Expert System for Creating Financial Reports Explained in Simple Terms

How an expert system for creating financial reports can be created, explained in simple terms understandable to both accountants and software engineers with the purpose of enabling communications to actually create such a cloud-based software application.

By Charles Hoffman, CPA (Charles.Hoffman@me.com)

March 18, 2022 (DRAFT)

http://xbrl.squarespace.com/

"Great things are done by a series of small things brought together." *Vincent Van Gogh*

Executive summary:

- This document describes how a rules-based expert system which assists in the construction of a financial report would work in terms understandable to accountants and software engineers.
- Created financial reports may, or may not, be output in the XBRL technical format. Reports would just as well be published in PDF, HTML, Microsoft Word, Open Doc, Adobe InDesign, or other human readable presentation format.
- The purpose of this rules-based expert system is to use more modern approaches to creating such reports and making the process better, faster, and cheaper than current approaches to creating financial reports which are basically outdated.
- As such, this application contemplates a paradigm shift; a new modern approach to financial reporting not an incremental innovation of the current approach.

As explained in the book *The Great Upheaval*[1], the world is in the midst of a "great upheaval" where the world is transitioning from an analog, industrial economy to a digital, knowledge economy. To understand the notion of a rules-based expert system effectively the reader must understand that an analog, industrial economy works differently than a digital, knowledge economy.

What makes "digital, knowledge economy" work in the context required by financial reporting has been explained by me in other documents such as *Computational Professional Services*[2], *Essence of Accounting*[3], *Financial Report Knowledge Graphs*[4], and the *Seattle Method*[5]. That "digital" accounting, reporting, auditing, and analysis can work has been proven and I will not repeat that proof here.

I have the tendency to create over documented explanations in order to help the reader have access to all the information that I have access to in order to follow the logic of my explanations and reach their own conclusions. In this document I will try and keep explanations as terse and easy to read as possible, linking to additional explanations where a reader can go for details if they desire to do so.

The objective of this paper is to provide as non-technical as possible explanation of the essence of what a rules-based expert system does for those creating such financial reports. This will be explained in terms that an accountant can understand, in terms that a software engineer can understand, and such that accountants and software engineers working together to create such an application can communicate effectively.

This document articulates one way of creating an expert system for constructing financial reports. There are other approaches.

Vision for Expert System for Creating Financial Reports

I am more convinced than ever that some sort of graph database is the way to go to store information for a report within some sort of database. I now understand the rigidity of a relational database, the increased flexibility of a graph database, the power of a graph database

[1] Author Levine and Scott J. van Pelt, *The Great Upheaval*, https://www.amazon.com/Great-Upheaval-Educations-Present-Uncertain/dp/1421442574
[2] Charles Hoffman, CPA, *Computational Professional Services*, http://xbrlsite.azurewebsites.net/2020/library/ComputationalProfessionalServices.pdf
[3] Charles Hoffman, CPA, *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf
[4] Charles Hoffman, CPA, *Financial Report Knowledge Graphs*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf
[5] Charles Hoffman, CPA, *Seattle Method*, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf

to represent relations, and the power of a graph database to express a logical schema, and the ease with which a logic/rules/knowledge/insights engine can "interact" with that graph. You do need some sort of logic engine to process the logic; which TerminusDB has (DATALOG). Other graph databases do NOT have that sort of engine.

PROLOG and RDF+OWL+SHACL (the Semantic Web Stack) can likewise do just as good a job as a graph database and they have logic engines and/or semantic reasoners to interact with the graph. You COULD do this same thing with a relational database, but you would have to build many things that you get for free with a graph database and some sort of logic/rules/reasoning/knowledge/insights engine to interact with that knowledge graph.

But I am wondering if creating your OWN logic engine (which Pesseract did) is a better way to go or if it would be reinventing a wheel that already exists. Building some sort of logic processor is a non-trivial task. But doing so could have advantages because you could specifically TUNE the logic engine to be exactly what you need it to do.

An observation with respect to what Auditchain did with Pacioli is that Pacioli is a batch oriented processor currently. What you need is a more "dynamic" processor, more like a video game during the report creation process. Allowing users to do the wrong things, then discovering them at the END of the process, and then letting the user then FIX the issue is not the way to go.

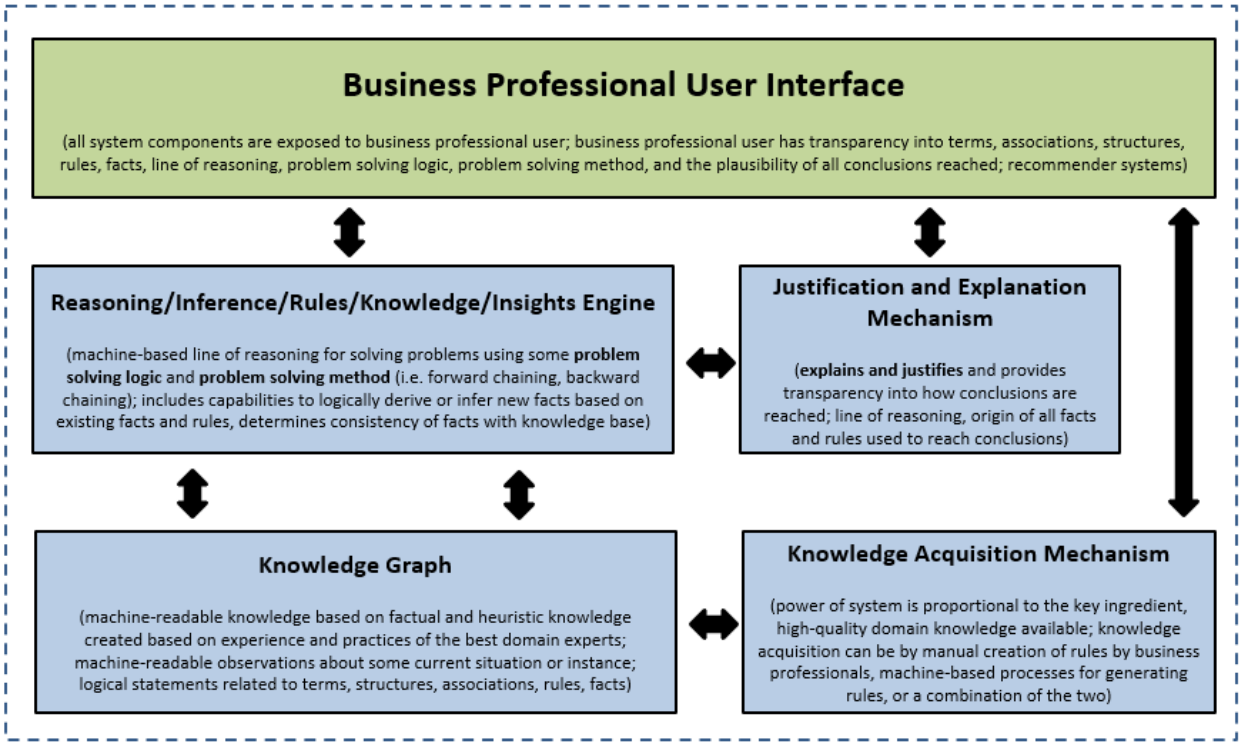
The knowledge graph includes all the core pieces (logical statements) of a report (terms, associations, structures, rules, facts). In addition to this, additional pieces (logical statements) are added such as the model structure rules, the reporting checklist, the disclosure mechanics, the type-subtype (wider-narrower or general-special) associations, the fundamental accounting concept relations, disclosures, topics, templates, exemplars. All of these pieces (logical statements) provide metadata that can be leveraged to enable a logic/rules/reasoning/knowledge/insights engine to assist the human user of the software application. Add to that a state machine that provides additional information and you have a massive amount of useful information that can be used to drive recommender systems, project management systems, construction systems, and other such ways the software can augment the business user's skills.

This document assists in the process of creating a blueprint to make such a rules-based expert system to be created.

Components of a Knowledge Based System

I have provided versions of a graphic which you see the most current version below for several years which summarizes the "Components of a Knowledge Based System" which is as some people call it "a good ole fashioned rules-based expert system".

That graphic is the basis for Pesseract[6] which is a working proof of concept which a software engineer and I created to figure out how to construct an expert system for creating financial reports. Fundamentally, that graphic shows all the pieces you need to create such an expert system. Pesseract has within it what is essentially a special purpose problem solving logic engine.

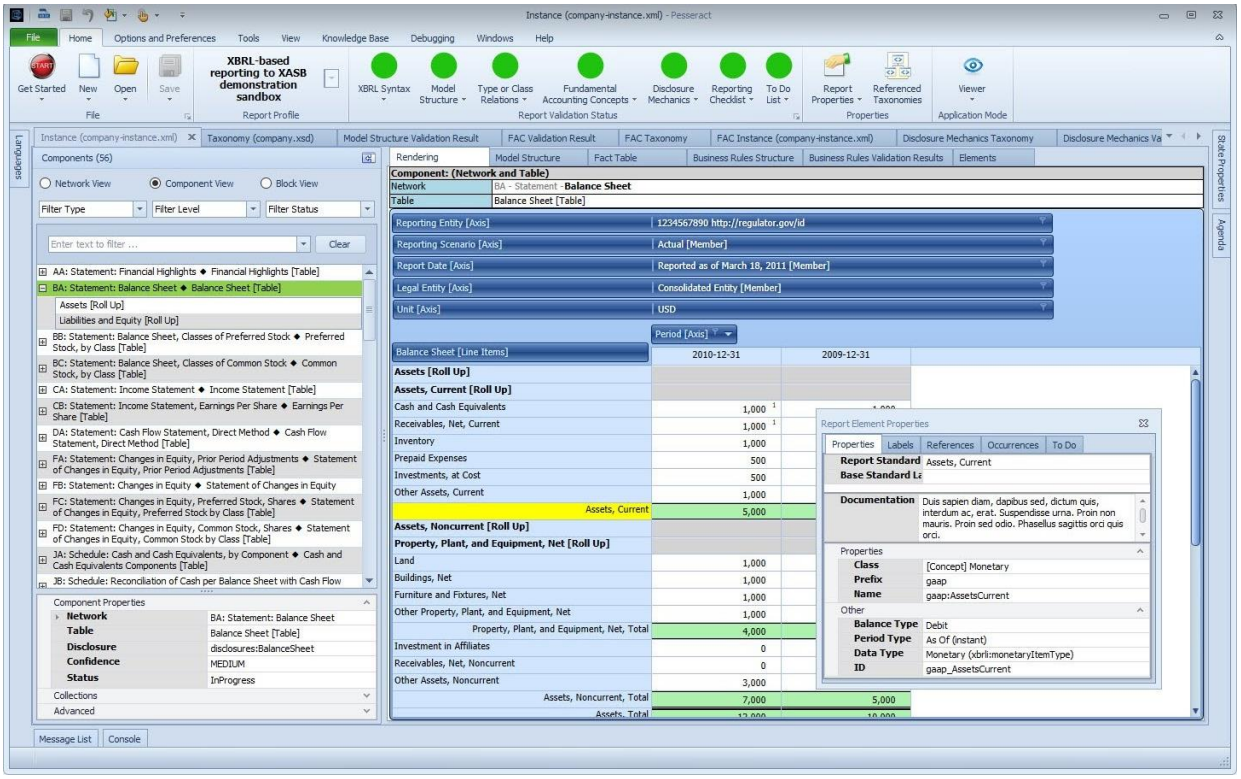


Business Professional User Interface

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 business professional user interface will very likely be a cloud-based interface.

The following is one of a number of screen shots[7] of the working proof of concept software application Pesseract which provides an example of a user interface with which a business professional would interact:

[6] Pesseract, http://pesseract.azurewebsites.net/
[7] Additional Pesseract User Interface Screenshots, https://photos.app.goo.gl/cWeZYaMBEbmSSm7v8



A financial report is a knowledge graph[8]. The user interface is non-technical requiring only business and accounting knowledge to effectively understand the software application and how to use it. The interface is a specialized logical interface that leverages the *Logical Theory Describing Financial Report*[9]. Because the interface is specialized for one specific area of knowledge instead of a general technical interface that can be used for any area of knowledge, it can be easy to use.

Justification and Explanation Mechanism

The justification and explanation mechanisms of the software application explains and justifies and provides transparency into how conclusions have been reached by the software application and communicates this information to the software user. 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.

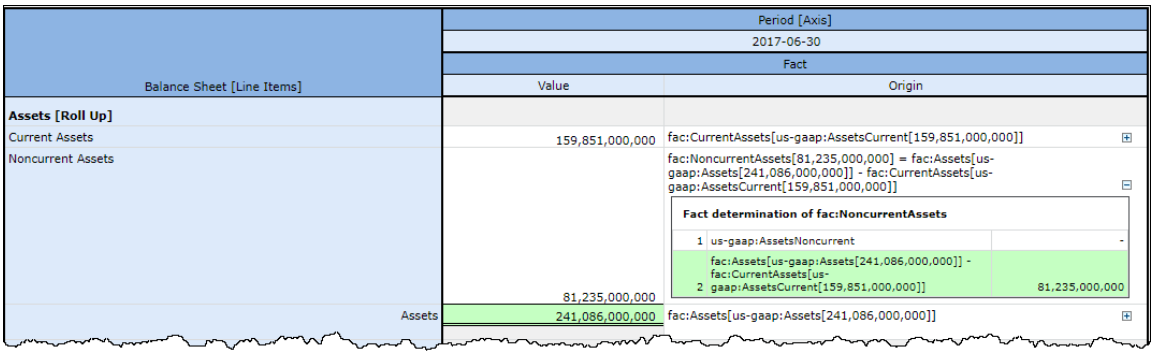
Below you see the fundamental accounting concept relations continuity cross check verification checks provided by XBRL Cloud's Evidence Package[10] which is a review tool that can be used to verify XBRL-based financial reports:

---

[8] *Financial Report Knowledge Graphs*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf

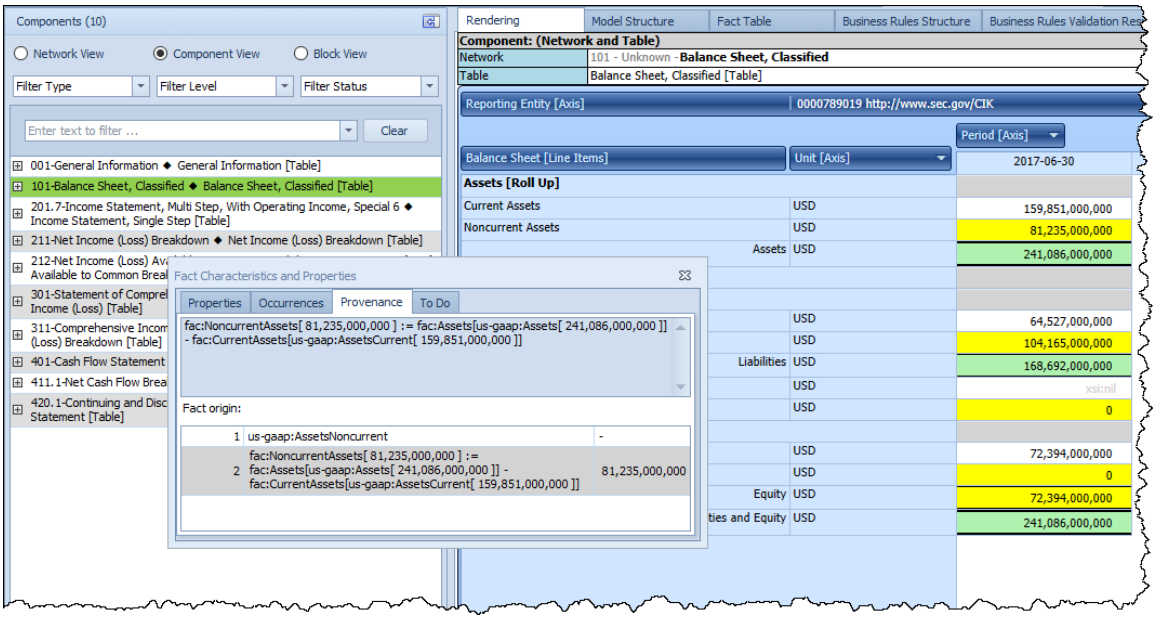
[9] *Logical Theory Describing Financial Report*, http://xbrlsite.com/seattlemethod/LogicalTheoryDescribingFinancialReport_Terse.pdf

[10] 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.

