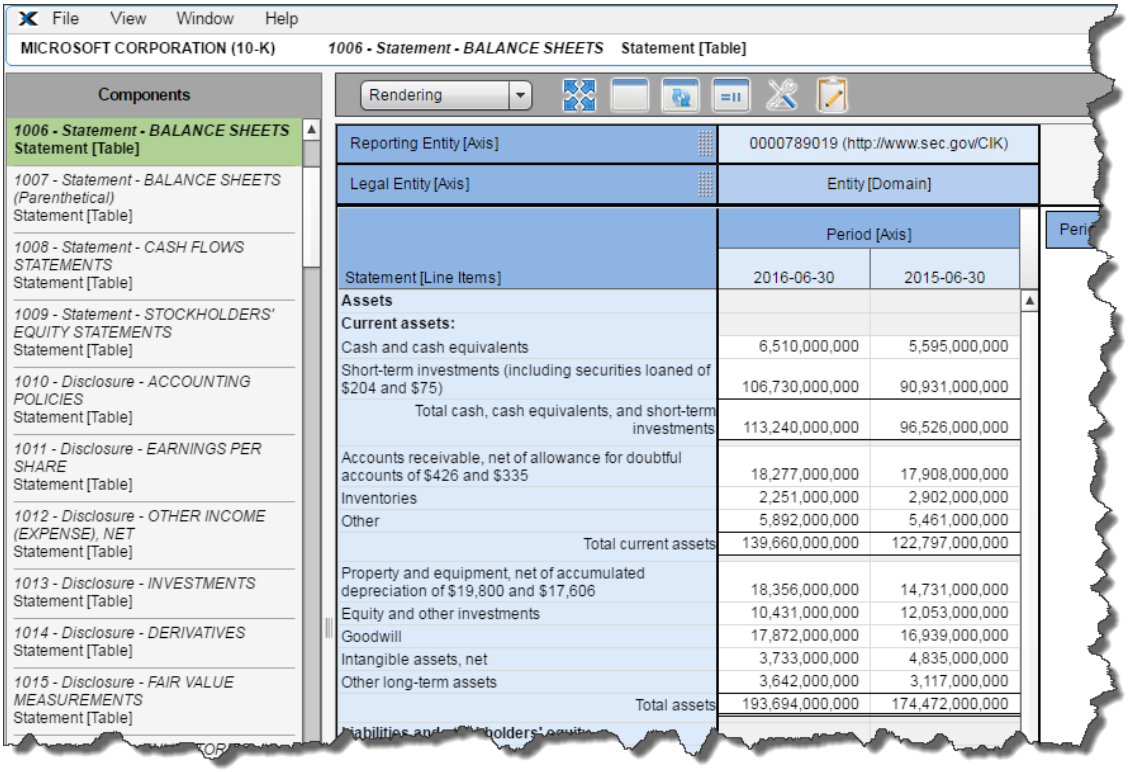
Many professional accountants don't have the background knowledge to understand how to make computers perform work effectively[47]. But these skills can be learned.

With machine readability of financial reports computers can read the reported financial information, truly *understand* that information, and help users of the report make use of the reported information. But not only users of reported financial information will benefit. Creators of reports will also benefit. Computers can also help during the report creation process. For example, computers can compare reported information to mandated disclosure rules and make sure the report's creator complied with those rules.

Below is a rendering of the balance sheet of Microsoft Corporation viewable in a free publicly available XBRL-based digital financial report viewer application[48]:

[47] *Computer Empathy*, http://xbrlsite.azurewebsites.net/2018/Library/ComputerEmpathy.pdf

[48] You can try out an XBRL-based digital financial report here: https://edgardashboard.xbrlcloud.com/flex/viewer/XBRLViewer.html#instance=http://www.sec.gov/Archives/edgar/data/789019/000119312516662209/msft-20160630.xml



Here is a summary of some of the general benefits of structured digital financial reporting and machine readability of that information:

- **Increased report flexibility** - reported information can be easily and reliably reconfigured, reformatted and otherwise repurposed without rekeying to suit the specific needs of an analyst or regulator.
- **Reliable repurposing of information and improved communication** - ambiguity is reduced because for a computer to make use of the information, that information cannot be ambiguous. Going through the process of making the information easy for a computer to understand also makes it easier for humans to communicate more effectively and helps them bring into consciousness ambiguities that exist in the current process but are unconscious of[49].
- **Reliable process automation** - processes can be reliably automated because computers can reliably move information through the workflow. Linking digital financial information together based on the meaning of the information can be much more reliable than trying to link physical locations within spreadsheets, which commonly change.
- **Increased software adaptability** - software can easily adapt itself to specific reporting scenarios and user preferences because it understands the information it is working with; rather than having to get software developers involved and program to make changes, accounting professionals adjust metadata themselves to make adjustments they require.

[49] *Differentiating Alternatives from Ambiguity in US GAAP,* http://xbrl.squarespace.com/journal/2015/4/22/differentiating-alternatives-from-ambiguity-in-us-gaap.html

This is not to say that humans will no longer be involved in creating or consuming financial reports. Clearly, machines will never be able to exercise judgment, which will remain something only humans can do. But to understand exactly what computers will be able to do, will never be able to do and how exactly to successfully get a computer to perform work; you need to understand a little bit about how to harness the power of a computer.

Just like a calculator helps a professional accountant do math faster and more reliably, software will augment the skills of professional accountants and help them with the many tasks related to creating financial reports.

No magic is involved here. Rather, digital financial reporting relies on well-understood information technology practices, agreement on standard technical syntaxes, and carefully and clear articulation of already agreed-upon financial reporting rules articulated in a manner that computers can effectively make use of.

1.18. Digital Financial Reporting Alternative

As we have said, the general purpose financial statement (or financial report) has existed for over two millennium. Formats for general purpose financial statements have included clay, paper, word processor documents such as Microsoft Word, PDF, and HTML. The common thread that all these reports have is that a machine cannot read these reports because the reports are unstructured.

The institution of accountancy needs to create a digital, or structured, version of the general purpose financial statement which is machine-readable.

With digital books, maps, photos, films, music, blueprints, etc.; what about the digital general purpose financial statement does not make sense? Perhaps this is stating the obvious.

The digital general purpose financial report is an improvement that helps move the institution of accountancy forward, providing an improvement to that institution. Given today's increasing volume of financial information, complexity of financial information, and importance of financial information; it makes perfect sense to provide such a digital alternative or option.

Financial analysis has been digital for many years; first via the electronic spreadsheet and now with a multitude of options including business intelligence (BI) or other sorts of analysis software.

Structured financial reporting that is both human-readable and machine-readable and based on the global standard XBRL completely changes the paradigm of financial reporting.

In later sections we will show you in detail how a digital financial report works, a few new skills professional accountants must acquire to work within this new digital financial report paradigm, and understand new tasks that machine-based processes can perform for professional accountants.

The next section provides a brief description of what might be possible by looking at another document that has already made this transition to digital: the blueprint.

1.19. Financial Report is a Logical System

So how do you make a financial report machine readable? How do you get computers to perform useful work? The general answer is engineering. More specifically, you describe the financial report in terms that a computer can relate to and work with[50]. That is done by creating an ontology-like thing which describes the logical system of terms, relations, and assertions that make up a financial report[51]. Software then uses this machine-readable conceptualization to help you create XBRL-based financial reports or consume the information conveyed by such reports.