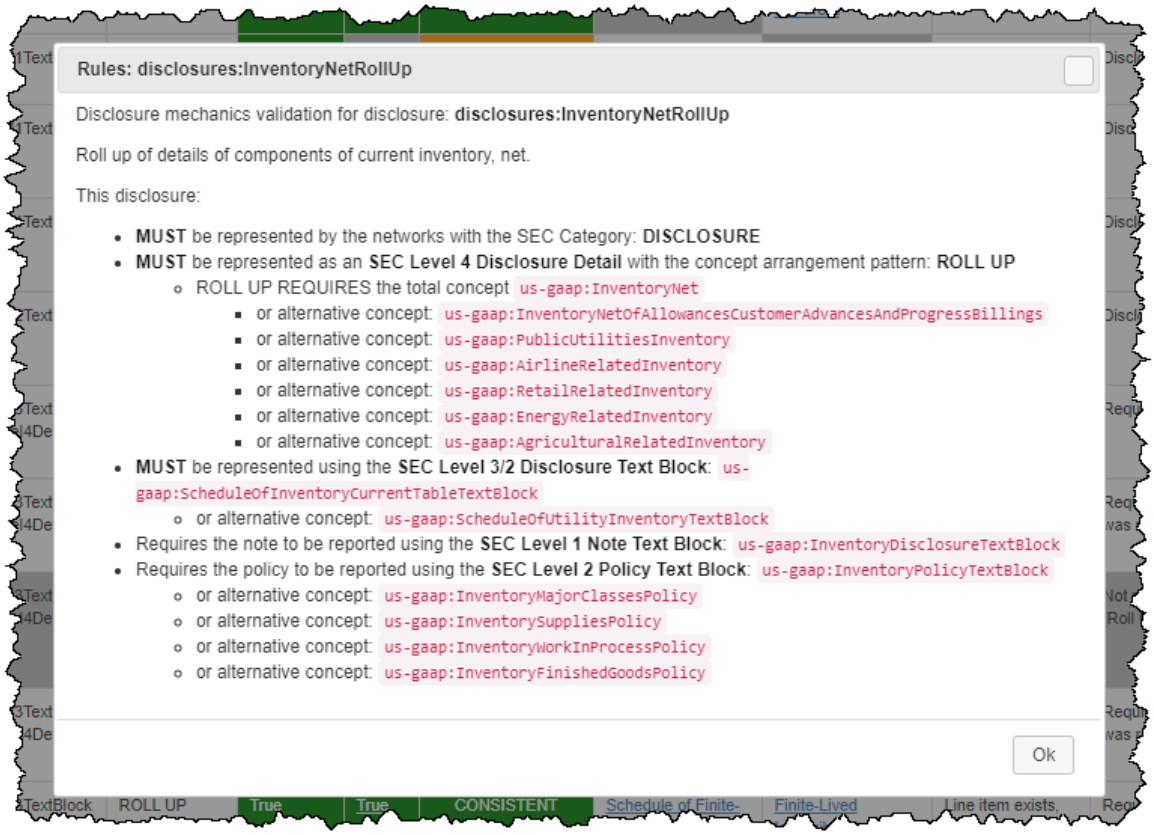
Pesseract provides similar functionality:



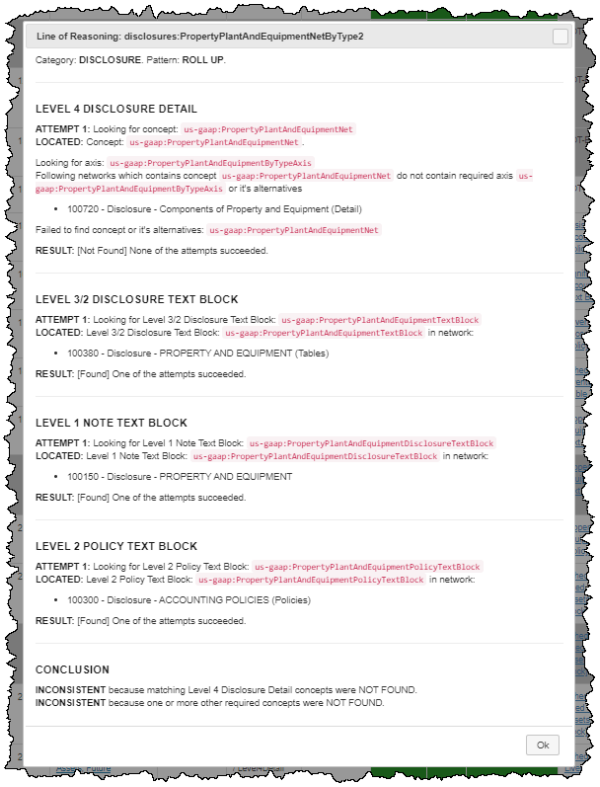
XBRL Cloud's Disclosure Mechanics and Reporting Checklist[11] provides the rules used, line of reasoning used, and conclusions reached for determining if a disclosure is structured consistent with its expected specification:

Disclosure mechanics rules:

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



Line of reasoning:

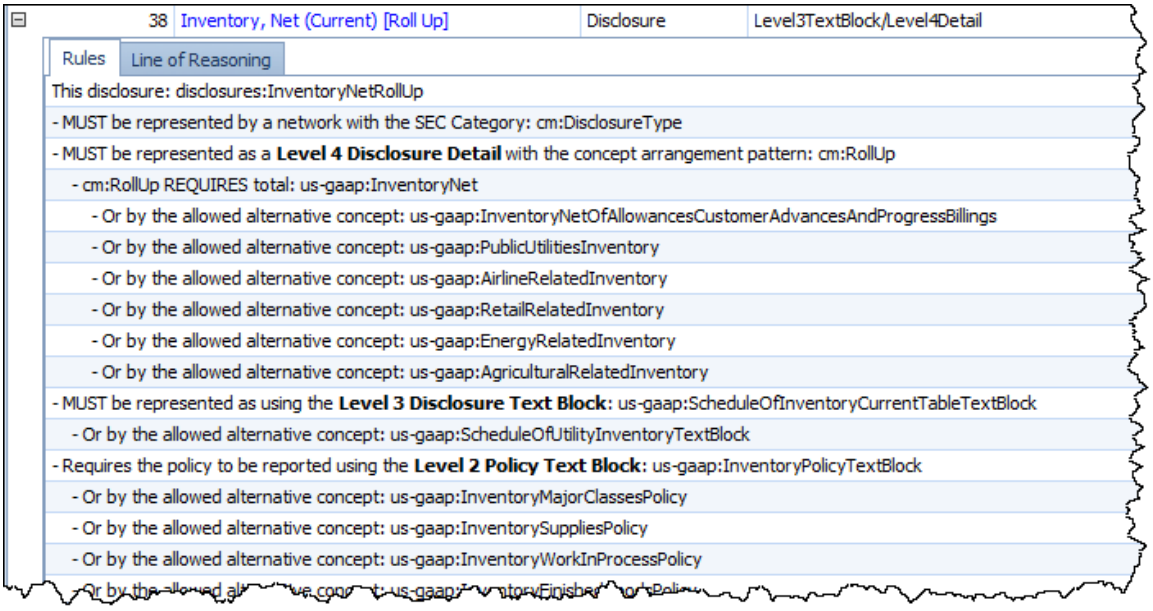


Conclusion reached:

#	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
	Statement of Income and	DISCLOSURE	Level4Detail	ROLL UP	False	True	CONSISTENT	NOT-EXPECTED	Net Income (Loss) Attributable to Parent	Alternative representation	Not necessary, satisfied by Statement of Comprehensive Income [Roll

Similar functionality is offered by Pesseract:

Disclosure mechanics rules:



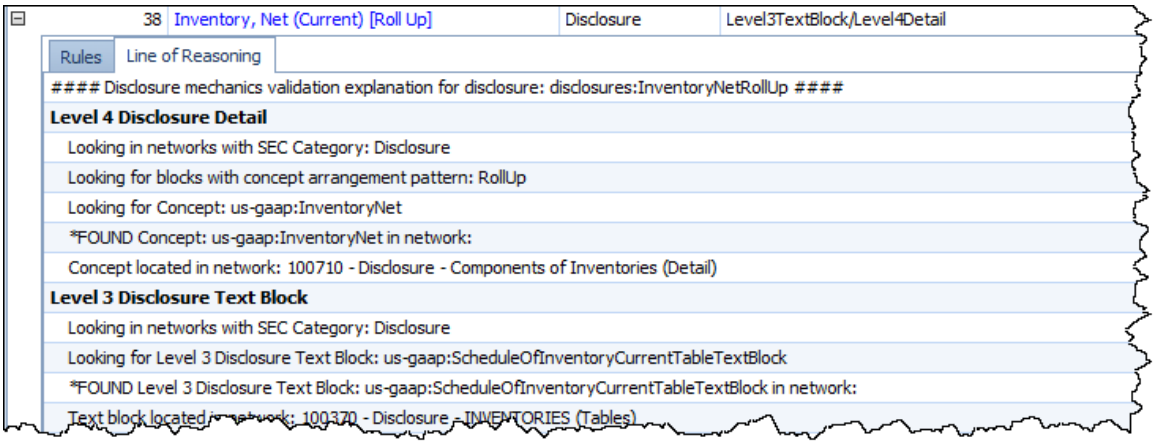
38	Inventory, Net (Current) [Roll Up]	Disclosure	Level3TextBlock/Level4Detail

Rules | Line of Reasoning

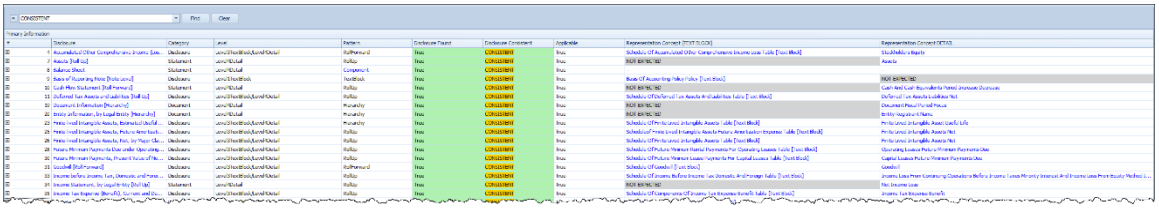
This disclosure: disclosures:InventoryNetRollUp

- MUST be represented by a network with the SEC Category: cm:DisclosureType
- MUST be represented as a **Level 4 Disclosure Detail** with the concept arrangement pattern: cm:RollUp
  - cm:RollUp REQUIRES total: us-gaap:InventoryNet
    - Or by the allowed alternative concept: us-gaap:InventoryNetOfAllowancesCustomerAdvancesAndProgressBillings
    - Or by the allowed alternative concept: us-gaap:PublicUtilitiesInventory
    - Or by the allowed alternative concept: us-gaap:AirlineRelatedInventory
    - Or by the allowed alternative concept: us-gaap:RetailRelatedInventory
    - Or by the allowed alternative concept: us-gaap:EnergyRelatedInventory
    - Or by the allowed alternative concept: us-gaap:AgriculturalRelatedInventory
- MUST be represented as using the **Level 3 Disclosure Text Block**: us-gaap:ScheduleOfInventoryCurrentTableTextBlock
  - Or by the allowed alternative concept: us-gaap:ScheduleOfUtilityInventoryTextBlock
- Requires the policy to be reported using the **Level 2 Policy Text Block**: us-gaap:InventoryPolicyTextBlock
  - Or by the allowed alternative concept: us-gaap:InventoryMajorClassesPolicy
  - Or by the allowed alternative concept: us-gaap:InventorySuppliesPolicy
  - Or by the allowed alternative concept: us-gaap:InventoryWorkInProcessPolicy
  - Or by the allowed alternative concept: us-gaap:InventoryFinishedGoodsPolicy

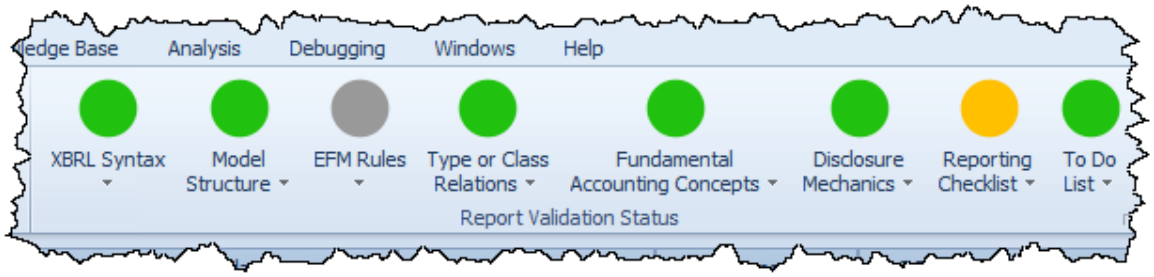
Line of reasoning:



Conclusion reached[12]:



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)[13]. The following interface provides a summary of verification results:



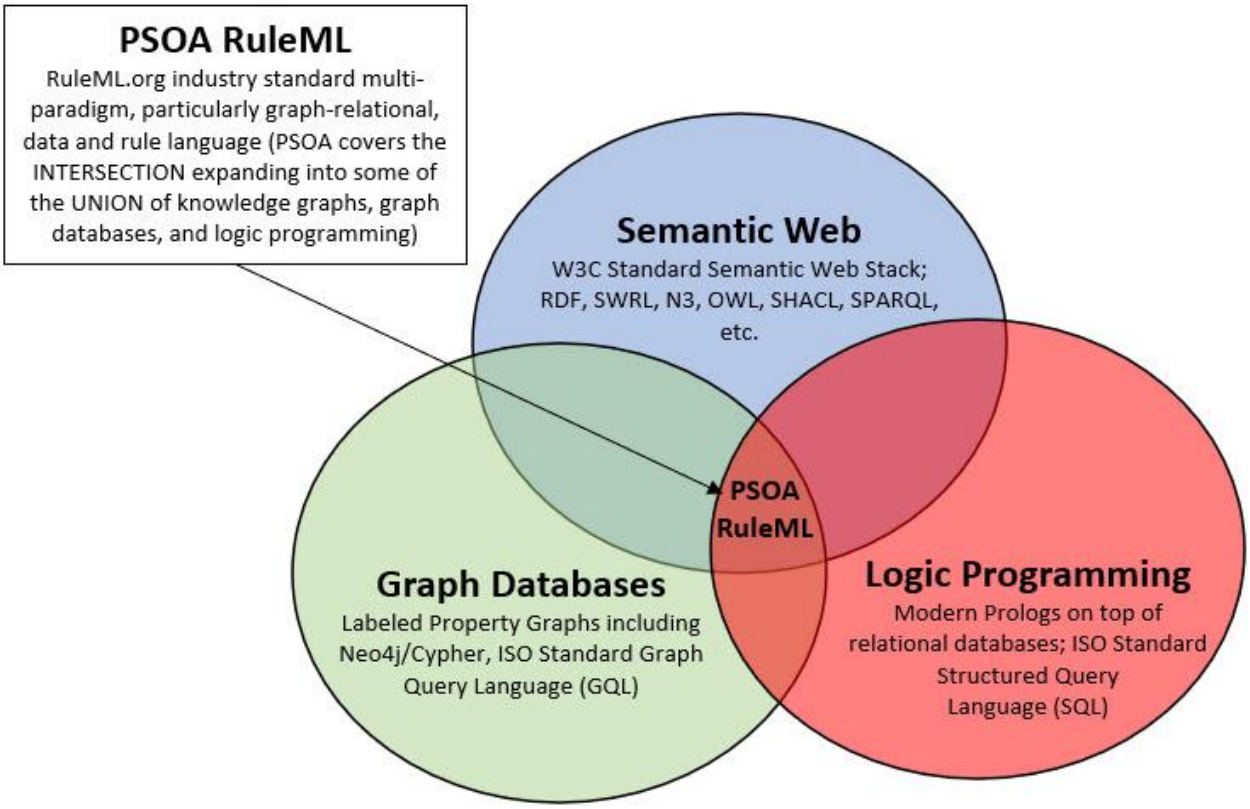
Reasoning, Inference, Knowledge, Insights, Rules Engine

The reasoning, inference, rules, knowledge, and insights engine (not exactly sure what to call it) 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.

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

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

As pointed out by the RuleML industry consortium[14], there are three primary problem solving logic paradigms[15]: Semantic Web approach, Graph Database approach, and Logic Programming approach. There is an intersection between these three paradigms represented by PSOA RuleML[16] where one paradigm can be translated logically to each of the other paradigms.



DATALOG[17] is the "sweet spot" in terms of balancing processing power, processing safety to avoid logical paradoxes which cause systems to completely fail, and transportability between different technical syntax alternatives. For example, pyDatalog[18]. This video, *Datalog in Practice*[19], helps you understand Datalog.

Knowledge Graph

Information that the rules/logic/reasoning/knowledge/insights engine interacts with is stored in a machine-readable knowledge graph. Some people are of the opinion that the "facts" (fact database) are separate from the knowledge graph. In that case, then consider this the "Knowledge Graph" plus the

[14] RuleML.org, *Graph-Relational Data, Ontologies, and Rules*, http://wiki.ruleml.org/index.php/Graph-Relational_Data,_Ontologies,_and_Rules
[15] Implementing Knowledge Graphs, http://xbrl.squarespace.com/journal/2021/9/20/implementing-knowledge-graphs.html
[16] RuleML.org, PSOA RuleML, http://wiki.ruleml.org/index.php/PSOA_RuleML
[17] Wikipedia, *Datalog*, https://en.wikipedia.org/wiki/Datalog
[18] pypi.org, pyDatalog, https://pypi.org/project/pyDatalog/
[19] YouTube, Datalog in Practice, https://youtu.be/RQVZ0tBj2F4

"Fact Database". Combining or separating those two pieces is an implementation detail which really does not matter; effectively a machine can interact with all of the information that it needs to interact with. The knowledge graph is constrained using a physical schema[20] which constrains the technical syntax schemas for the physical syntax used and logical schemas[21] to constrain the logic that is stored withing that physical schema. The Seattle Method[22] outlines the financial report levels[23] that specifies the necessary logic which must be constrained to keep the system quality high.

Burying Complexity

A *kluge* is a term from the engineering and computer science world that refers to something that is convoluted and messy but gets the job done. Elegance is the quality of being pleasingly ingenious, simple, neat. Elegance is about beating down complexity.

Complexity can never be removed from a system, but complexity can be moved. The *Law of Conservation of Complexity* states: "Every application has an inherent amount of irreducible complexity. The question is who will have to deal with that complexity: the application user, the application developer, or the platform developer that the application runs on?"

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.

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

Simplistic and simple are not the same thing.

Simplistic is dumbing down a problem in order to make the problem easier to solve. Simplistic ignores complexity in order to solve a problem which can get you into trouble. Simplistic is over-simplifying. Simplistic means that you have a naïve understanding of the area of knowledge, you don't understand the complexities of the area of knowledge. Removing or forgetting complicated things does not allow for the creation of a real-world solution that will actually work for an area of knowledge.

Simple is something that is not complicated, that is easy to understand or do. Simple means without complications. An explanation of something can be consistent with the area of knowledge in the real world, consider all important subtleties and nuances, and still be simple, straight forward, and therefore easy to understand and use.

---

[20] Wikipedia, *Physical Schema*, https://en.wikipedia.org/wiki/Physical_schema
[21] Wikipedia, *Logical Schema*, https://en.wikipedia.org/wiki/Logical_schema
[22] Charles Hoffman, CPA, *Seattle Method*, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf
[23] Auditchain, *Financial Report Levels*, http://accounting.auditchain.finance/library/FinancialReportLevels.pdf

Hard problems call for great efforts. Great effort was undertaken to bury complexity, achieve elegance, such that the millions of professional accountants that will use this system will benefit from that effort.

Exchange of Complex Information

A key to employing "digital" in accounting, reporting, auditing, and analysis in the enterprise is that information exchange needs to work effectively, predictably, reliably, safely, and correctly. XBRL is the global standard for the exchange of business information. The *Seattle Method* leverages and enhances that global standard.

The *Seattle Method* has worked to create a proven, good practices, standards-based pragmatic approach to creating provably high quality XBRL-based general purpose financial reports where report creators are permitted to modify the report model.

Auditchain[24] enhances the XBRL standard by adding the features of trust, provenance, coordination, and an economy.

The objective is seen not as some regulatory mandate but rather an opportunity to understand and leverage the power of digital. When complex information, like financial information, can be effectively exchanged then opportunities open up that make seemingly magical things possible. (But don't tell anyone that it really is not magic, it was really just rolling up our sleeves and doing the hard work to make this actually work.)

Area of Knowledge

An area of knowledge is a highly organized socially constructed aggregation of shared knowledge for a distinct subject matter. An area of knowledge has a specialized insider vocabulary, jargon, underlying assumptions (axioms, theorems, constraints), and persistent open questions that have not necessarily been resolved (i.e. flexibility is necessary).

Accounting is an area of knowledge. You can explain aspects of the accounting area of knowledge, such as the nature of a financial report, using a logical theory which explains a logical model. A logical theory can be tested and proven by providing a proof.

Knowledge can be represented in human-readable form, in machine-readable form, or in a machine-readable form that can be effectively converted into human-readable form. Other terms for area of knowledge are a knowledge domain or simply domain or universe of discourse.

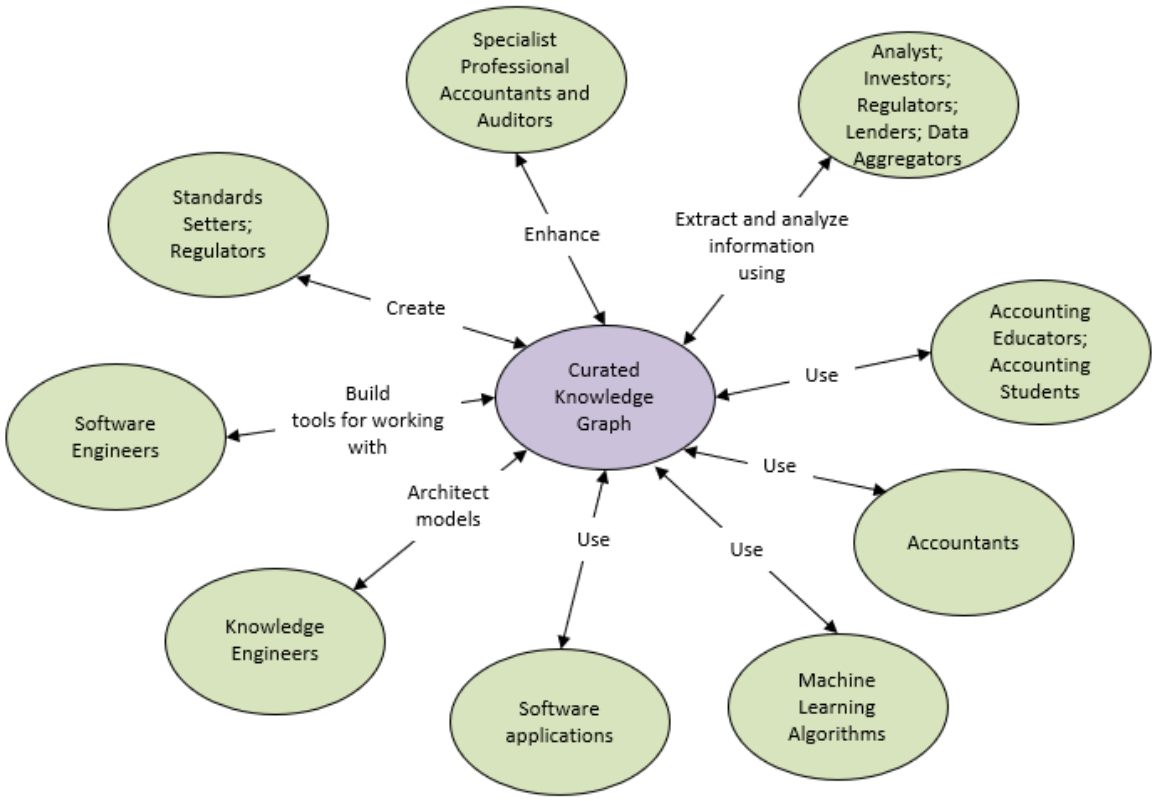
Information Supply Chain

Accounting, reporting, auditing, and analysis can be thought of as a supply chain; an information supply chain. Jane Gleeson-White titled Chapter 1 of her book *Double Entry: How the Merchants of Venice*

[24] *Auditchain Explained in Simple Terms*, http://xbrlsite.azurewebsites.net/2022/library/AuditchainExplained.pdf

Created Modern Finance[25], Accounting: Our First Communications Technology. Double entry bookkeeping is a mathematical model[26] created over 800 years ago but perfectly suited for the information age.

The members of this information supply chain can interact more effectively and efficiently using high quality curated machine-readable metadata represented in a global standard technical format.

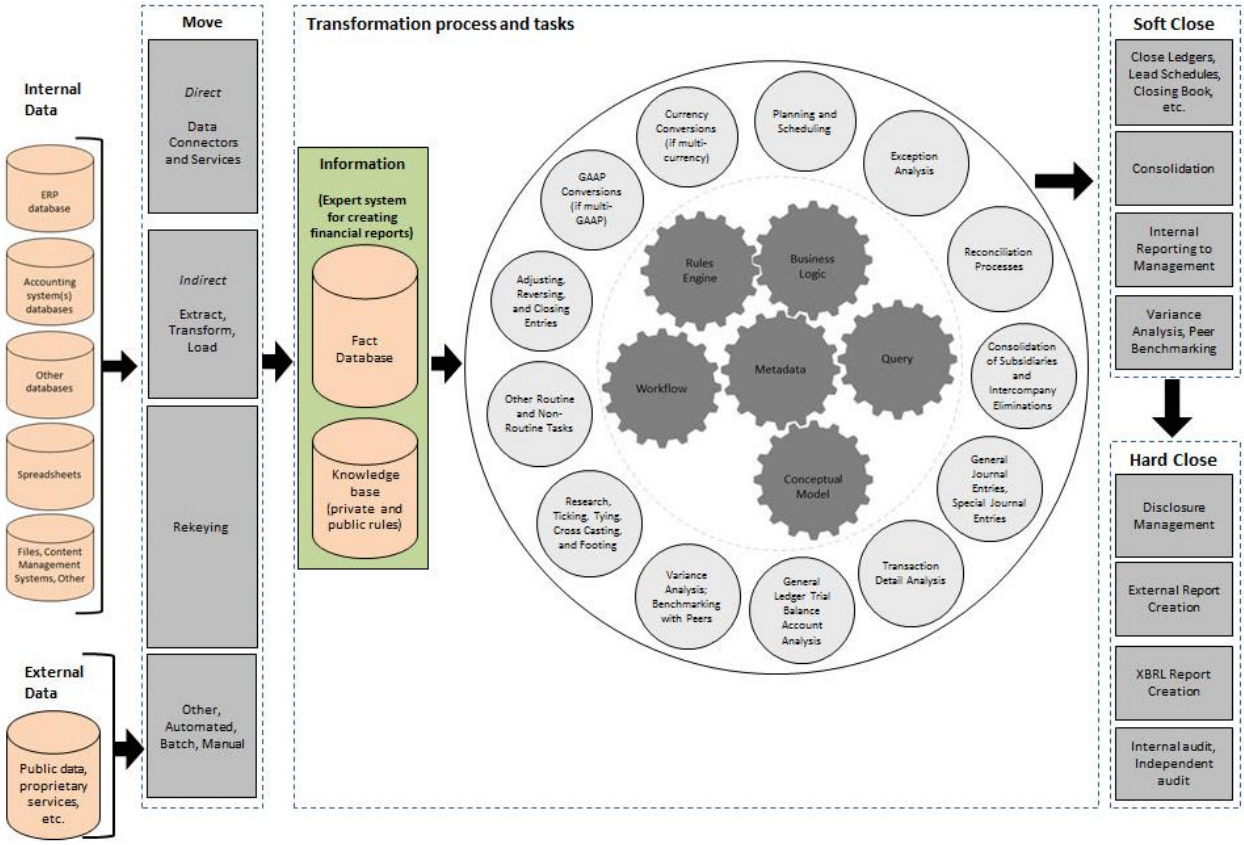


XBRL based global standard information, high-quality curated knowledge, and a logic/reasoning/rules/ knowledge/insights engine with proper process control can significantly reduce system friction.

[25] Jane Gleeson-White, *Double Entry: How the Merchants of Venice Created Modern Finance*, https://www.amazon.com/gp/product/B007Q6XKA8/

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

The process of creating financial reports, auditing those reports, and analyzing report information can be completely rethought.



Such a system removes the repetitive, mechanical, mundane, even gruesome and grueling tasks and processes from the responsibility of humans freeing up accountants, auditors, and analysts to do what they do best: judgement, analysis, creativity, non-routine tasks, unstructured tasks, politics, compassion, innovation, improvising.

Lean Six Sigma

Processes and tasks are controlled and quality is kept very high by using Lean Six Sigma principles, philosophies, and techniques. Lean Six Sigma[27] is a discipline that combines the problem-solving methodologies and quality enhancement techniques of Six Sigma[28] with the process improvement tools and efficiency concepts of Lean Manufacturing[29]. Born in the manufacturing sector, Lean Six Sigma works to produce products and services in a way that meets consumer demand without creating wasted time, money and resources.

Specifically, Lean[30] is 'the purposeful elimination of wasteful activities.' It focuses on making process throughout your company faster, which effects production over a period of time. Six Sigma[31] works to develop a measurable process that is nearly flawless in terms of defects, while improving quality and removing as much variation as possible from the system. For additional details, please refer to Lean Six Sigma[32].

PROOF

To demonstrate how to create an expert system for creating financial reports in this document I will primarily be using the PROOF[33] example of an XBRL-based financial report. Occasionally I might use other examples when they provide significantly enhanced information, but for continuity I will try to stick to the PROOF example as much as I can.

Pacioli Verification of Proof

The following is the verification script that will run the complete verification set provided by the *Seattle Method* against the PROOF report using the Pacioli Power User Tool:

https://pacioli.auditchain.finance/tools/PowerUserTool.swinb

Verification Script to Run: (paste this into the tool)

---

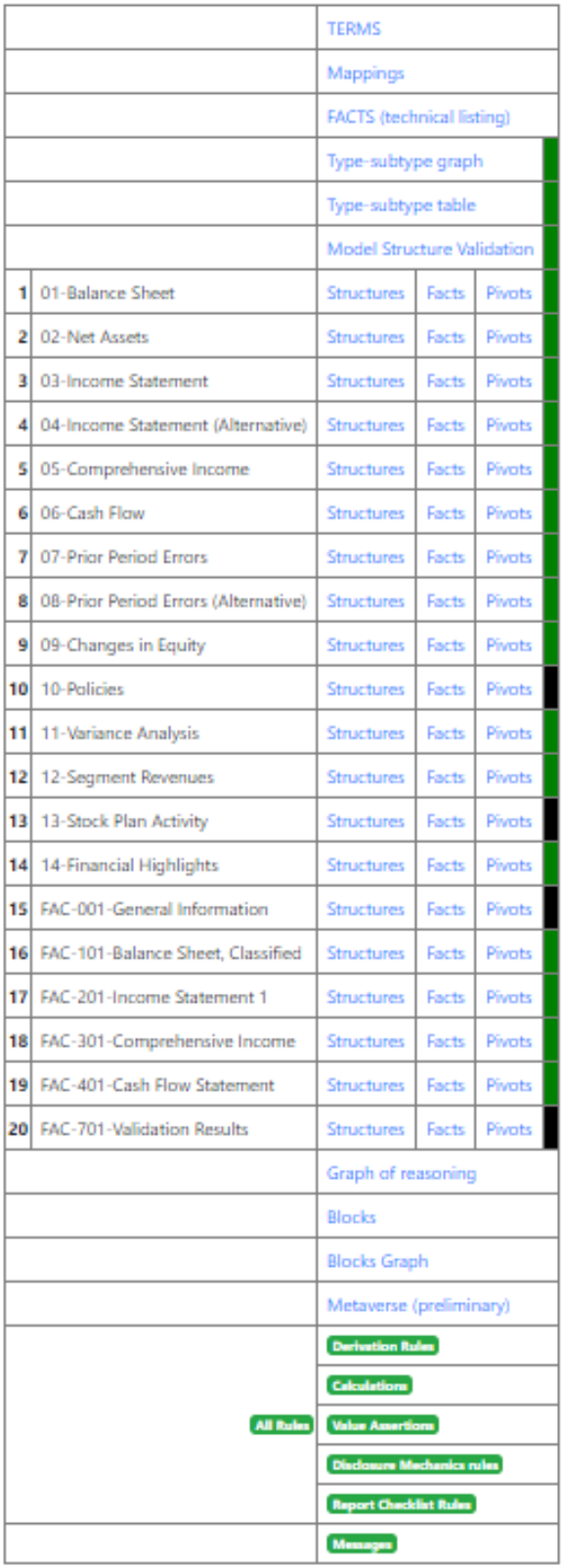
[27] Wikipedia, *Lean Six Sigma*, https://en.wikipedia.org/wiki/Lean_Six_Sigma
[28] Wikipedia, *Six Sigma*, https://en.wikipedia.org/wiki/Six_Sigma
[29] Wikipedia, *Lean Manufacturing*, https://en.wikipedia.org/wiki/Lean_manufacturing
[30] YouTube.com, *Lean Six Sigma in 8 Minutes*, https://youtu.be/s2HCrhNVfak
[31] YouTube.com, *Six Sigma in 9 Minutes*, https://youtu.be/4EDYfSl-fmc
[32] Charles Hoffman, CPA, *Lean Six Sigma*, http://www.xbrlsite.com/mastering/Part01_Chapter02.K_LeanSixSigma.pdf
[33] PROOF, http://www.xbrlsite.com/seattlemethod/proof/documentation/index.html

% Proof, all Verification, OK %
checkReport3("http://www.xbrlsite.com/seattlemethod/proof/reference-implementation/instance.xml",
['http://www.xbrlsite.com/seattlemethod/proof/fac/ReportingStyles/PROOF-BSC-IS01-CF1_schema.xsd',
'http://www.xbrlsite.com/seattlemethod/proof/type-subtype/type-subtype2.xsd',
'http://www.xbrlsite.com/seattlemethod/proof/type-subtype/type-subtype-Things2-rules-def.xml',
'http://www.xbrlsite.com/seattlemethod/proof/disclosure-mechanics/disclosure-mechanics.xsd',
'http://www.xbrlsite.com/seattlemethod/proof/reporting-checklist/reporting-checklist-rules-def.xml',
'http://www.xbrlsite.com/seattlemethod/proof/model-structure/ModelStructure-rules-strict-def.xml'],
[newRulesFormat, saveToIPFS, extendedJSON, cacheValidity(3600)], Result, IPFSlink).

Result:

https://ipfs.infura.io/ipfs/QmThRvsiKiBMME3X9ECZcJWFDdZVDR3NptBsJvF4JQ9Uqm

As can be seen, the report is verified to be 100% valid per the set of rules that have been provided to the Pacioli logic/rules/reasoning/knowledge/insights engine.

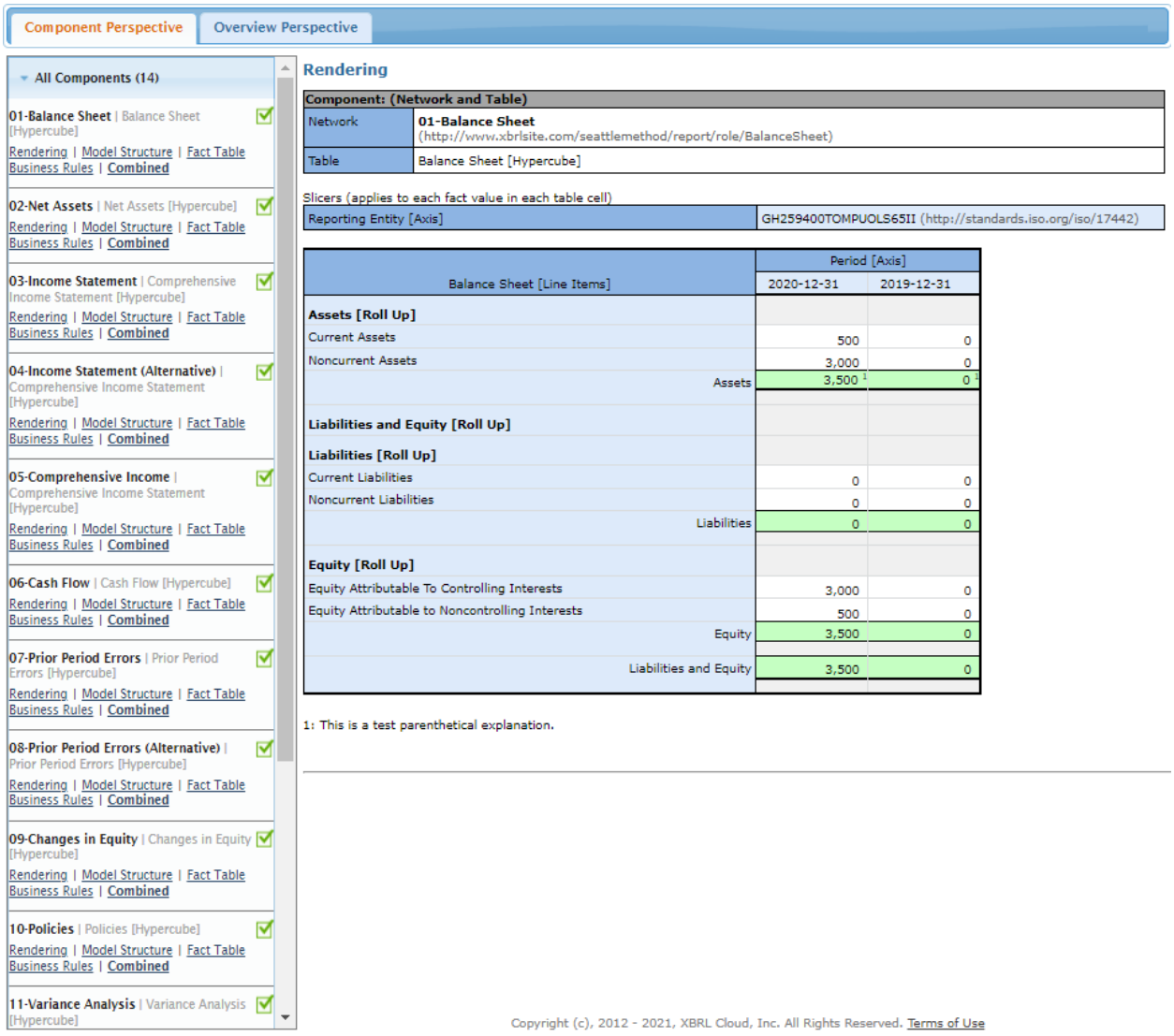


While the organization of the information per the Pacioli verification report are not necessarily optimal, the information is very useful in understanding how to construct an expert system for creating financial

reports. The full set of rules used to drive the report specification and creation can be obtained from the links above and reading other links referenced by those initial links.