We will get into all these details in later sections. But now, let's take a look at another industry that went through this digital transition and what we can learn from that transition.

2. Components of a Knowledge Based System

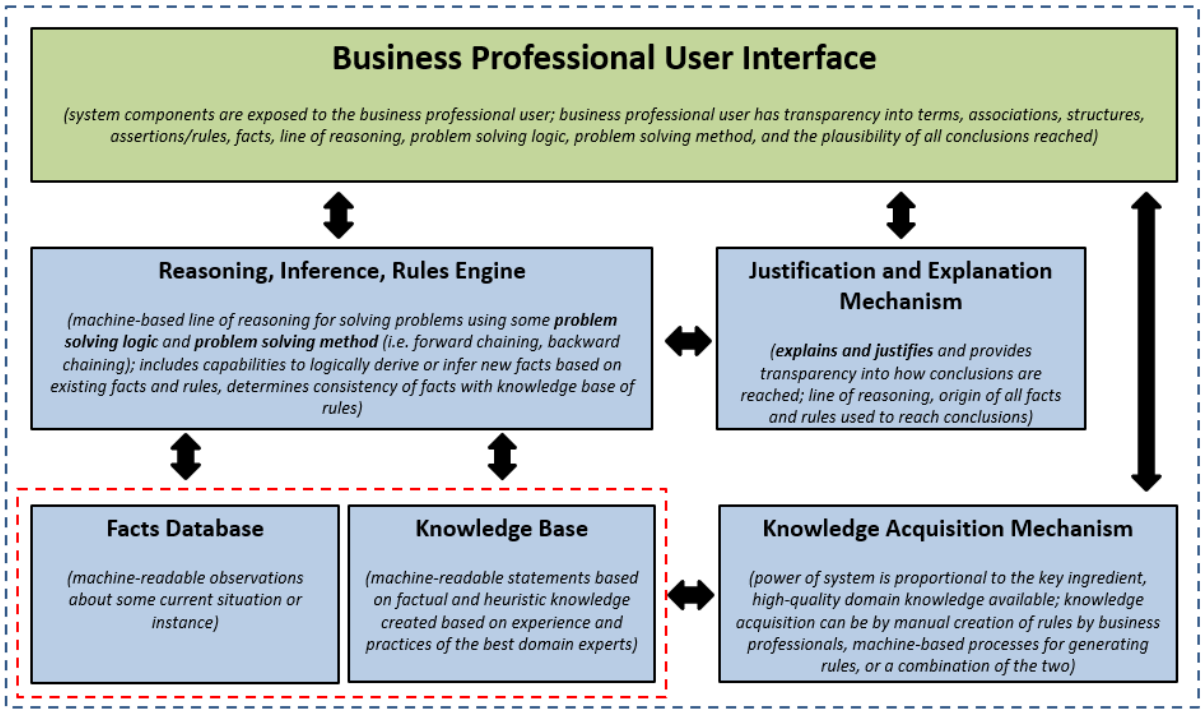
To create augmented intelligence, you build a knowledge based system. Information is stored in a fact database and a knowledge base. The system applies problem solving logic using a problem-solving method. The knowledge based system supplies an explanation and justification mechanism to help users understand the line of reasoning used to reach conclusions. The system then presents that information back to the user.

Nothing is a "black box". The origin of information used to reach conclusions is always apparent.

The following is a summary of the components of a knowledge based system. Each of the components shown in the graphic above will be described and examples provided in the following sections.

[50] Charles Hoffman, CPA, *Artificial Intelligence and Knowledge Engineering Basics in a Nutshell*, http://xbrlsite.azurewebsites.net/2019/Library/KnowledgeEngineeringInNutShell.pdf

[51] Charles Hoffman, CPA, *Enhanced Description of Ontology-like Thing*, http://xbrl.squarespace.com/journal/2019/7/19/enhanced-description-of-ontology-like-thing.html

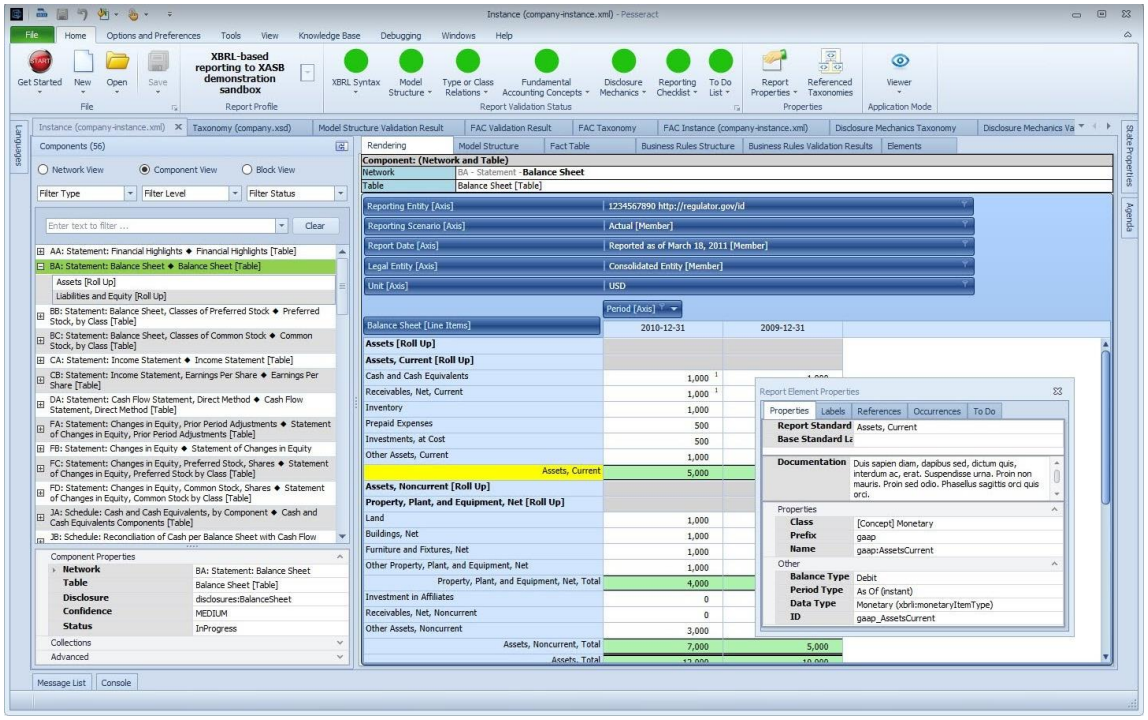


2.1. Business Professional User Interface

The business professional user interface are the components that are exposed to the business professional using the system. Business professionals need transparency as to the terms, associations, structures, rules, facts, line of reasoning, problem solving logic, problem solving method, and the plausibility of all conclusions reached by the system.

The following is one of a number of screen shots[52] of the working proof of concept software application Pesseract which provides an example of a user interface with which a business professional would likely interact:

[52] Additional Pesseract User Interface Screenshots, https://photos.app.goo.gl/cWeZYaMBEbmSSm7v8



The user interface is non-technical requiring only business and accounting knowledge to effectively understand the software application and how to use it.