XBRL Cloud Verification of Proof

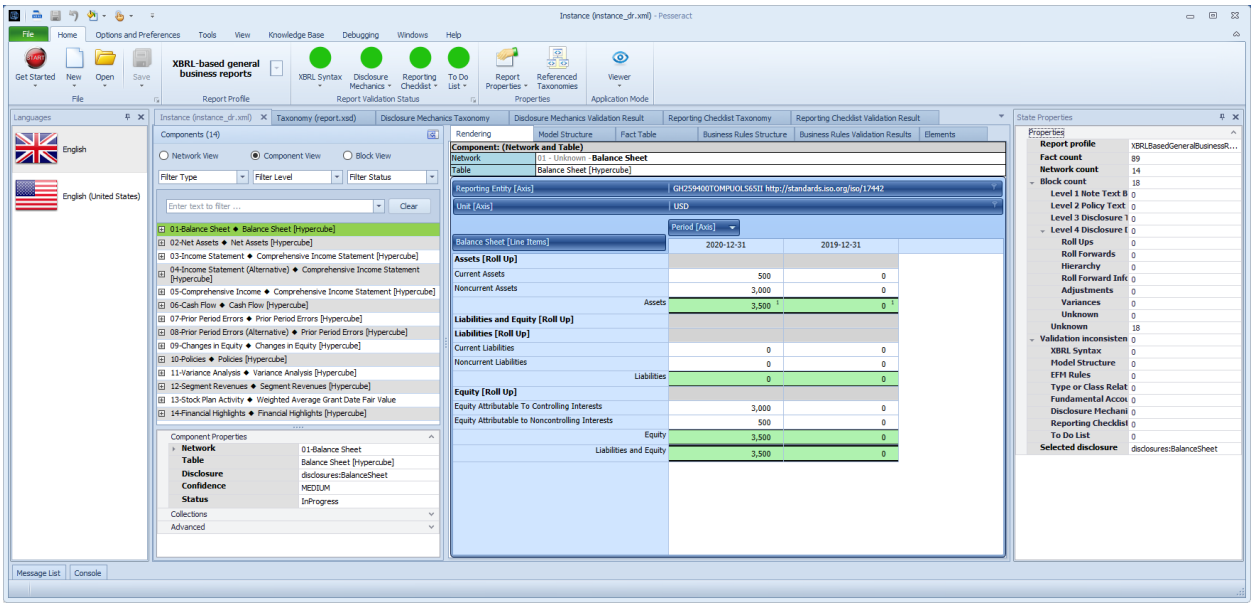
The following is a product offered by XBRL Cloud which is used to verify XBRL-based reports which they refer to as the Evidence Package[34]:



[34] XBRL Cloud Evidence Package for PROOF, http://www.xbrlsite.com/site1/seattlemethod/proof/reference-implementation/evidence-package/

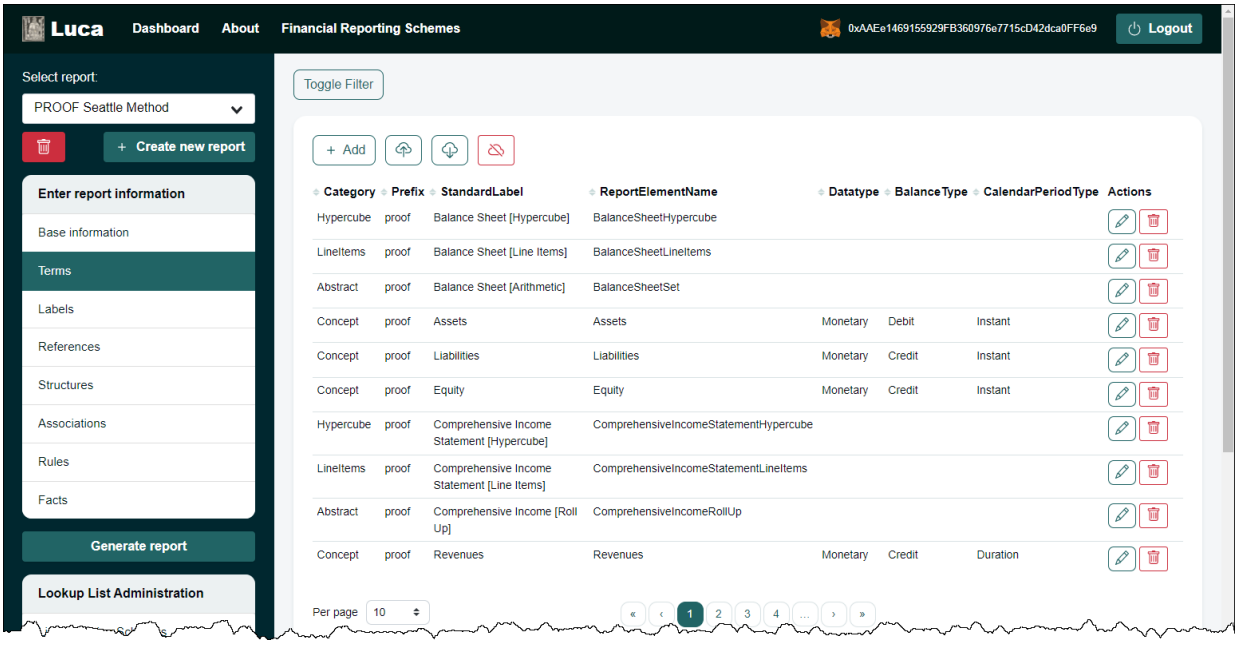
Pesseract Verification of Proof

Pesseract[35] is a working proof of concept that can be used to view and verify XBRL-based financial reports. Pesseract is an application that must be downloaded and installed (i.e. it is not a cloud based software application). If you need a license, please contact me and I will send you one.



Luca Verification of Proof

Luca[36] is a cloud-based application for creating XBRL-based financial reports.



[35] Pesseract, http://pesseract.azurewebsites.net/
[36] Luca, http://luca.yaxbrl.com/

Luca does not yet have 100% support for the Seattle Method, but it is very close and helpful in understanding XBRL-based Financial Reports[37].

Expert System for Creating Financial Reports

In this section I will explain each piece of the expert system for creating financial reports, the metadata that drives the expert system, the interaction of the pieces of the report, and other such information. I will try to stick to the interfaces of Pesseract[38] as much as possible but will add additional information to help explain the systems functionality.

The expert system is a combination of recommender systems[39], wizards, metadata, models, and other such software components. For additional information, please see to prior documents that I had previously created to explain these ideas: *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*[40] (2017) which explains Pesseract and *Guide to Building an Expert System for Creating Financial Reports*[41] (2018) which further builds on that first document.

In this document I will try and incorporate all relevant information from the prior two documents.

For the most comprehensive understanding of expert systems, I would strongly recommend that you read Frank Puppe's book *Systematic Introduction to Expert Systems*[42]. In particular, Part 1 the Introduction is extremely helpful.

Finally, special mention of CLIPS[43] is worth making. CLIPS was originally created by NASA but was then placed into the open source community. CLIPS has a lot of excellent ideas, in particular the notion of an agenda. It is worth reading through the CLIPS User Manual.

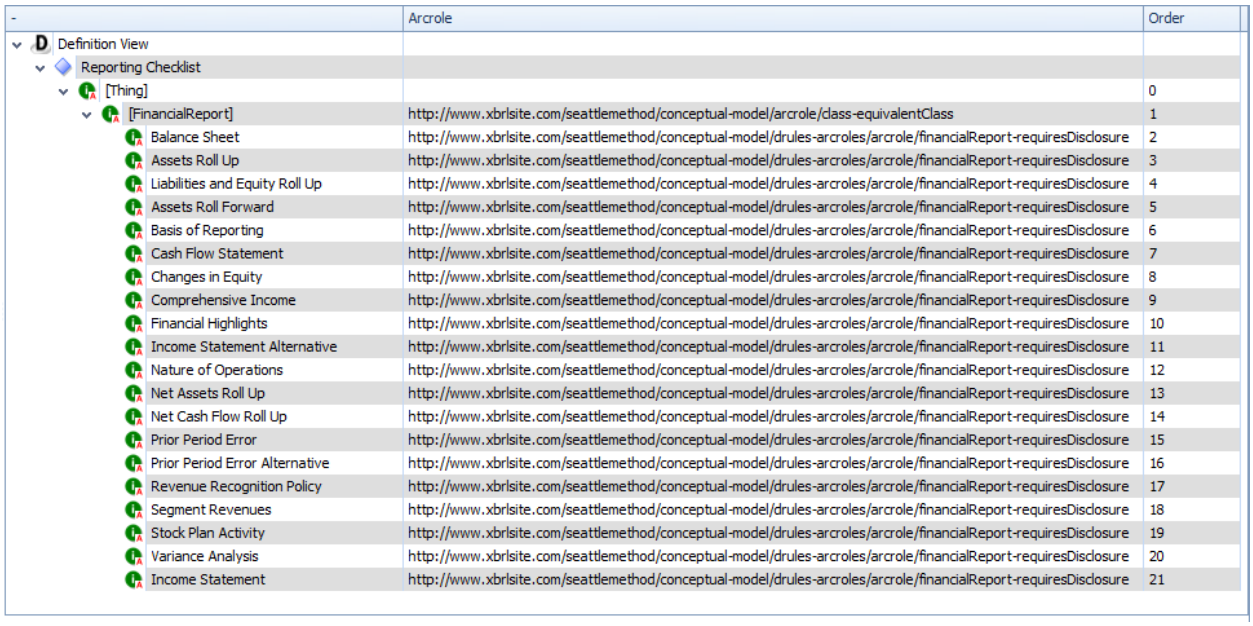
I will try and organize the examples in the most logical order that I can so that readers can understand how the application actually works. This will be way easier to explain when I can show you a real working version of this expert system for creating financial reports. That first version will likely be copied. Further creators will very likely improve upon the initial version over time.

---

[37] Cloud-based Luca, http://xbrl.squarespace.com/journal/2021/8/31/cloud-based-luca.html
[38] Pesseract, http://pesseract.azurewebsites.net/
[39] Recommender Systems, http://xbrl.squarespace.com/journal/2021/9/19/recommender-systems.html
[40] *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*, http://pesseract.azurewebsites.net/PuttingTheExpertiseIntoKnowledgeBasedSystem.pdf
[41] *Building an Expert System for Creating Financial Reports*, http://xbrlsite.azurewebsites.net/2018/Library/GuideToBuildingAnExpertSystemForCreatingFinancialReports.pdf
[42] Frank Puppe, *Systematic Introduction to Expert Systems*, https://www.amazon.com/Systematic-Introduction-Expert-Systems-Representations/dp/3642779735
[43] *Using CLIPS to Understand Expert Systems and Logic Programming*, http://xbrl.squarespace.com/journal/2016/9/15/using-clips-to-understand-expert-systems-and-logic-programmi.html

Agenda

Similar to how CLIPS works, the expert system for creating financial reports is driven by an agenda. The agenda is driven by the reporting checklist rules (disclosure rules)[44]. Viewing the reporting checklist rules in Pesseract you can see what information is included in the reporting checklist:



-	Arcrole	Order
Definition View		
Reporting Checklist		
[Thing]		0
[FinancialReport]	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-equivalentClass	1
Balance Sheet	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	2
Assets Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	3
Liabilities and Equity Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	4
Assets Roll Forward	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	5
Basis of Reporting	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	6
Cash Flow Statement	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	7
Changes in Equity	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	8
Comprehensive Income	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	9
Financial Highlights	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	10
Income Statement Alternative	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	11
Nature of Operations	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	12
Net Assets Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	13
Net Cash Flow Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	14
Prior Period Error	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	15
Prior Period Error Alternative	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	16
Revenue Recognition Policy	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	17
Segment Revenues	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	18
Stock Plan Activity	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	19
Variance Analysis	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	20
Income Statement	http://www.xbrlsite.com/seattlemethod/conceptual-model/drules-arcroles/arcrole/financialReport-requiresDisclosure	21

The reporting checklist is simply an XBRL definition linkbase that uses special arcroles to specify information about what is required to be provided within a financial report.

The terms reporting checklist and disclosure rules are used interchangeably.

While the PROOF reporting checklist is not that exciting because it only uses one arcrole which states that every mentioned disclosure is required, the US GAAP reporting checklist[45] uses the full spectrum of options that could exist within the reporting checklist:

- report-requiresDisclosure
- report-reportsConcept
- reportingLineItem-requiresDisclosure
- disclosure-allowedAlternativeDisclosure
- concept-allowedAlternativeConcept
- report-possibleDisclosure

[44] Reporting Checklist Rules, http://www.xbrlsite.com/seattlemethod/proof/reporting-checklist/reporting-checklist-rules-def.xml

[45] US GAAP Reporting Checklist (Prototype), http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/reporting-checklist/reporting-checklist-rules-def.xml

Note that reporting checklist rules are further explained in *Disclosure Rules (a.k.a. Reporting Checklist)*[46], please see that resource for further details.

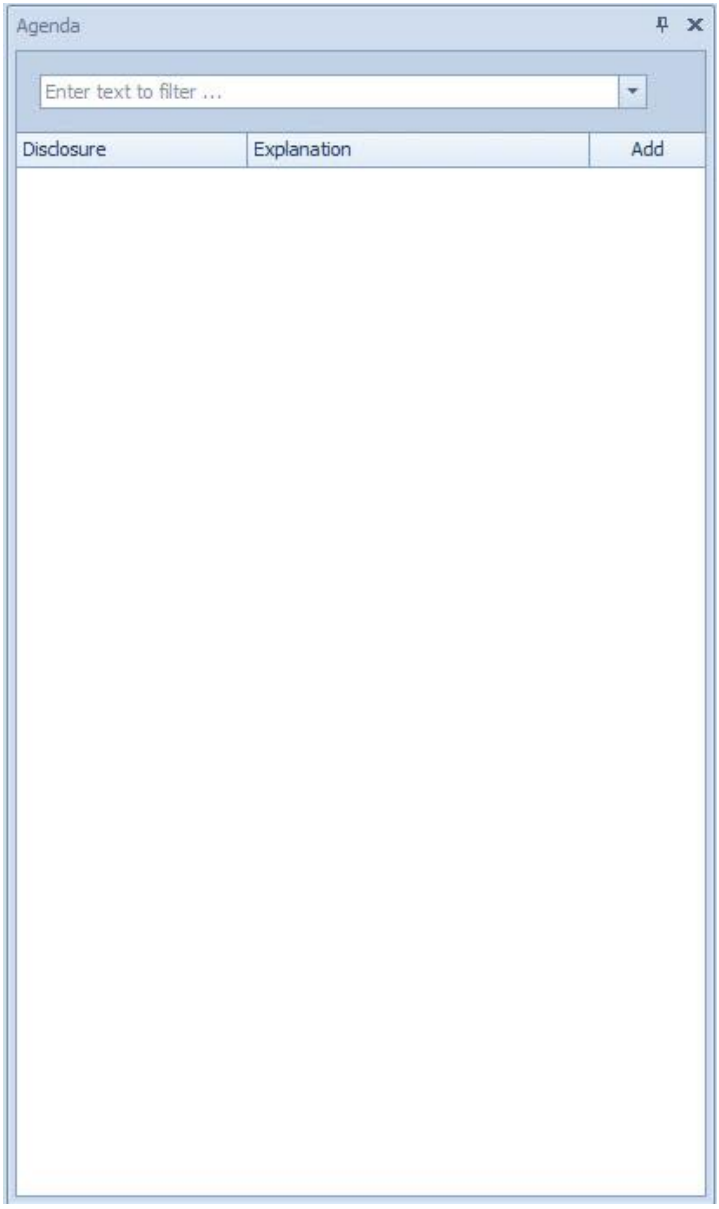
The reporting checklist rules specify what is required to be reported within a financial report for a specific financial reporting scheme, for a specific reporting style, for a specific industry within a financial reporting scheme, etc. Individual organizations could have policies for what is disclosed which can be recorded using this machine readable checklist. Combinations of reporting checklists can be used, leveraging XBRL's extensibility capabilities.

And so this is what the agenda would look like for the PROOF report before any information has been entered into the report[47]:

[46] *Disclosure Rules (a.k.a. Reporting Checklist)*, http://www.xbrlsite.com/mastering/Part02_Chapter05.N_DisclosureRules.pdf

[47] Empty Report, http://xbrlsite.com/seattlemethod/pictures/072-Agenda.jpg

Note that there is a one-to-one correlation between the reporting checklist and the agenda. This is because the XBRL definition relations drive both the reporting checklist rules and the agenda. By way of contrast, this is what the agenda looks like after the financial report has been completed[48]:

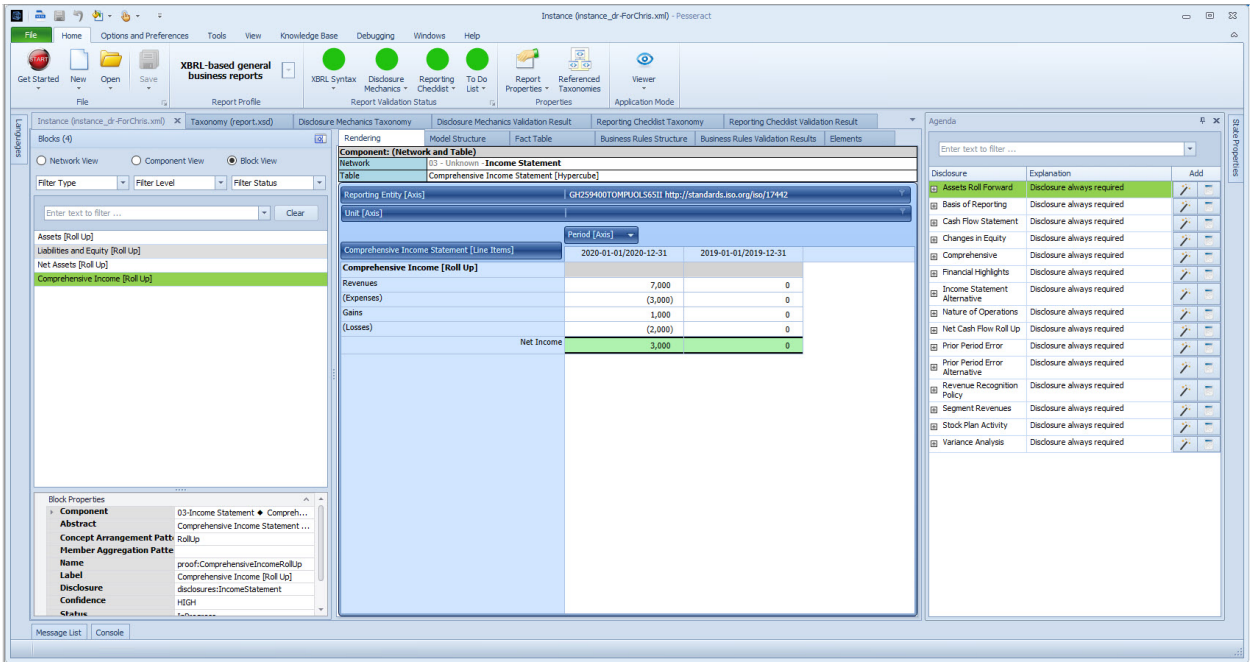


Above, the agenda is EMPTY because each required disclosure within the report which was specified within the reporting checklist has been created and exists within the report and therefore the agenda is empty.

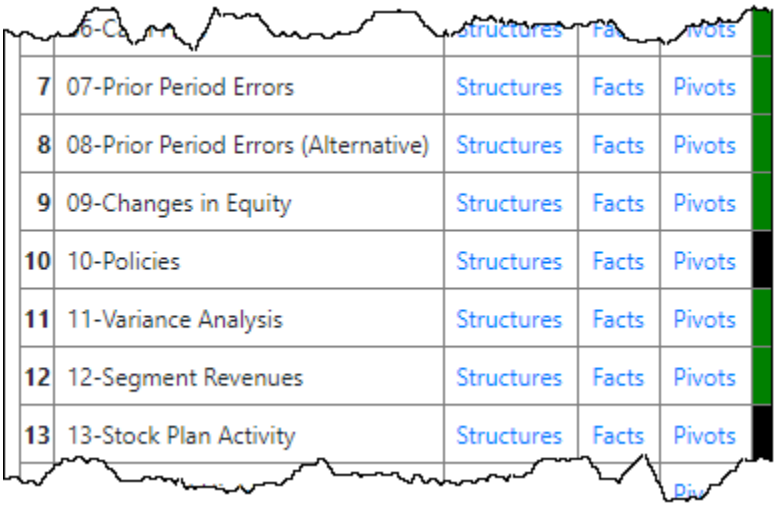
The entire process of creating each piece of a report is shown in this report creation example[49]. The graphic below[50] shows that four pieces of the report have been created (Assets [Roll Up], Liabilities and

[48] Completed Report, http://xbrlsite.com/seattlemethod/pictures/090-Agenda.jpg
[49] Report Creation Example, http://xbrlsite.com/seattlemethod/pictures/Index.html
[50] Partially completed agenda, http://xbrlsite.com/seattlemethod/pictures/076-Agenda.jpg

Equity [Roll Up], Net Assets [Roll Up], and Comprehensive Income [Roll Up]) and that 15 additional fragments of the report are still as of yet to be created:

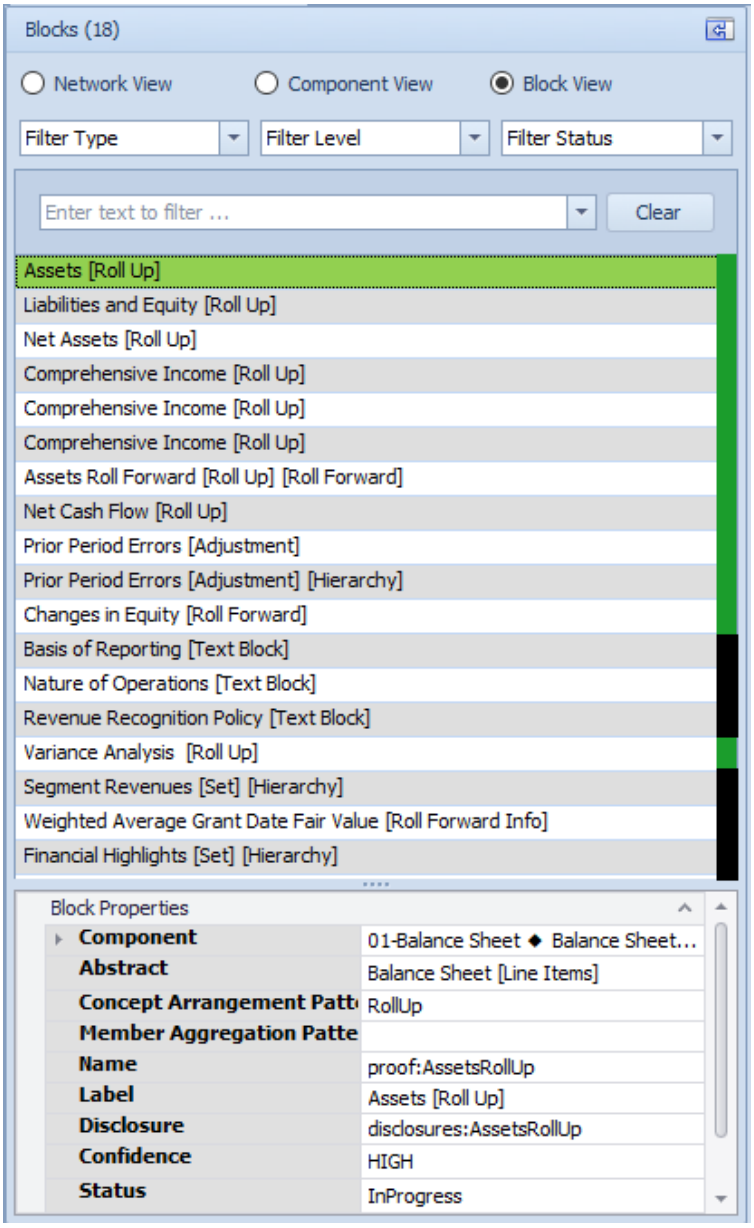


Note that Pacioli shows an additional and useful piece of information for each completed fragment of a financial report:



Above, in this case the GREEN bar shows that the report fragment has been completed and that the verification of the mathematical computations and all other verification tasks have been completed satisfactory. Because the “Policies” and “Stock Activity Plan” have no mathematical computations, the bars shown are BLACK. Had an error existed within a fragment of the report that had been created, a YELLOW or ORANGE or RED bar might be shown to indicate various different types of issues that might exist. This information relates to the project management features of the expert system is providing to

the creator of the financial report. An implementation of this feature in Pesseract might look something like the following:



Note that from the above information the user of the software can understand each fragment of the report, in the above each Block of information, is free from error. If, for example, the "Net Assets [Roll Up]" was showing as BLACK, that would mean that the rules for the roll up were missing from the report.

Note that Pacioli provides a Reporting Checklist report[51]. A problem with that report is that it is a flat list of the rules rather than a hierarchy. While the PROOF reporting checklist looks flat, generally reporting checklists are hierarchical. The US GAAP Reporting Checklist is a better representation of what a reporting checklist would tend to look like.

Identifying Disclosures

Accountants understand and work with the disclosures of a report. Effectively, a financial report is a series of disclosures organized within a sequence.

When I use the term "disclosure" I am referring to the collection of disclosures represented within the notes, policies, and each of the primary financial statements. Accountants have specific terminology and differentiate between information that is "presented" in the primary financial statements and that which is "disclosed" in the notes. Further, supplemental information that is not considered part of the notes is also covered by the term "disclosure" as I am using the term. Perhaps there might be some different better term that "disclosure" such as "logical disclosure" or something, but for now I am using the term "disclosure" and this paragraph explains how that term is defined.

Another important thing to understand is that there are several different approaches that could be used to organize disclosures within an XBRL-based financial report. One organization mechanism[52] which is many times required to be in a specific way but also can be used in arbitrary ways is the XBRL Network. Because Network use can be arbitrary, a disclosure could be contained in different networks almost arbitrarily. Another organization mechanism is the XBRL Dimensions Hypercube (a.k.a. Table). Like Networks, Hypercubes must sometimes be used in specific ways and at other times Hypercube use can be arbitrary. The final organization mechanism is what I refer to as the information block or simply Block. A Block is never arbitrary. A Block is a logical artifact that is always represented using specific physical technical syntax. A Block is simply a logical pattern that, by definition, can always be mapped to a Disclosure.

For more information on Networks and Hypercubes please see *Representing Structures*[53]. For more information related to Blocks, please see *Structures*[54].

Finally, while it is theoretically possible to provide a one-to-one correlation between a Network and a Disclosure; this is not done in practice in the US GAAP XBRL Taxonomy, the IFRS XBRL Taxonomy, or XBRL-based financial reports. While it is theoretically possible to provide a one-to-one correlation between a Hypercube and a Disclosure; and while this can be a very practical approach; this approach is also not used by the US GAAP XBRL Taxonomy, the IFRS XBRL Taxonomy, or in SEC or ESMA financial reports. Further, both the US GAAP and IFRS XBRL Taxonomies represent disclosures using the exact same Hypercube, such as "us-gaap:StatementTable" and therefore that approach cannot be used to create such a one-to-one correlation between Hypercubes and Disclosures. Further, both the US GAAP XBRL Taxonomy and IFRS XBRL Taxonomy violate a fundamental best practice recommended by XBRL International related to

---

[51] Pacioli Reporting Checklist report, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/disclosureChecks.html

[52] See Essentials of XBRL-based Digital Financial Reporting for a thorough discussion, http://xbrlsite.azurewebsites.net/2021/essentials/EssentialsOfXBRLBasedDigitalFinancialReporting.pdf

[53] *Representing Structures*, http://www.xbrlsite.com/mastering/Part02_Chapter05.H_RepresentingStructuresUsingHypercubes.pdf

[54] *Structures*, http://www.xbrlsite.com/mastering/Part02_Chapter05.E_Structures.pdf

mixing dimensional models[55]. As such, there are disclosures contained in both the US GAAP and IFRS XBRL Taxonomies that are not associational with any Hypercube.

And so this begs the question, "If you cannot identify Disclosures using Networks or Hypercubes; then how exactly do you (a) uniquely identify disclosures and (b) provide a one-to-one correlation between that object and the disclosure.

The answer to this question is to use Blocks and Prototype Theory.

Prototype Theory

Prototype theory is described in detail within Structures section 3.1056. In summary, there are two approaches to identify something. The first approach is to use a unique identifier that is guaranteed to be unique to some set of objects. This approach works great if the objects you are working with provide unique identifiers. What do you do if you don't have unique identifiers within the set of objects you are working with? The second approach to identifying something is prototype theory. According to prototype theory, objects can be defined by their resemblance to a unique prototype that is a best or most typical example of the essence of the object, sharing the maximum number of features or traits with that prototype. A prototype consists of characteristic features.

Disclosure Mechanics Rules

Disclosure mechanics[57] rules provide the necessary information that allows software to (a) describe the essence of a specific disclosure basically providing a logical schema for a disclosure in machine readable form, (b) use that machine readable essence, ala prototype theory, to discover the information Block that is used to represent that Disclosure and, (c) verify that the representation of that Disclosure is consistent with the logical schema.

The Disclosures for the PROOF[58] can help you understand how all this works. First, Disclosures are defined in the form of the elements within an XBRL taxonomy schema[59]. Then, disclosure mechanics rules are created for each disclosure using XBRL definition relations and a specific set of arcroles that represent logical characteristics that make up the essence of each Disclosure and the XBRL definition relations are hooked together within an XBRL taxonomy schema[60].

The PROOF disclosure mechanics rules are pretty small, easy to understand, and don't really represent what the disclosure rules can do. A better example to explain these disclosure mechanics rules is the US

---

[55] *XBRL International Guidance Clarifies XBRL Dimensions Semantics*, http://xbrl.squarespace.com/journal/2015/3/31/xbrl-international-guidance-clarifies-xbrl-dimensions-semant.html

[56] Structures, Section 3.10, *Prototype Theory*, http://www.xbrlsite.com/mastering/Part02_Chapter05.E_Structures.pdf

[57] Disclosure Mechanics, http://www.xbrlsite.com/mastering/Part02_Chapter05.M_DisclosureMechnics.pdf

[58] PROOF, Disclosures, http://www.xbrlsite.com/seattlemethod/proof/documentation/Disclosures.html

[59] Disclosures, XBRL Taxonomy Schema, http://www.xbrlsite.com/seattlemethod/proof/disclosures-topics/disclosures.xsd

[60] Disclosure Mechanics Rules, XBRL Taxonomy Schema, http://www.xbrlsite.com/seattlemethod/proof/disclosure-mechanics/disclosure-mechanics.xsd

GAAP XBRL Taxonomy disclosure rules. Here is an example for the Disclosure "Inventory, Net (Current) [Roll Up]"[61]:

Natural language:

```
Rules for disclosure: disclosures:InventoryNetRollUp
This disclosure:

   - MUST be represented as the Concept Arrangement Pattern: cm:RollUp
      - cm:RollUp REQUIRES total concept: us-gaap:InventoryNet
         - OR alternative total concept: us-gaap:InventoryNetOfAllowancesCustomerAdvancesAndProgressBillings
         - OR alternative total concept: us-gaap:PublicUtilitiesInventory
         - OR alternative total concept: us-gaap:AirlineRelatedInventory
         - OR alternative total concept: us-gaap:RetailRelatedInventory
         - OR alternative total concept: us-gaap:EnergyRelatedInventory
         - OR alternative total concept: us-gaap:AgriculturalRelatedInventory
   - MUST be represented as using the Level 3 Disclosure [Text Block]: us-gaap:ScheduleOfInventoryCurrentTableTextBlock
      - OR alternative Level 3 Disclosure [Text Block]: us-gaap:ScheduleOfUtilityInventoryTextBlock
   - REQUIRES the policy to be reported using the Level 2 Policy Text Block concept: us-gaap:InventoryPolicyTextBlock
      - OR alternative Level 2 Policy Text Block concept: us-gaap:InventoryMajorClassesPolicy
      - OR alternative Level 2 Policy Text Block concept: us-gaap:InventorySuppliesPolicy
      - OR alternative Level 2 Policy Text Block concept: us-gaap:InventoryWorkInProcessPolicy
      - OR alternative Level 2 Policy Text Block concept: us-gaap:InventoryFinishedGoodsPolicy
   - REQUIRES the note to be reported using the Level 1 Note Text Block concept: us-gaap:InventoryDisclosureTextBlock
```

Here are the 20 Disclosures that are represented in the PROOF example:

Primary Information

#	Disclosure	Category	Level	Pattern	Disclosure ...	Disclosure ...	Applicable	Representation Concept [TEXT BLOCK]	Representation Concept DETAIL
1	Assets Roll Forward	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Assets
2	Assets Roll Up	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Assets

Rules | Line of Reasoning

This disclosure: disclosures:AssetsRollUp

- MUST be represented using the Hypercube/[Table] named: proof:BalanceSheetHypercube
- MUST be represented as a **Level 4 Disclosure Detail** with the concept arrangement pattern: cm:RollUp
  - cm:RollUp REQUIRES total: proof:Assets

#	Disclosure	Category	Level	Pattern	Disclosure ...	Disclosure ...	Applicable	Representation Concept [TEXT BLOCK]	Representation Concept DETAIL
3	Balance Sheet	Unknown	UNKNOWN	Component	True	CONSISTENT	True	-	-
4	Basis of Reporting	Unknown	Level1TextBlock	TextBlock	True	CONSISTENT	True	Basis of Reporting [Text Block]	NOT-EXPECTED
5	Cash Flow Statement	Unknown	UNKNOWN	Component	True	CONSISTENT	True	-	-
6	Changes in Equity	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Equity
7	Comprehensive Income	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Comprehensive Income
8	Financial Highlights	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT-EXPECTED	Revenues
9	Income Statement	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Income
10	Income Statement Alternative	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Income
11	Liabilities and Equity Roll Up	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Liabilities and Equity
12	Nature of Operations	Unknown	Level1TextBlock	TextBlock	True	CONSISTENT	True	Nature of Operations [Text Block]	NOT-EXPECTED
13	Net Assets Roll Up	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Assets
14	Net Cash Flow Roll Up	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Cash Flow
15	Prior Period Error	Unknown	Level4Detail	Adjustment	True	CONSISTENT	True	NOT-EXPECTED	Equity
16	Prior Period Error Alternative	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT-EXPECTED	Equity
17	Revenue Recognition Policy	Unknown	Level1TextBlock	TextBlock	True	CONSISTENT	True	Revenue Recognition Policy [Text Block]	NOT-EXPECTED
18	Segment Revenues	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT-EXPECTED	Revenues
19	Stock Plan Activity	Unknown	Level4Detail	RollForwar...	True	CONSISTENT	True	NOT-EXPECTED	Nonvested Fair Value
20	Variance Analysis	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Income