2.2. Justification and Explanation Mechanism

The justification and explanation mechanisms of the software application explains and justifies and provides transparency into how conclusions are reached by the software application. The rules used, facts used, line of reasoning, and origin of all facts are knowable to the business user of the software. There is transparency into all conclusions that are reached by the software application. Nothing is a black box.

Below you see the fundamental accounting concept relations continuity cross check verification checks provided by XBRL Cloud's Evidence Package[53] which is a review tool that can be used to verify XBRL-based financial reports:

Balance Sheet [Line Items]	Period [Axis] 2017-06-30 Fact Value	Origin
Assets [Roll Up]		
Current Assets	159,851,000,000	fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]
Noncurrent Assets	81,235,000,000	fac:NoncurrentAssets[81,235,000,000] = fac:Assets[us-gaap:Assets[241,086,000,000]] - fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]
Assets	241,086,000,000	fac:Assets[us-gaap:Assets[241,086,000,000]]

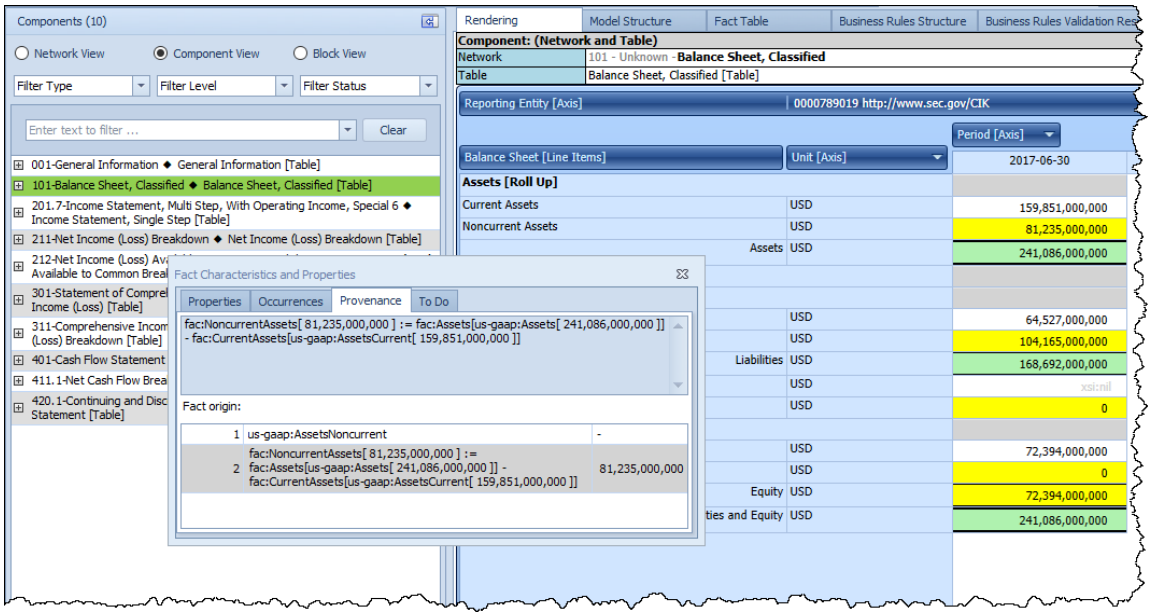
Fact determination of fac:NoncurrentAssets

1	us-gaap:AssetsNoncurrent	-
2	fac:Assets[us-gaap:Assets[241,086,000,000]] - fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]	81,235,000,000

[53] XBRL Cloud Evidence Package, http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/evidence-package/USFACRenderingSummary.html

If you look at the fundamental accounting concept relations continuity cross check verification results you see that the business user can trace each fact two it's origin, understand all rules used by the software to reach conclusions, etc.

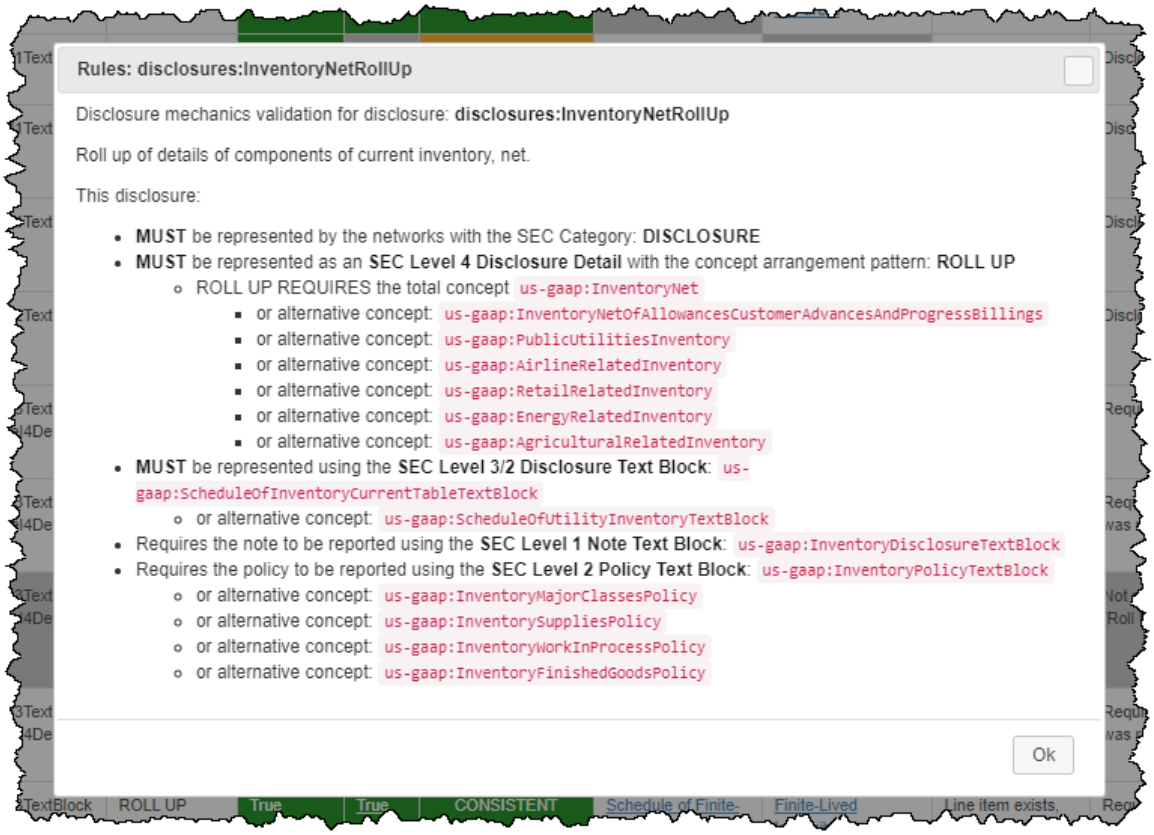
Pesseract provides similar functionality:



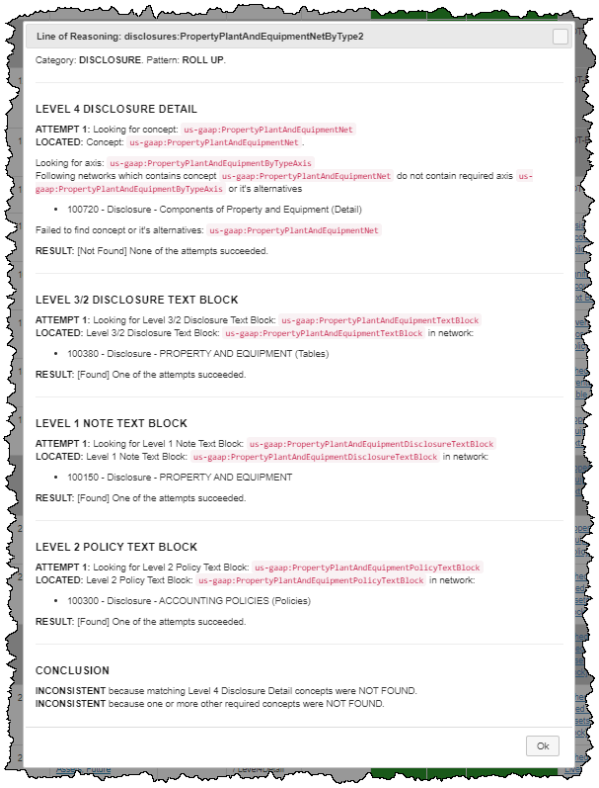
XBRL Cloud's Disclosure Mechanics and Reporting Checklist[54] provides the rules used, line of reasoning used, and conclusions reached for determining if a disclosure is structured consistent with its expected specification:

Disclosure mechanics rules:

[54] XBRL Cloud Disclosure Mechanics and Reporting Checklist, http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html



Line of reasoning:

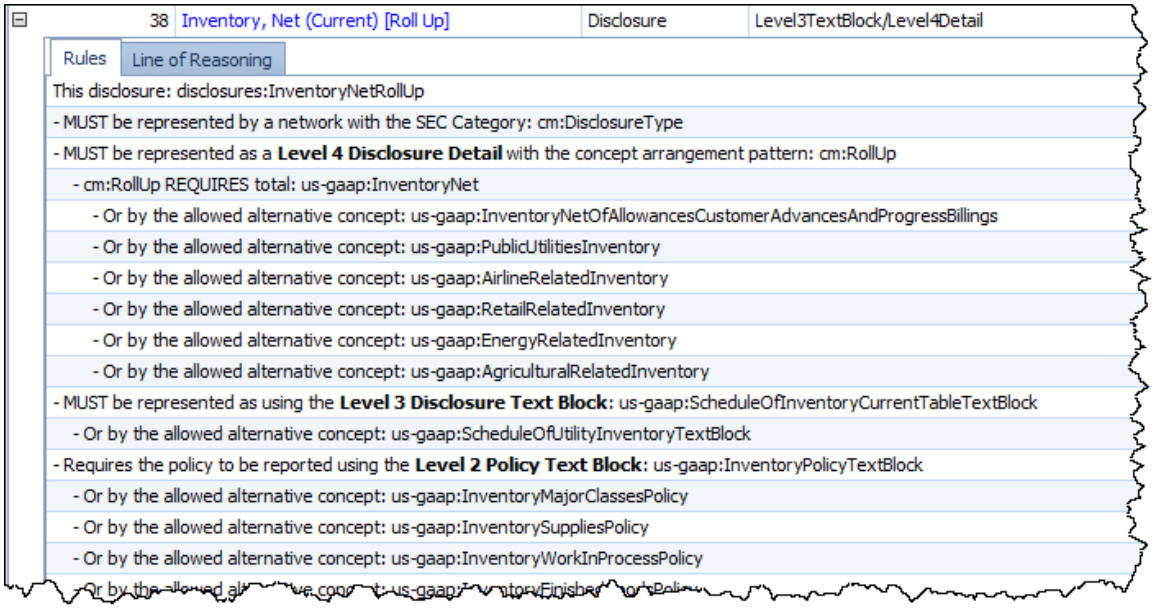


Conclusions reached:

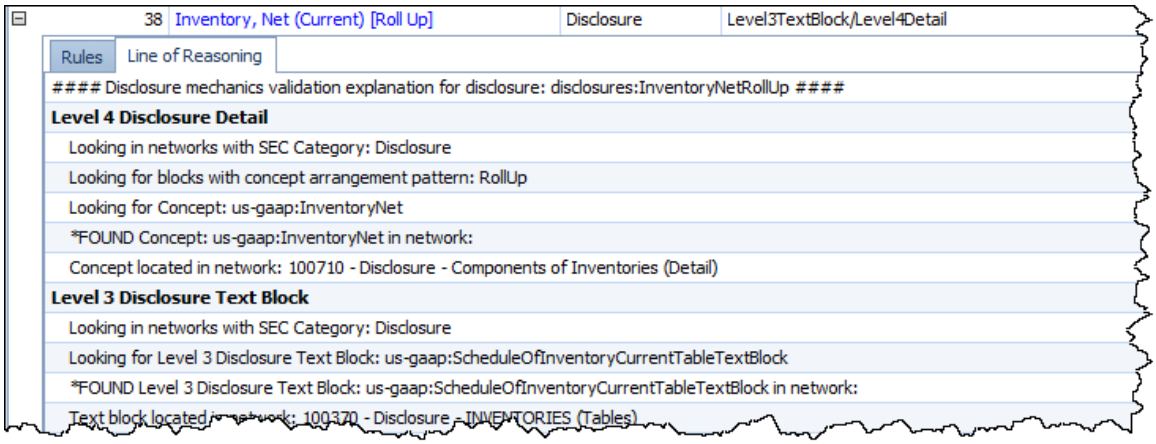
#	Disclosure	Category	Level	Pattern	Applicable	Found	Disclosure Consistent	Representation Concept [TEXT BLOCK]	Representation Concept [DETAIL]	Checklist Category	Reason
1	Document Information [Hierarchy]	DOCUMENT	Level4Detail	HIERARCHY	True	True	CONSISTENT	NOT-EXPECTED	Document Fiscal Period Focus	Required disclosure	Disclosure always required
2	Document and Entity Information [Hierarchy]	DOCUMENT	Level4Detail	HIERARCHY	False	True	CONSISTENT	NOT-EXPECTED	Entity Registrant Name	Alternative representation	Not necessary, satisfied by Document Information [Hierarchy] disclosure
3	Entity Information, by Legal Entity [Hierarchy]	DOCUMENT	Level4Detail	HIERARCHY	True	True	CONSISTENT	NOT-EXPECTED	Entity Registrant Name	Required disclosure	Disclosure always required
4	Document and Entity Information [Hierarchy]	DOCUMENT	Level4Detail	HIERARCHY	False	True	CONSISTENT	NOT-EXPECTED	Entity Registrant Name	Alternative representation	Not necessary, satisfied by Entity Information, by Legal Entity [Hierarchy] disclosure
5	Balance Sheet	STATEMENT	Level4Detail	COMPONENT	True	True	CONSISTENT	NOT-EXPECTED	NOT-EXPECTED	Required disclosure	Disclosure always required, satisfied by Assets [Roll Up] and Liabilities and Equity [Roll Up]
6	Assets [Roll Up]	STATEMENT	Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Assets	Part of disclosure	Disclosure always required
7	Liabilities and Equity [Roll Up]	STATEMENT	Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Liabilities and Equity	Part of disclosure	Disclosure always required
8	Income Statement, by Legal Entity [Roll Up]	STATEMENT	Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Net Income (Loss) Attributable to Parent	Required disclosure	Disclosure always required
9	Statement of Income and Comprehensive Income [Roll Up]	DISCLOSURE	Level4Detail	ROLL UP	False	True	CONSISTENT	NOT-EXPECTED	Net Income (Loss) Attributable to Parent	Alternative representation	Not necessary, satisfied by Income Statement, by Legal Entity [Roll Up] disclosure
10	Statement of Comprehensive Income [Roll Up]	STATEMENT	Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Comprehensive Income (Loss), Net of Tax, Attributable to Parent	Required disclosure	Disclosure always required
[illegible]	[illegible]	DISCLOSURE	Level4Detail	ROLL UP	False	True	CONSISTENT	NOT-EXPECTED	[illegible]	[illegible]	[illegible]