Notice the "Rule" and the "Line of Reasoning" for the Assets Roll Up Disclosure. That same information exists for each Disclosure. You cannot see the Line of Reasoning above, so we show that below:

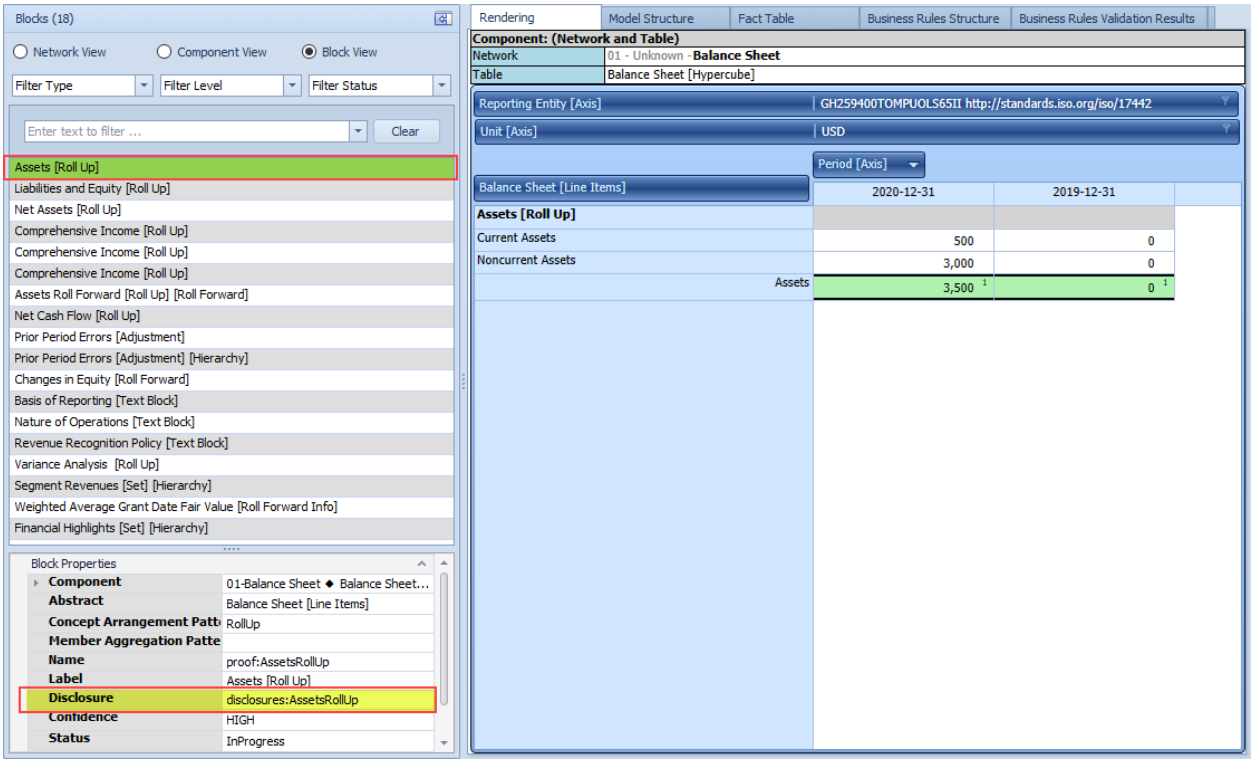
[61] US GAAP, Inventory, Net (Current) [Roll Up], http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/disclosures-topics/disclosures-detail/Disclosure-517.html

Rules | Line of Reasoning

####Disclosure mechanics validation explanation for disclosure: disclosures:AssetsRollUp####
Level 4 Disclosure Detail
Looking for blocks with concept arrangement pattern: RollUp
Looking for Hypercube/[Table]: proof:BalanceSheetHypercube
*FOUND Hypercube/[Table]: proof:BalanceSheetHypercube in network:
Looking for Concept: proof:Assets
*FOUND Concept: proof:Assets in network:
Concept located in network: 01-Balance Sheet
CONCLUSION
Disclosure found in report: True
Disclosure mechanics are CONSISTENT because the Level 4 Disclosure Detail concept was FOUND.
END of disclosure mechanics validation explanation for this disclosure

The Line of Reasoning is provided by the logic/rules/reasoning/knowledge/insights engine and can help the application user understand why a Disclosure was discovered or why it was not discovered. This is useful for debugging disclosure mechanics rules as they are being created and debugging reports to make them properly functioning financial reports.

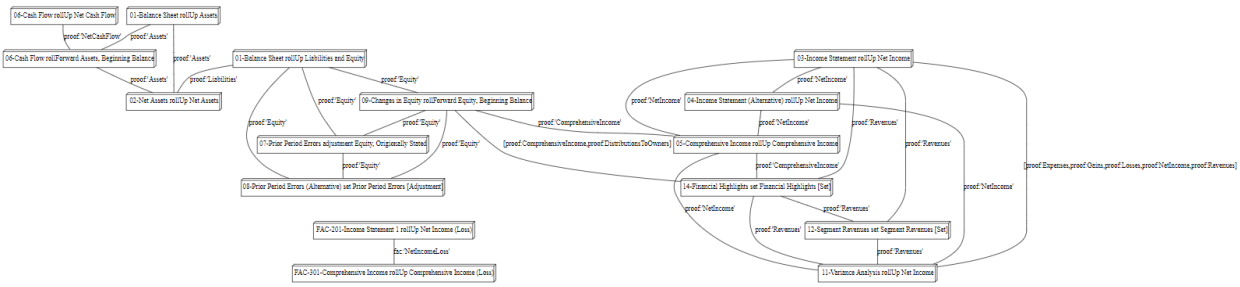
If you go back to the financial report in Pesseract you can see that every information Block has been identified as being a specific Disclosure. This is how the Agenda works. The reporting checklist specifies what disclosures need to exist in the report and the disclosure mechanics rules is used to find the actual disclosure in the report.



Pacioli provides another approach to viewing this information[62]:

#	Network	Hypercube	Block	Disclosures
1	01-Balance Sheet	Balance Sheet [Hypercube]	Assets [RollUp]	disclosures:AssetsRollUp, disclosures:BalanceSheet
2	01-Balance Sheet	Balance Sheet [Hypercube]	Liabilities and Equity [RollUp]	disclosures:BalanceSheet, disclosures:LiabilitiesAndEquityRollUp
3	02-Net Assets	Net Assets [Hypercube]	Net Assets [RollUp]	disclosures:NetAssetsRollUp
4	03-Income Statement	Comprehensive Income Statement [Hypercube]	Net Income [RollUp]	disclosures:IncomeStatement
5	04-Income Statement (Alternative)	Comprehensive Income Statement [Hypercube]	Net Income [RollUp]	disclosures:IncomeStatement, disclosures:IncomeStatementAlternative
6	05-Comprehensive Income	Comprehensive Income Statement [Hypercube]	Comprehensive Income [RollUp]	disclosures:ComprehensiveIncome
7	06-Cash Flow	Cash Flow [Hypercube]	Net Cash Flow [RollUp]	disclosures:CashFlowStatement, disclosures:NetCashFlowRollUp
8	06-Cash Flow	Cash Flow [Hypercube]	Assets, Beginning Balance [RollForward]	disclosures:AssetsRollForward, disclosures:CashFlowStatement
9	07-Prior Period Errors	Prior Period Errors [Hypercube]	Equity, Origionally Stated [Adjustment]	disclosures:PriorPeriodError
10	08-Prior Period Errors (Alternative)	Prior Period Errors [Hypercube]	Prior Period Errors [Adjustment] [Hierarchy]	disclosures:PriorPeriodErrorAlternative
11	09-Changes in Equity	Changes in Equity [Hypercube]	Equity, Beginning Balance [RollForward]	disclosures:ChangesInEquity
12	10-Policies	Policies [Hypercube]	Basis of Reporting [Text Block] [Level1TextBlock]	disclosures:BasisOfReporting
13	10-Policies	Policies [Hypercube]	Nature of Operations [Text Block] [Level1TextBlock]	disclosures:NatureOfOperations
14	10-Policies	Policies [Hypercube]	Revenue Recognition Policy [Text Block] [Level1TextBlock]	disclosures:RevenueRecognitionPolicy
15	11-Variance Analysis	Variance Analysis [Hypercube]	Net Income [RollUp]	disclosures:VarianceAnalysis
16	12-Segment Revenues	Segment Revenues [Hypercube]	Segment Revenues [Set] [Hierarchy]	disclosures:SegmentRevenues
17	13-Stock Plan Activity	Weighted Average Grant Date Fair Value [Hypercube]	Nonvested Fair Value, Beginning Balance [RollForwardInfo]	disclosures:StockPlanActivity
18	14-Financial Highlights	Financial Highlights [Hypercube]	Financial Highlights [Set] [Hierarchy]	disclosures:FinancialHighlights

Pacioli's organization is not optimal because it combines some Disclosures together in the same cell. What would be better is a completely new report that provides a list of actual Disclosures rather than grouping the Disclosures as they are shown above. The Disclosures Mechanics list[63] provides that information, but no way to navigate to the actual Disclosure. But Pacioli also provides an interesting Blocks Graph view which shows the logical relationships between the information Blocks[64]:



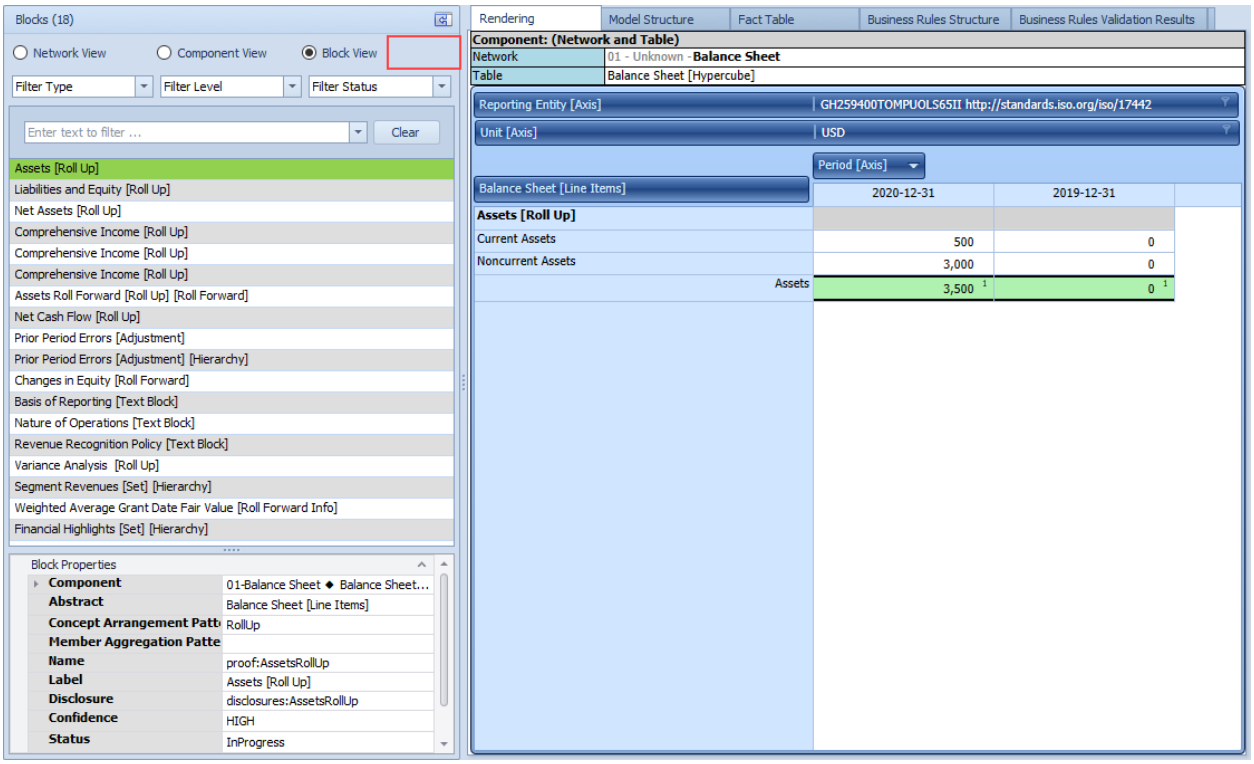
[62] Pacioli Blocks report, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/blocks.html

[63] Pacioli Disclosure Mechanics list, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/disclosures.html

[64] Pacioli Blocks Graph report, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/blocksGraph.html

This graph view could very likely be better organized to help understand the report information, navigate to the different Blocks that make up a financial report, and other such things.

The Pacioli view of the Blocks helps you see an improvement that could be made to the Pesseract software application. See the screen shot of the user interface used to review the report fragments of the financial statement. Imagine the addition of another radio button at the top of the form that would provide the "Disclosure View" of the report:

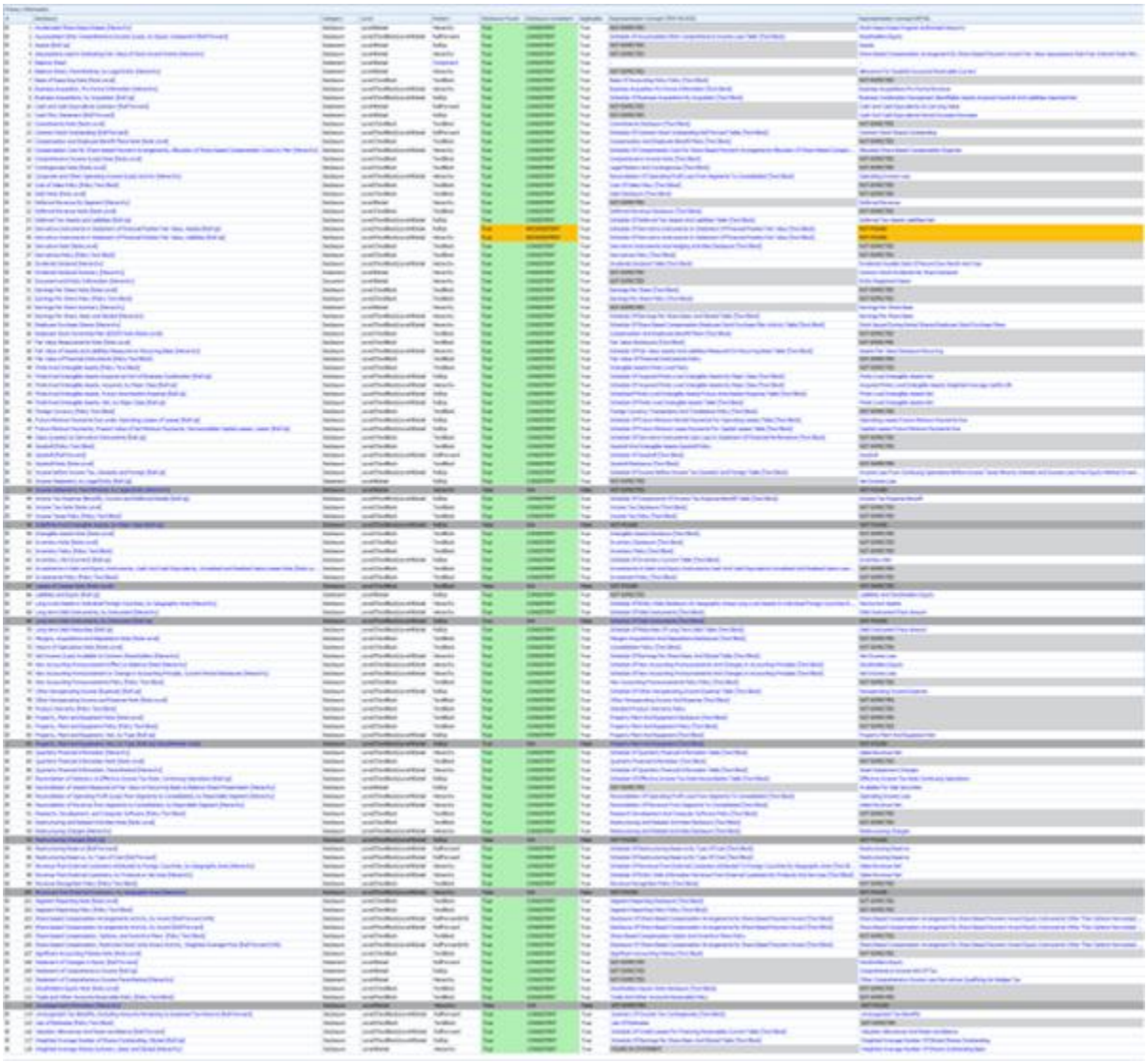


Adding that Disclosure view would provide a complete set of views of a financial report: by Network, by Network + Hypercube (called Component), by Block, and now by Disclosure.