Similar functionality is offered by Pesseract:

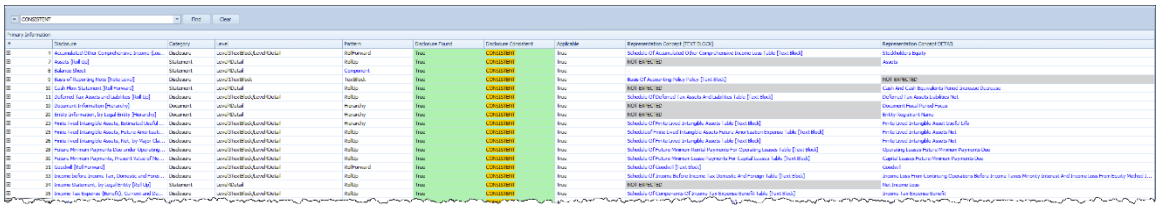
Disclosure mechanics rules:



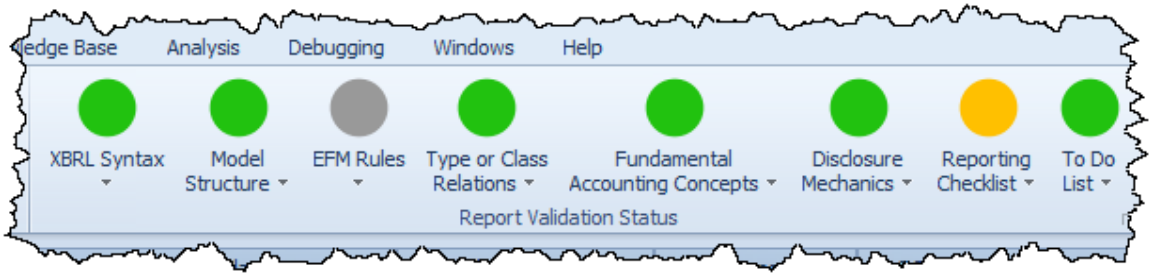
Line of reasoning:



Conclusions reached[55]:



Similar mechanisms exist for all other categories of rules verified using the method that has been created which leverages OMG's Standard Business Report Model (SBRM)[56]:



2.3. Reasoning, Inference, Rules Engine

The reasoning, inference, and rule engine use the machine-based rules, a line of reasoning for solving problems using some problem solving logic and problem solving method (i.e. forward chaining, backward chaining) to reach conclusions about facts and all other statements made within the logical system. This includes capabilities to logically derive or infer new facts or other information based on existing facts and rules. It also includes the capability to determine consistency of facts with the systems knowledge base of rules.

The following approaches tend to be capable of meeting the needs of this method:

[55] Pesseract disclosure mechanics verification of 94.8% of all 124 disclosures verified, http://xbrlsite.azurewebsites.net/2020/Prototype/Microsoft/Microsoft2017_Discovery.jpg

[56] SBRM Progress Report, http://xbrl.squarespace.com/journal/2020/1/30/sbrm-progress-report.html

- **Ontology + Rules**: For example, OWL[57] (or SWRL[58]) + SHACL[59] + RDF[60] (or N3[61]) provide sufficient fragments of first order logic. (Some call this Modern Symbolic AI[62])
- **Modern Prolog**: Prolog such as SWI Prolog[63] or Scryer Prolog[64] seem to have all of the necessary functionality. The up side is that there are a lot of Prolog implementations[65]. The down side is that none of these Prologs can call itself "the standard". Each has pros and cons. Prolog interoperates with relational (SQL) databases.
- **ISO Prolog**: ISO has created a standard Prolog[66]. ISO Prolog can be regarded as a subset of Full Prolog. There is solid motivation for implementations to support ISO Prolog as the international standard Prolog, many already do to one degree or another.
- **Datalog**: Datalog[67], or "function-free Horn Logic", is more tractable than Horn Logic[68] (Pure Prolog) and ISP Prolog (Full Prolog). RuleML.org points out[69], "Datalog is the language in the intersection of SQL and Prolog. It can thus be considered as the subset of logic programming needed for representing the information of relational databases, including (recursive) views." So Datalog interoperates with relational databases.
- **PSOA RuleML**: PSOA[70] (Positional-Slotted Object-Applicative) RuleML is a multi-paradigm, particularly graph-relational, data and rule language. PSOA interoperates with graph and relational databases. RuleML.org points out[71], "PSOA RuleML's databases (fact bases) generalize the instance level of Graph and Relational Databases; its knowledge bases complement facts by rules for deductive retrieval (extending the Datalog-level, function-free expressiveness of Deductive Databases to the Horn-logic expressiveness of Logic Programming), interoperation, and reasoning, as well as for optionally emulating part of the schema level."

---

[57] W3C, *OWL*, https://www.w3.org/TR/owl2-overview/

[58] W3C, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, https://www.w3.org/Submission/SWRL/

[59] W3C, *SHACL*, https://www.w3.org/TR/shacl/

[60] W3C, RDF, https://www.w3.org/RDF/

[61] W3C, *Notation3 (N3): A readable RDF syntax*, https://www.w3.org/TeamSubmission/n3/

[62] Shawn Riley, *Modern Symbolic AI in 2020*, https://medium.com/@shawn.p.riley/modern-symbolic-ai-in-2020-dfcc27abbc5c

[63] SWI Prolog, https://www.swi-prolog.org/

[64] Scryer Prolog, https://github.com/mthom/scryer-prolog

[65] Wikipedia, *Comparison of Prolog Implementations*, https://en.wikipedia.org/wiki/Comparison_of_Prolog_implementations

[66] ISO, *ISO Prolog*, https://www.iso.org/standard/21413.html

[67] Wikipedia, *Datalog*, https://en.wikipedia.org/wiki/Datalog

[68] Wikipedia, *Horn Logic*, https://en.wikipedia.org/wiki/Horn_clause

[69] RuleML.org, http://ruleml.org/papers/Primer/RuleMLPrimer2012-08-09/RuleMLPrimer-p3-2012-08-09.html

[70] RuleML.org, *PSOA*, http://wiki.ruleml.org/index.php/PSOA_RuleML

[71] RuleML.org, *PSOA RuleML Bridges Graph and Relational Databases*, https://wiki.ruleml.org/index.php/PSOA_RuleML_Bridges_Graph_and_Relational_Databases

- **GQL/Cypher**: GQL[72] is an ISO project[73] to create a global standard query language (like SQL) for graph databases, graph query language. Open Cypher[74] which is based on Cypher is the query language of Neo4j.
- **SQL + More**: While it is proven[75] that you can store XBRL-based information in a relational database; you have to add functionality to process the information. Essentially, you have to construct a rules engine to process the information and prove the system is properly functioning. This is very possible but tends to not be very efficient.
- **XBRL + SBRM + More**: XBRL[76] is an open standard technical syntax published by XBRL International, SBRM[77] is a forthcoming standard to be published by OMG that formalizes a logical conceptualization of a business report. While XBRL provides the functionality to represent all that is needed to express knowledge and much of what is necessary to process that knowledge and prove the knowledge is represented correctly. However, certain specific processing is missing that must be supplemented to create a complete system. As such, that additional processing logic must be provided.

There are undoubtedly other logic engines that can be used to process XBRL-based digital financial reports. Other completely different approaches such as the decision model approach[78] could possibly be used but would need to include an ontology-type component. Any syntax used should be 100% convertible to all other syntaxes and be able to round tripped back into the original syntax. Then, you could switch between whatever approach you wanted.

The following are examples of the sorts of reasoning, inference, and rules engine which could be used[79]:

[72] GQL Standards.org, *GQL Standard*, https://www.gqlstandards.org/
[73] Wikipedia, *GQL Graph Query Language*, https://en.wikipedia.org/wiki/GQL_Graph_Query_Language
[74] OpenCypher.org, *Open Cypher*, https://www.opencypher.org/
[75] Proof representation, http://xbrlsite.azurewebsites.net/2020/master/proof/index.html
[76] XBRL International, https://www.xbrl.org/
[77] OMG, SBRM, https://www.omg.org/intro/SBRM.pdf
[78] Wikipedia, *Decision Model*, https://en.wikipedia.org/wiki/Decision_model
[79] Rules Engine Comparison, http://xbrlsite.azurewebsites.net/2020/Library/RulesEngineComparison.jpg

Business Rules Processor	For more information	Approach	Information Syntax/Format	XBRL Processor	Formula Processor	Supports Inference	Problem Solving Method	Probabilistic Reasoning (Modal	State Machine	General Problem Solving Logic Capabilities	Fact Database	Knowledge Base	Rule Creation Interface (XBRL Taxonomy, XBRL Formula, Other)	g of Business Report Conceptual Model	g of Financial Report Conceptual	Explanation Mechanism (Transparency into Line of Reasoning and	Knowledge Acquisition Mechanisms
Arelle (Open source API level interface)	http://arelle.org/	XBRL	Standard XBRL	Yes	Yes	No	Sequential	No	No	INCOMPLETE, Limited to XBRL Formula	XBRL Instance	XBRL Taxonomy	None provided	No	No	Not provided, can be created	Manual
XBRL Development Tools (Altova)	https://www.altova.com/xbrl-tools	XBRL	Standard XBRL	Yes	Yes	No	Sequential	No	No	INCOMPLETE, Limited to XBRL Formula	RaptorXML+XBRL Server	RaptorXML+XBRL Server	XBRL Specific but oriented to technical users	No	No	Not provided, can be created	Manual
Interstage Xwand (Fujitsu)	http://www.fujitsu.com/global/products/software/middleware/application-	XBRL	Standard XBRL	Yes	Yes	No	Unknown	No	No	INCOMPLETE, Limited to XBRL Formula	XBRL Instance	XBRL Taxonomy	XBRL Specific but oriented to technical users	No	No	Not provided, can be created	Manual
Sphinx (CoreFiling)	https://www.corefiling.com/products/sphinx/	XBRL	Standard XBRL	Yes	Yes	No	Sequential	No	No	INCOMPLETE, Limited to XBRL Formula	XBRL Instance	XBRL Taxonomy	XBRL Specific but oriented to technical users	No	No	Unknown	Manual
Clean Score (XBRL Cloud)	https://www.xbrlcloud.com/cleanscore.html	XBRL-based Business Reporting	Profile based Standard	Yes	Yes	Yes	Sequential	No	No	GOOD (Subset of RuleLog)	XML Infoset stored in file system	XML Infoset stored in file system	None provided	Yes	Yes	Good, usable by business professionals	Manual
Pesseract Knowledge Based Financial Report Creation System	http://pesseract.azurewebsites.net/	XBRL-based Business Reporting	Profile based Standard	Yes	No	Yes	Forward chaining	No	Yes	GOOD (Subset of RuleLog)	XML Infoset stored in file system	XML Infoset stored in file system	None provided at present time, will be business user oriented	Yes	Yes	Good, usable by business professionals	Manual
SWI PROLOG	https://www.swi-prolog.org/	Logic Programming	Open source defacto standard	No	No	Yes	Backward chaining (can do forward)	No	No	VERY GOOD (Turing machine; can be limited to DATALOG)	Proprietary or general format	Defacto standard PROLOG format	None provided	No	No	Unknown	Manual
CLIPS	http://www.clipsrules.net/	Logic Programming	Open source, based on PROLOG	No	No	Yes	Forward chaining	No	No	VERY GOOD (Turing machine; can be limited to DATALOG)	Proprietary or general format	Unknown	None provided	No	No	Unknown	Manual
FlexRule Business Logic Platform	http://www.flexrule.com/solution/	Business Rules	Proprietary or general format	No	No	Yes	Forward chaining	No	No	BETTER (Larger subset of RuleLog)	Proprietary or general format	Proprietary or general format	Comprehensive but oriented to technical users	No	No	Unknown	Manual
InRule (InRule Technologies)	http://www.inrule.com/products/inrule/	Business Rules	Proprietary or general format	No	No	Yes	Forward chaining	No	No	BETTER (Larger subset of RuleLog)	Proprietary format	Proprietary format	Oriented toward non-technical users and business professionals	No	No	Unknown	Manual
Smarts (Sparkling Logic)	https://www.sparklinglogic.com/smarts-decision-manager/	Business Rules	Proprietary or general format	No	No	Yes	Forward chaining	No	No	BETTER (Larger subset of RuleLog)	Proprietary or general format	Proprietary or general format	Comprehensive but oriented to technical users	No	No	Unknown	Manual
Fluent Editor (Cognitum)	http://www.cognitum.eu/semantics/FluentEditor/	Semantic Web Stack	Standard RDF, OWL, SWRL	No	No	Yes	Forward chaining	No	No	BETTER (Larger subset of RuleLog)	RDF stored in file system	RDF, OWL, SWRL stored in file system (Semantic	Comprehensive but oriented to technical users	No	No	Unknown	Manual
TopBraid Platform (TopQuadrant)	https://www.topquadrant.com/technology/topbraid-platform-overview/	Semantic Web Stack	Standard RDF, SHACL, RDFS, OWL	No	No	Yes	Forward and Backward	Yes	No	BEST (RuleLog plus)	RDF triple store repository	SHACL, RDFS, OWL, SPIN stored in file system (Semantic	Comprehensive but oriented to technical users; rule creation templates usable by	No	No	Good, usable by business professionals	Manual or Automated
Enterprise Data Governance (EDG); TopQuadrant	https://www.topquadrant.com/products/topbraid-enterprise-data-governance/	Semantic Web Stack															
Ergo Logic System (Coherent Knowledge)	http://coherentknowledge.com/financial-domain-application/	Semantic Web Stack	Standard RDF, RIF	No	No	Yes	Forward and Backward	Yes	No	BEST (RuleLog plus)	RDF stored in file system	RDF, OWL, RIF stored in file system (Semantic	Comprehensive but oriented to technical users	No	No	Good, electronic audit trail with provenance usable by business	Manual or Automated

Currently, while no one single rules engine can process 100% of what is required to be processed[80] a financial report or an accounting process automation workflow[81]; the above processors can be combined to achieve 100% of the capabilities which are necessary[82].

2.4. Fact Database

The fact database is essentially equivalent to the facts that are reported within an XBRL instance. The separation of the facts reported from the knowledge base of rules that support those reported facts is somewhat arbitrary.

There are many approaches to storing facts within a database[83]. Each approach has a set of PROS and CONS; no approach is 100% the best or 100% the worst. What appear to be the most viable alternatives include:

- **SQL database**: These are the most pervasive and the most popular today.
- **RDF triple store**: These are popular for working with the W3C Semantic Web Stack. These are sometimes implemented within a SQL database.
- **Graph database**: Graph databases such as Neo4j[84] are increasing in popularity, standard query languages are being developed like Cypher[85].

[80] Charles Hoffman, CPA, *Chain of Capabilities Necessary to Automate Accounting Processes*, http://xbrlsite.azurewebsites.net/2018/Library/ChainOfCapabilities.pdf

[81] Charles Hoffman, CPA, et. al., *Understanding Digital*, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingDigital.pdf

[82] Continuous Accounting Workflow Prototype, http://xbrlsite.azurewebsites.net/2020/master/continuous-accounting/index.html

[83] Understanding Database/Query Options (Part 2), http://xbrl.squarespace.com/journal/2014/4/27/understanding-databasequery-options-part-2.html

[84] Neo4j, http://xbrl.squarespace.com/journal/2020/7/7/neo4j.html

[85] Cypher, https://www.opencypher.org/

- **NOSQL databases**: NOSQL databases such as MondoDB are increasing in popularity because they require no schema which can be a feature or a bug depending upon whether you desire a database schema.
- **DATOMIC**: Datomic[86] is a fact database or cell store[87] that has a built in DATALOG rules engine.

What is the right database alternative to use? That is a decision that should be made by qualified technical professionals.

- ***Knowledge Base***

The knowledge base is essentially equivalent to the information that supports reported facts that is represented within XBRL taxonomy schemas, XBRL linkbases, and other information provided in the form of XBRL Formulas. The knowledge base is essentially machine-readable statements based on factual and heuristic knowledge created based on experience and practices of the best domain experts.

The following are example knowledge bases for several financial reporting schemes:

- US GAAP[88]
- IFRS[89]
- IPSAS[90]
- FRF for SMEs[91]
- US GAAP Not-for-Profit[92]

Other testing, prototype, and other such XBRL-based financial reporting schemes were represented in order to collect information which could yield information useful to create on framework for representing all financial reporting schemes. That information is summarized in *Mastering XBRL-based Digital Financial Reporting*[93].

In essence, it is possible to represent any financial reporting scheme[94] using the notion of profiles[95] to adjust for any minor differences between how each financial reporting scheme chooses to implement XBRL-based digital financial reporting.

But how do you get the knowledge that ends up in a knowledge base? You need some sort of mechanism for acquiring knowledge.

---

[86] Datomic Cloud, https://www.datomic.com/
[87] Ghislain Fourny, PhD, *Cell Stores*, https://arxiv.org/pdf/1410.0600.pdf
[88] US GAAP financial reporting scheme, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/documentation/Index.html
[89] IFRS financial reporting scheme, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ifrs/documentation/Index.html
[90] IPSAS financial reporting scheme (prototype), http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ipsas/documentation/Index.html
[91] FRF for SMEs financial reporting scheme, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/frf-sme/documentation/Index.html
[92] US GAAP Not-for-Profit financial reporting scheme, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/nfp/documentation/Index.html
[93] *Mastering XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2020/master/
[94] *Comparison of Financial Reporting Schemes High Level Concepts*, http://xbrlsite.azurewebsites.net/2018/Library/ReportingSchemes-2018-12-30.pdf
[95] *XBRL-based Digital Financial Reporting Profiles and General Business Reporting Profile*, http://xbrlsite.azurewebsites.net/2018/Library/Profiles-2018-10-22.pdf

2.5. Knowledge Acquisition Mechanism

The power of any knowledge based system is proportional to the key ingredient of the knowledge based system which is high-quality machine-readable domain knowledge available to that system. Knowledge acquisition is the process of obtaining that domain knowledge.

There are three approaches to acquiring knowledge:

1. A rules-based approach which involves humans creating machine-readable knowledge.
2. A patterns-based approach which involves machine learning to capture domain knowledge which is useful when there is a high tolerance for error. Further, extensive machine-readable training data is necessary to use this machine-learning based approach.
3. A combination of approaches #1 and #2 to create a hybrid approach to acquiring knowledge.

For the domain of financial reporting, there is ZERO probability that approach #2 (i.e. machine learning) can be used to acquire the initial financial reporting domain knowledge.

However, after some unknown period of time when enough machine-readable information has been created by human domain experts; then that human created machine-readable information can be leveraged to create additional new information.

For example, information about disclosures[96] can be used to learn how to create algorithms for identifying other such disclosures simply by probing existing XBRL-based financial reports submitted to financial regulators such as the SEC and ESMA. That machine-readable information along with humans to guide and tweak the process can be used to identify rules for other unknown disclosures by looking for specific known patterns.

We don't want every enterprise or regulator creating proprietary approaches to creating knowledge based systems for storing and working with financial reports. A better approach for everyone is to have high-quality global standard models which makes creating software more efficient and therefore less costly.

It takes skill and experience of a domain to create knowledge for a domain. Business professionals have that skill and experience and will need software which they can realistically use to put collate, categorize, associate, and otherwise create useful machine-readable knowledge.

[96] Disclosure Best Practices, http://xbrlsite-app.azurewebsites.net/DisclosureBestPractices/DisclosureBestPractices.aspx?DisclosureName=IncomeStatement

3. Learning about Digital Financial Reporting from CAD/CAM

Contrasting something new that does not yet exist to something similar that does exist is one way of understanding something[97]. Digital financial reporting has the opportunity to do for the financial report and the financial reporting supply chain what CAD/CAM and BIM[98] did for not only the blueprint, but for the entire product design and manufacturing life cycle.

3.1. Digital Blueprint

Computer-aided design[99] (CAD) is the use of computer systems to aid in the creation, modification, analysis, or optimization of a design. CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing. CAD output is often in the form of electronic files for print, machining, or other manufacturing operations. Computer-aided manufacturing[100] (CAM) is the use of software to control machine tools such as numerically controlled machines (NC).

In CAD/CAM software architectural objects have relationships to one another and interact with each other intelligently. For example, a window has a relationship to the wall that contains it. If you move or delete the wall, the window reacts accordingly.

In addition, in CAD/CAM software machine-readable architectural objects maintain dynamic links with construction documents and specifications, resulting in more accurate project deliverables. When someone deletes or modifies a door, the door schedule is automatically updated in your local application's database and perhaps even in the database of the door supplier. Spaces and areas are update automatically when the size of a room is changed and calculations such as total square footage are always up to date. That means, say, that the amount of paint necessary to cover a room or an entire building is always updated. Blueprints can be sent directly to numerically controlled[101] (NC) machines.

Imagine what it would be like to construct a 100 story sky scraper, an iPhone, or a Boeing 777 if all the blueprints were paper-based. Turning this around, digital blueprints enable process and other improvements which allow more sophisticated products to be created effectively and efficiently.

3.2. CAD/CAM is an Expert System

CAD/CAM software is an expert system that understands architectural design and engineering objects. CAD/CAM systems understand what things like buildings, walls, doors are and the relations between those things. CAD/CAM software will not let you put, say, a door in a roof or otherwise construct nonsensical objects.

---

[97] This blog post provides three videos which show how CAD works, helps you to imagine how digital financial reporting will work; *Intelligent XBRL-based Digital Financial Reports*, http://xbrl.squarespace.com/journal/2017/1/1/intelligent-xbrl-based-digital-financial-reports.html

[98] Vimeo, Jason Pratt, *The Difference Between CAD and BIM*, https://vimeo.com/4415128

[99] *Computer-aided Design*, https://en.wikipedia.org/wiki/Computer-aided_design

[100] *Computer-aided Manufacturing*, https://en.wikipedia.org/wiki/Computer-aided_manufacturing

[101] *Numerical control*, https://en.wikipedia.org/wiki/Numerical_control

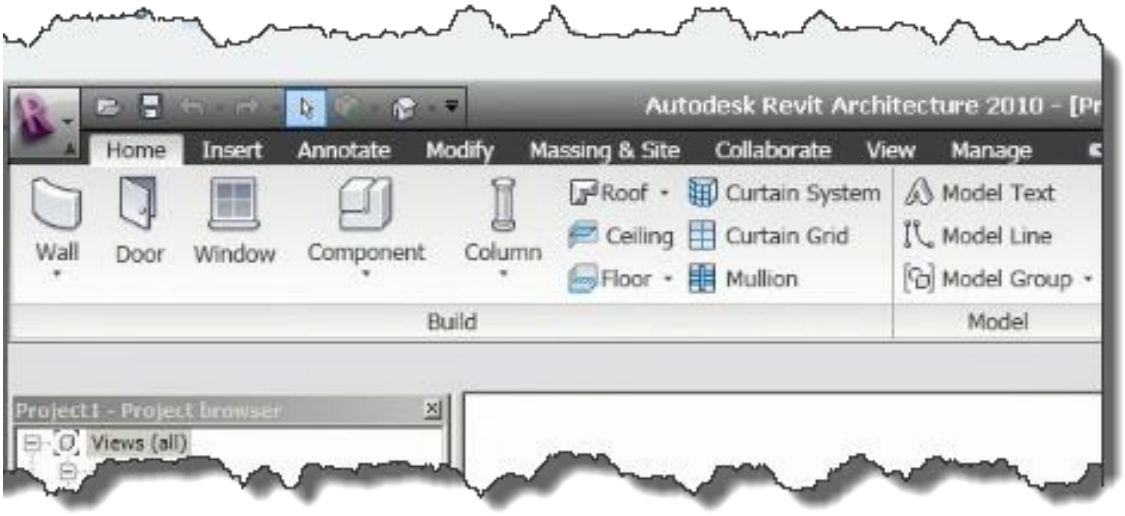
Systems such as CAD/CAM work by enabling software systems to work with objects and relations between different objects using machine readable rules or logic. Nontechnical professionals work with high-level objects that they understand rather than low-level technical artifacts that they do not[102]. Machine-readable metadata is leveraged which supercharges software applications performing work[103].

3.3. BIM

CAD/CAM was popularized in the 1980s. CAD basically lets you work with geometric shapes such as lines, squares, triangles, and other drawing tools to electronically create drawings which could then be printed out. Essentially, CAD was a lot like creating an "e-blueprint".

BIM is significantly different than traditional CAD. BIM, or Building Information Modeling. BIM is a disruptive technology[104]. CAD was a drawing tool, BIM is a modeling tool. BIM models the systems of a building, structure, product, etc. BIM is about information. With BIM you work with objects such as windows, doors, roofs, walls, etc. This is by contrast to CAD where you work with lines, squares, and triangles.

If you watch the video, *The Difference between CAD and BIM*[105], you can better understand the advantages of BIM. In the video, for the first 3 minutes and 52 seconds the narrator shows how you work with lines, arcs, and other shapes to draw. Then after that, he shows how he works with walls, doors, windows, roofs, ceilings, flours and other such objects to work in a BIM tool. Views of the building are created automatically as the model is configured.



[102] Charles Hoffman, *Leveraging Functional Components for XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2019/Library/LeveragingFunctionalComponents.pdf

[103] Charles Hoffman, *Curated Machine-Readable Information (also Human-Readable) is the Future*, http://xbrl.squarespace.com/journal/2019/6/14/curated-machine-readable-information-also-human-readable-is.html

[104] Luckett & Farley, *The Evolution of Drafting*, https://www.youtube.com/watch?v=_ULPY3B2BoQ&feature=youtu.be

[105] Jason Pratt, *The Difference between CAD and BIM*, https://vimeo.com/4415128

3.4. Digital Financial Report Creation will be like BIM, not CAD

Today, most software used for creating XBRL-based financial reports is similar to CAD. That was a mistake made by software creators. The next evolution of creating XBRL-based financial reports will be like using BIM, not like using CAD. The technical syntax of what you are doing will be completely invisible to the business user of the software, pushed into the background the XBRL technical syntax will (a) always be right and (b) never be exposed to the software user.

As explained in the *Logical Theory Describing a Business Report*[106] and the *Financial Report Semantics and Dynamics Theory*[107]; the models of a business report and a financial report can be defined. Patterns can be identified and leveraged.

As is explained in *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*[108] and *Guide to Building an Expert System for Creating Financial Reports*[109], these ideas are already tested and proven to work effectively.

The document *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*[110] explains the logical system of a financial report. There are six reporting schemes that provide metadata and test these ideas[111]. A working proof of concept has been created, Pesseract[112], that uses the metadata and the logical model and tests this method, framework, principles, etc.

---

[106] Charles Hoffman, CPA and Rene van Egmond, *Logical Theory Describing a Business Report*, http://xbrlsite.azurewebsites.net/2019/Library/LogicalTheoryDescribingBusinessReport.pdf

[107] Charles Hoffman, CPA and Rene van Egmond, *Financial Report Semantics and Dynamics Theory*, http://xbrlsite.azurewebsites.net/2016/Library/Theory-2017-06-26.pdf

[108] Charles Hoffman, CPA and Hamed Mousavi, *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*, http://pesseract.azurewebsites.net/PuttingTheExpertiseIntoKnowledgeBasedSystem.pdf

[109] Charles Hoffman, CPA, *Guide to Building an Expert System for Creating Financial Reports*, http://xbrlsite.azurewebsites.net/2018/Library/GuideToBuildingAnExpertSystemForCreatingFinancialReports.pdf

[110] *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*, http://xbrl.squarespace.com/journal/2019/12/30/special-theory-of-machine-based-automated-communication-of-s.html

[111] *Modern Approach to Creating a Financial Reporting Scheme*, http://xbrl.squarespace.com/journal/2019/12/19/modern-approach-to-creating-a-financial-reporting-scheme.html

[112] *Pesseract*, http://pesseract.azurewebsites.net