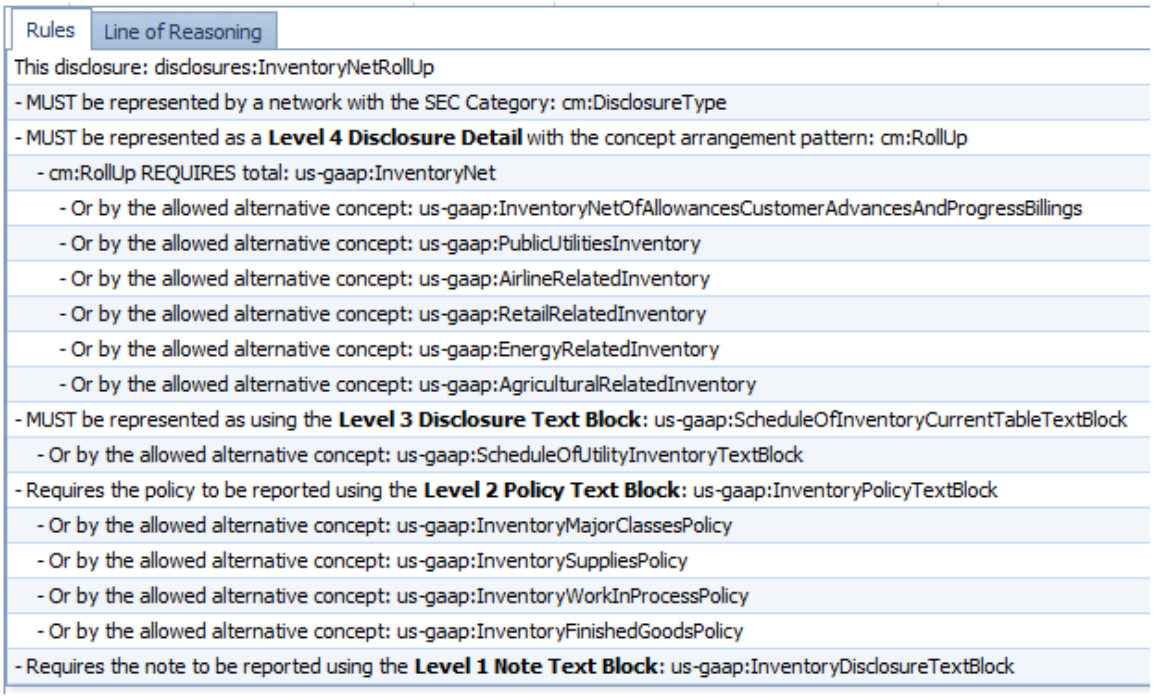
NOTE: Notice that there are 18 information Blocks and 20 Disclosures. That is because there are two disclosures, the "Balance Sheet" and the "Cash Flow Statement" that are Disclosures that have no Blocks of their own, rather they are made up of other Blocks. The Balance Sheet is made up of the two Blocks, Assets [Roll Up] and Liabilities and Equity [Roll Up]. The Cash Flow Statement is made up of the "Net Cash Flow [Roll Up] and the Assets [Roll Forward].

The following is a summary of the disclosures within the Microsoft 10-K for 2017[65] that were identified and verified using US GAAP disclosure mechanics rules:

[65] Disclosure Mechanics Verification Results, http://xbrlsite.azurewebsites.net/2022/Library/DisclosureMechanics_Microsoft.jpg



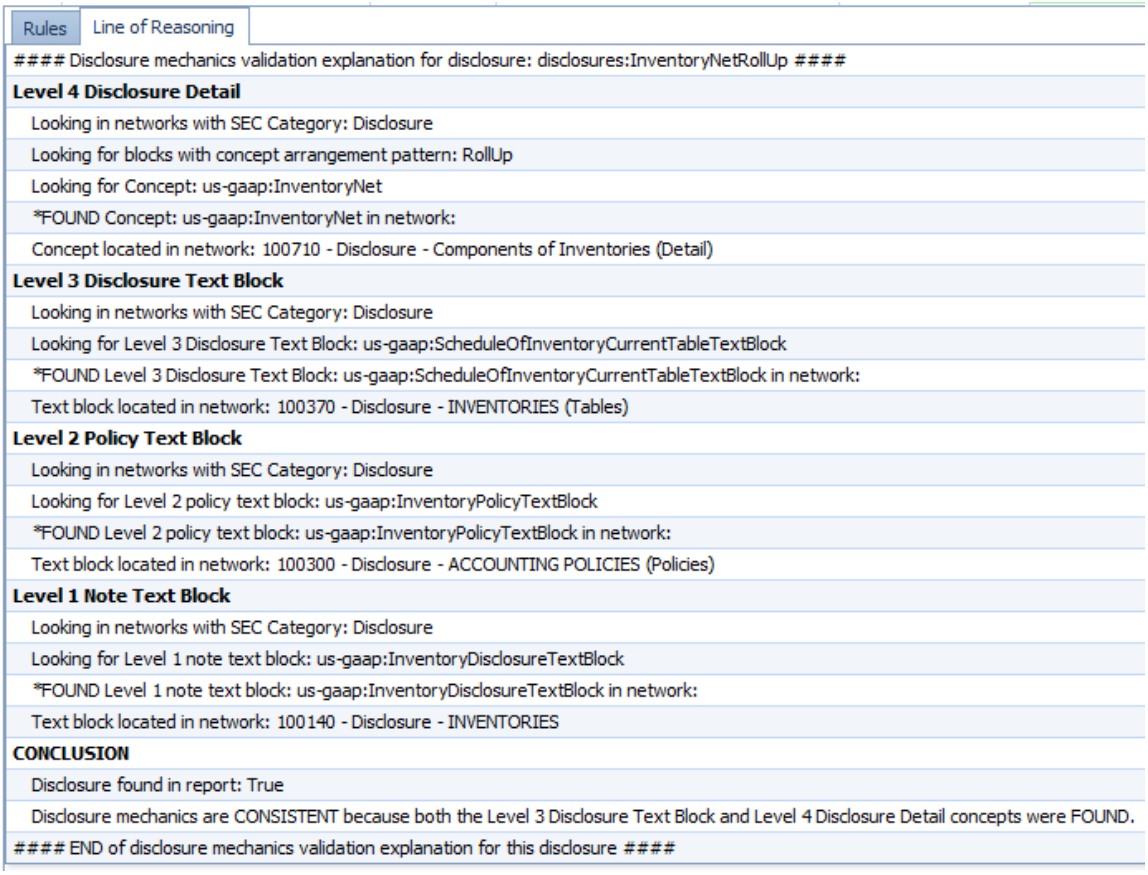
US GAAP related Disclosure Rule:



Rules | Line of Reasoning

This disclosure: disclosures:InventoryNetRollUp
- MUST be represented by a network with the SEC Category: cm:DisclosureType
- MUST be represented as a **Level 4 Disclosure Detail** with the concept arrangement pattern: cm:RollUp
  - cm:RollUp REQUIRES total: us-gaap:InventoryNet
    - Or by the allowed alternative concept: us-gaap:InventoryNetOfAllowancesCustomerAdvancesAndProgressBillings
    - Or by the allowed alternative concept: us-gaap:PublicUtilitiesInventory
    - Or by the allowed alternative concept: us-gaap:AirlineRelatedInventory
    - Or by the allowed alternative concept: us-gaap:RetailRelatedInventory
    - Or by the allowed alternative concept: us-gaap:EnergyRelatedInventory
    - Or by the allowed alternative concept: us-gaap:AgriculturalRelatedInventory
- MUST be represented as using the **Level 3 Disclosure Text Block**: us-gaap:ScheduleOfInventoryCurrentTableTextBlock
  - Or by the allowed alternative concept: us-gaap:ScheduleOfUtilityInventoryTextBlock
- Requires the policy to be reported using the **Level 2 Policy Text Block**: us-gaap:InventoryPolicyTextBlock
  - Or by the allowed alternative concept: us-gaap:InventoryMajorClassesPolicy
  - Or by the allowed alternative concept: us-gaap:InventorySuppliesPolicy
  - Or by the allowed alternative concept: us-gaap:InventoryWorkInProcessPolicy
  - Or by the allowed alternative concept: us-gaap:InventoryFinishedGoodsPolicy
- Requires the note to be reported using the **Level 1 Note Text Block**: us-gaap:InventoryDisclosureTextBlock

US GAAP related Line of Reasoning:



Rules | Line of Reasoning

Disclosure mechanics validation explanation for disclosure: disclosures:InventoryNetRollUp

Level 4 Disclosure Detail

Looking in networks with SEC Category: Disclosure
Looking for blocks with concept arrangement pattern: RollUp
Looking for Concept: us-gaap:InventoryNet
*FOUND Concept: us-gaap:InventoryNet in network:
Concept located in network: 100710 - Disclosure - Components of Inventories (Detail)

Level 3 Disclosure Text Block

Looking in networks with SEC Category: Disclosure
Looking for Level 3 Disclosure Text Block: us-gaap:ScheduleOfInventoryCurrentTableTextBlock
*FOUND Level 3 Disclosure Text Block: us-gaap:ScheduleOfInventoryCurrentTableTextBlock in network:
Text block located in network: 100370 - Disclosure - INVENTORIES (Tables)

Level 2 Policy Text Block

Looking in networks with SEC Category: Disclosure
Looking for Level 2 policy text block: us-gaap:InventoryPolicyTextBlock
*FOUND Level 2 policy text block: us-gaap:InventoryPolicyTextBlock in network:
Text block located in network: 100300 - Disclosure - ACCOUNTING POLICIES (Policies)

Level 1 Note Text Block

Looking in networks with SEC Category: Disclosure
Looking for Level 1 note text block: us-gaap:InventoryDisclosureTextBlock
*FOUND Level 1 note text block: us-gaap:InventoryDisclosureTextBlock in network:
Text block located in network: 100140 - Disclosure - INVENTORIES

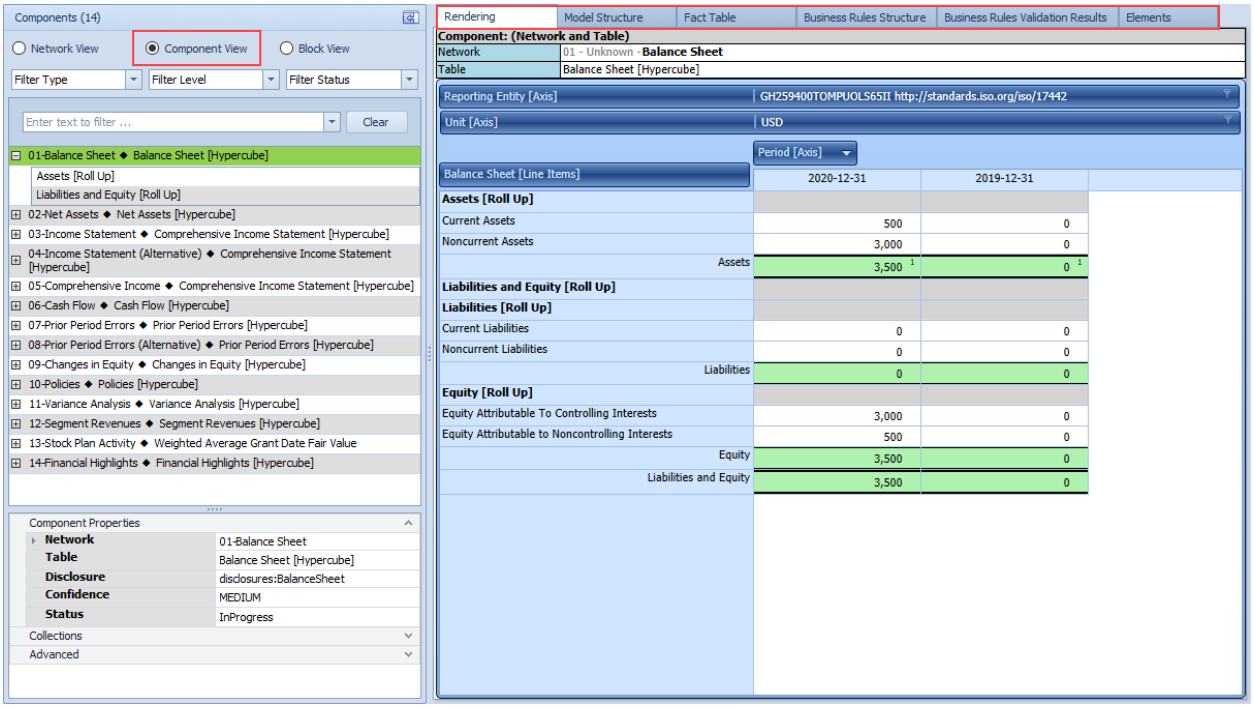
CONCLUSION

Disclosure found in report: True
Disclosure mechanics are CONSISTENT because both the Level 3 Disclosure Text Block and Level 4 Disclosure Detail concepts were FOUND.

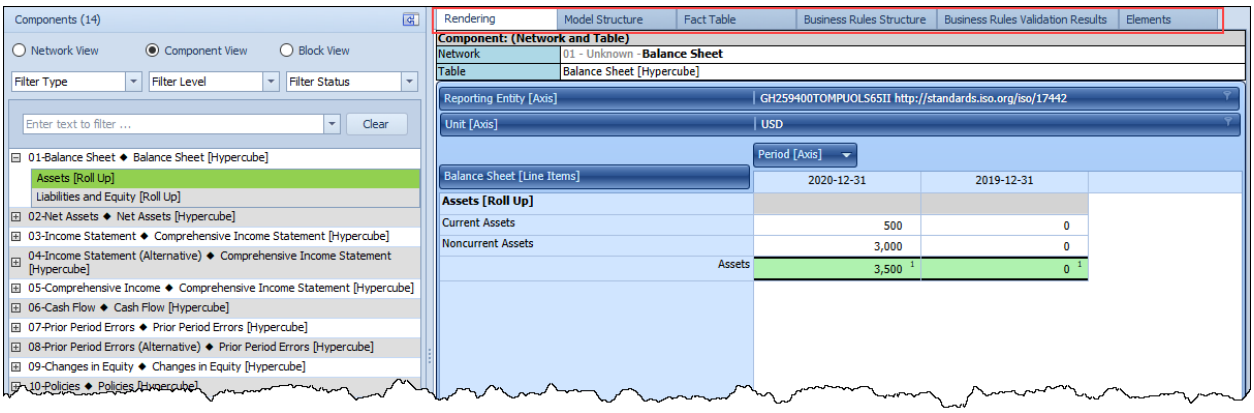
END of disclosure mechanics validation explanation for this disclosure

Viewing Disclosures (Report Fragments)

The graphic below shows the Components view of the report fragments (remembering that a Component is a Network PLUS a Hypercube). Note that if the Component is expanded you see the Blocks that make up the Component. Below you see that the "Balance Sheet <> Balance Sheet [Hypercube]" is made up of two information Blocks, "Assets [Roll Up]" and "Liabilities and Equity [Roll Up]". If you were to click on either of the Blocks the view in the right pane would be changed to show only that Block in the rendering view[66].



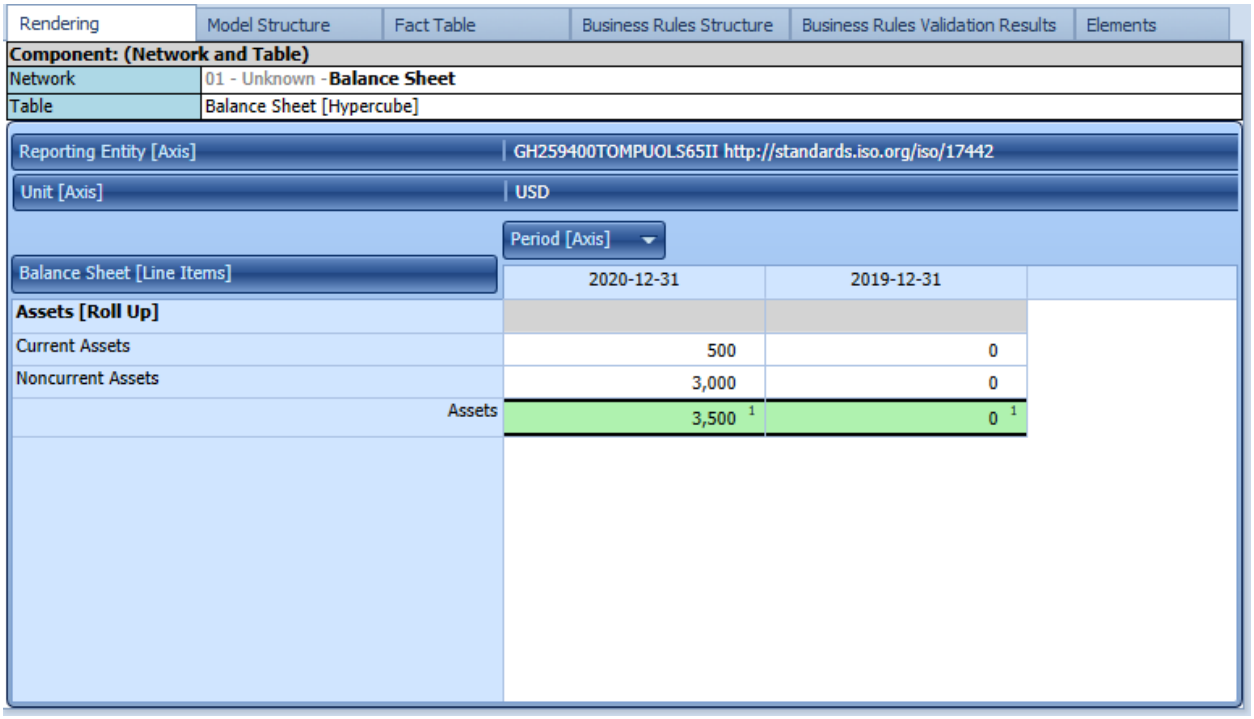
For example, we click on the "Assets [Roll Up]" information Block:



For each Disclosure, Information Block, Component or Network you can get six different views:

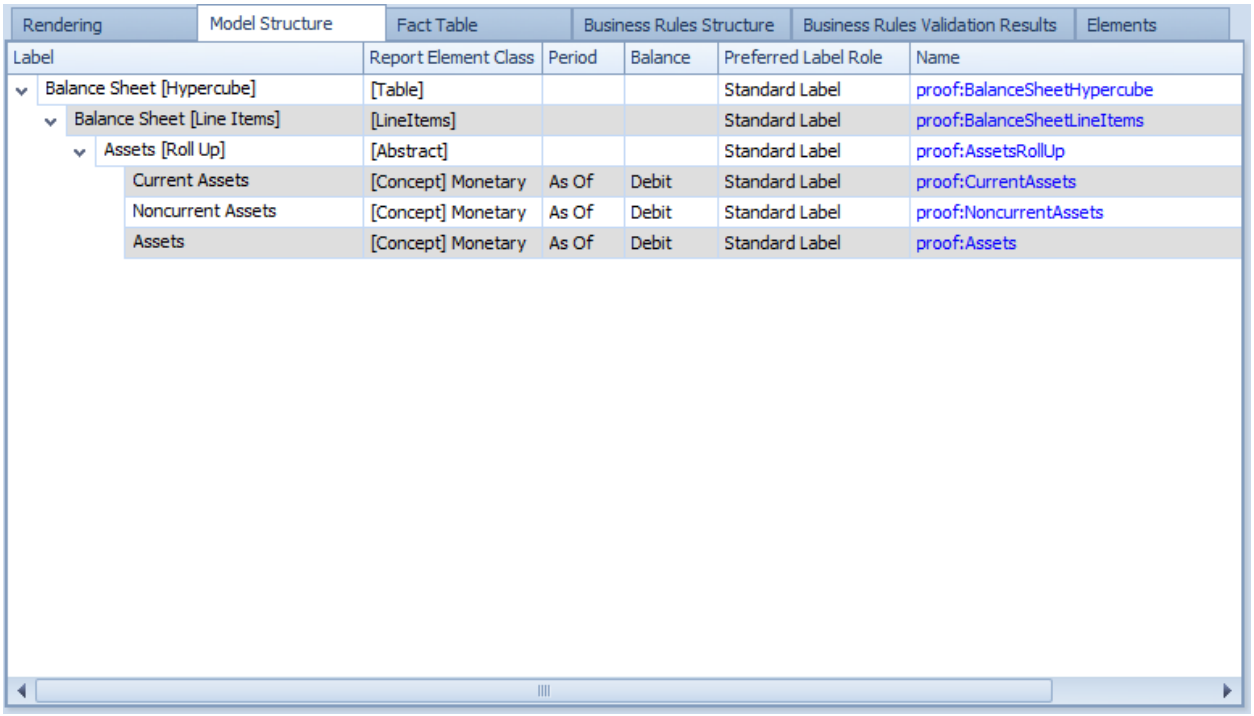
[66] Report Fragment Views, http://xbrlsite.com/seattlemethod/pictures2/Index.html

Rendering View:

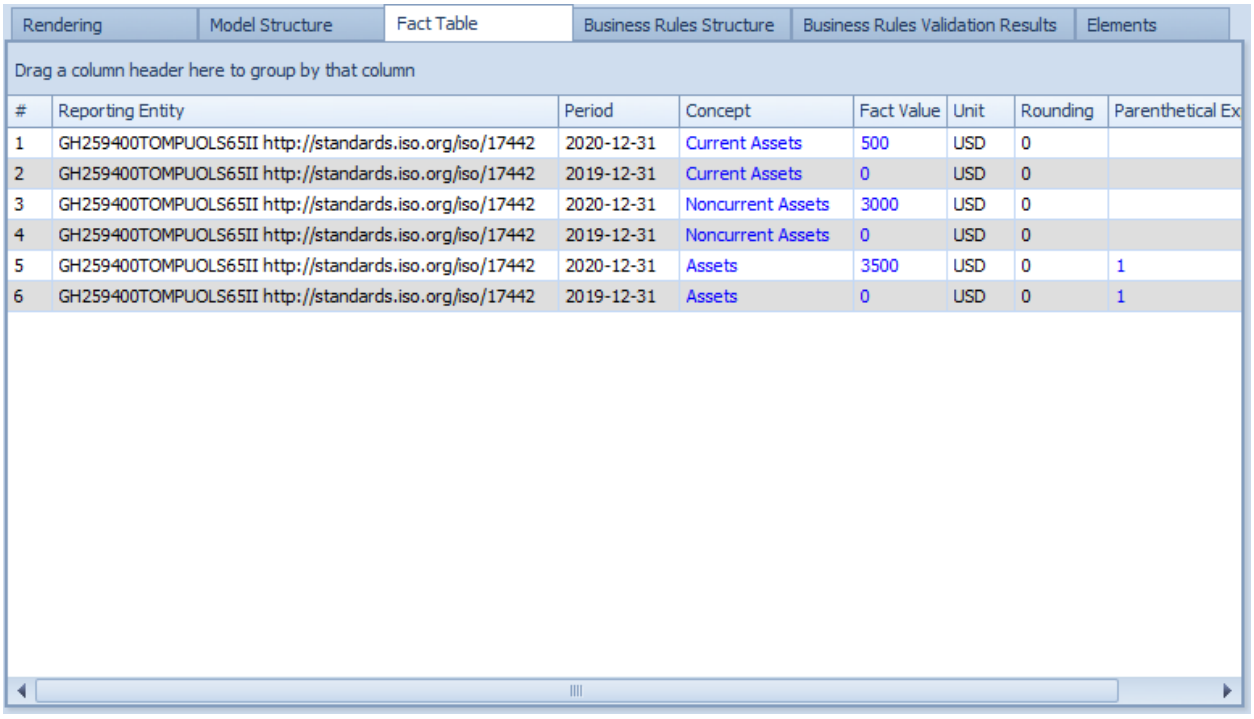


The rendering is dynamically created from information contained in each of the other views of the report. Further, the rendering is a pivot table that the user can reconfigure.

Model Structure View:



Fact Table View:

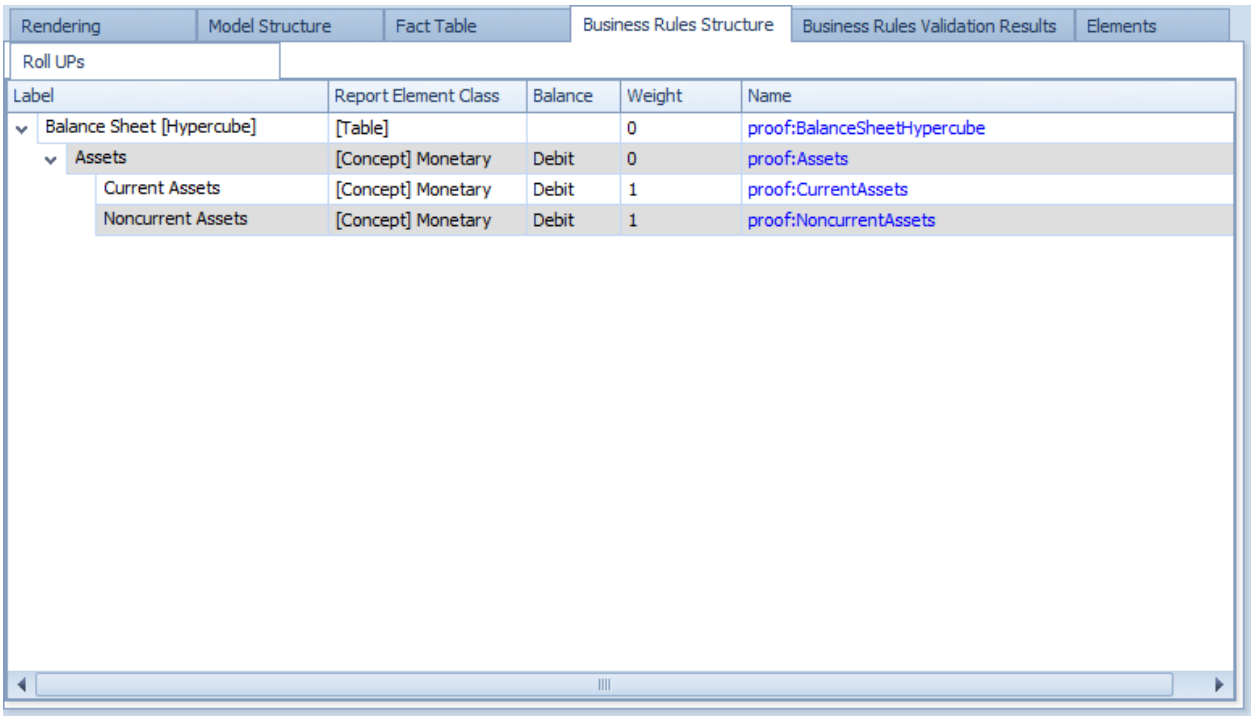


Rendering | Model Structure | Fact Table | Business Rules Structure | Business Rules Validation Results | Elements

Drag a column header here to group by that column

#	Reporting Entity	Period	Concept	Fact Value	Unit	Rounding	Parenthetical Ex
1	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-12-31	Current Assets	500	USD	0	
2	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2019-12-31	Current Assets	0	USD	0	
3	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-12-31	Noncurrent Assets	3000	USD	0	
4	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2019-12-31	Noncurrent Assets	0	USD	0	
5	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2020-12-31	Assets	3500	USD	0	1
6	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442	2019-12-31	Assets	0	USD	0	1

Rules View: (Mathematical rules)

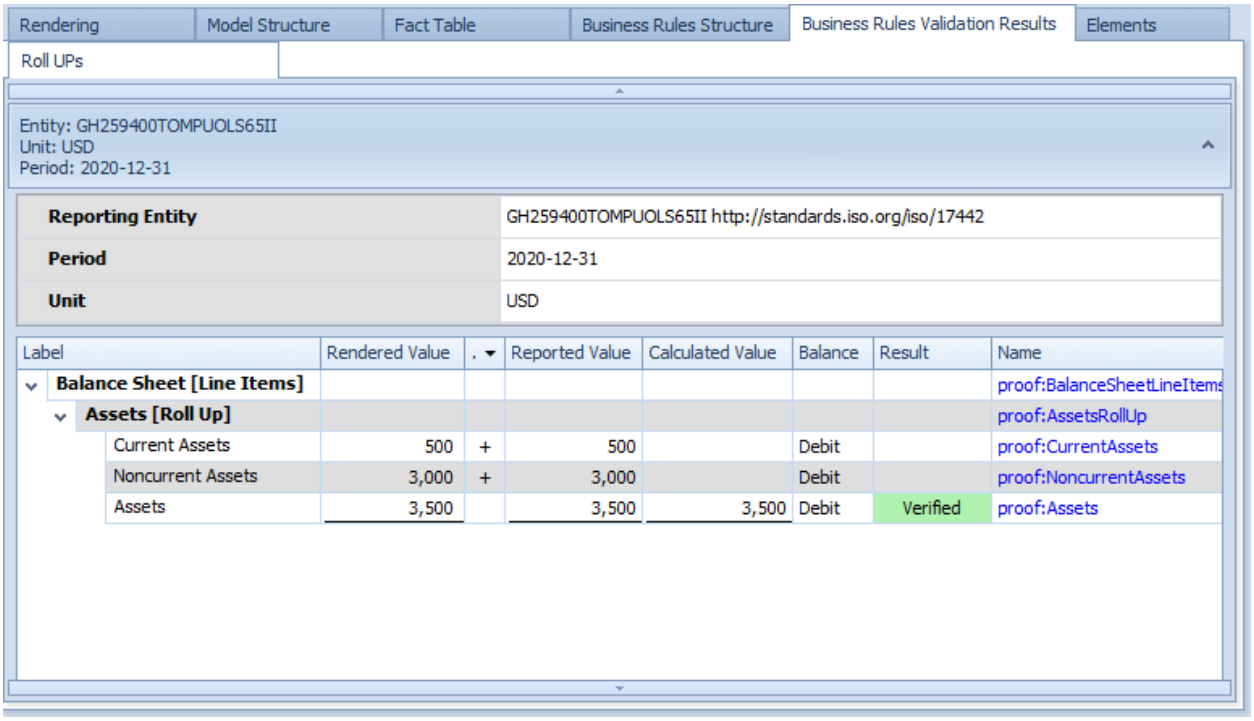


Rendering | Model Structure | Fact Table | Business Rules Structure | Business Rules Validation Results | Elements

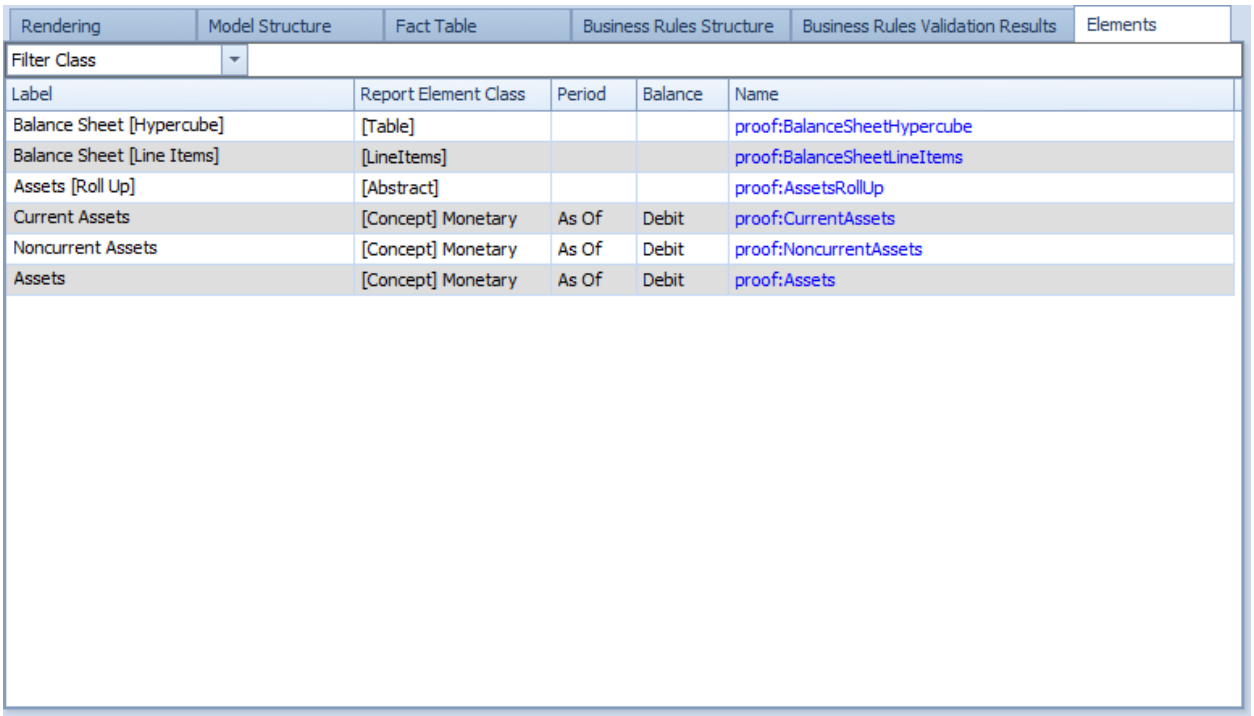
Roll UPs

Label	Report Element Class	Balance	Weight	Name
Balance Sheet [Hypercube]	[Table]		0	proof:BalanceSheetHypercube
Assets	[Concept] Monetary	Debit	0	proof:Assets
Current Assets	[Concept] Monetary	Debit	1	proof:CurrentAssets
Noncurrent Assets	[Concept] Monetary	Debit	1	proof:NoncurrentAssets

Validation Results View:



Report Elements View:



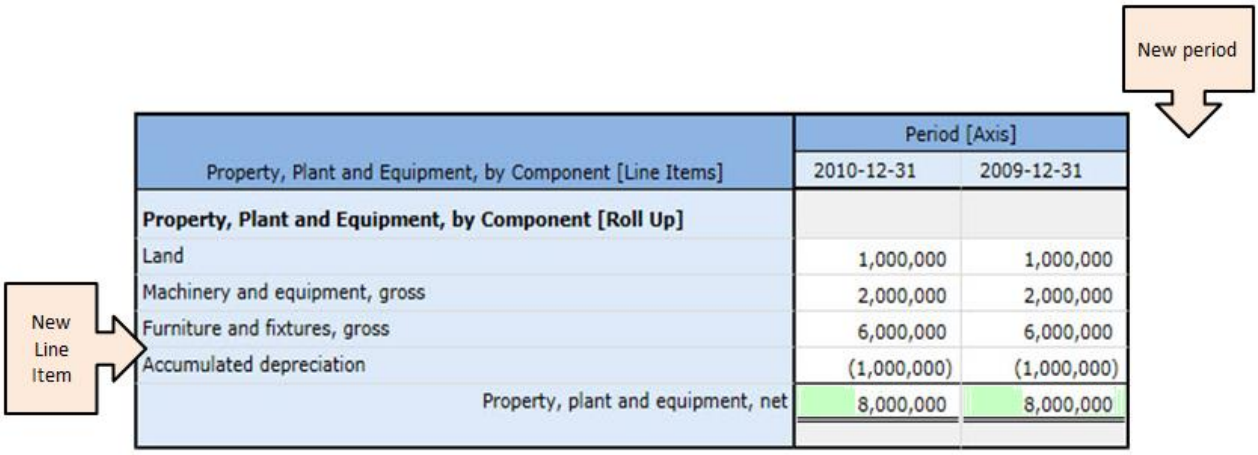
Information is entered into the report by (a) importing the information into the report or (b) manually entering information into the report.

Each view can be edited. Facts can be added to the Fact Table View. Associations can be added to the Model Structure View. Rules can be added to the Rules view. You don't ever edit the Verification Results View, but as more information is entered, that view gets updated. Report elements can be added to the Report Element View which then become available in the Model Structure View and Fact Table View.

Slots

Information Blocks have "slots". A Slot is simply a place in an information Block where it makes logical sense for new objects to be added to the Block. Different types of Blocks, different concept arrangement patterns[67] and member arrangement patterns[68], have different slots.

Below you can see one Block, showing two Slots for that block. One Slot is that a new Line Item can be added within the roll up total. Or, a second Slot is that a new period can be added to the Block.



This is not a comprehensive discussion of Blocks and Slots, it only provides the general ideas of what a Block is, what a Slot is, and that a financial report can be broken down into a set of Blocks each of which has specific Slots where information can be added.

Poka-yoke: Mistake proofing software

Poka-yoke[69] is a Lean Six Sigma technique used to prevent mistakes through smarter design. Poka-yoke is a Japanese term that means "mistake-proofing". A poka-yoke is any mechanism consciously added to a process that helps an equipment operator avoid mistakes. Its purpose is to eliminate defects by preventing, correcting, or drawing attention to human errors as the errors occur.

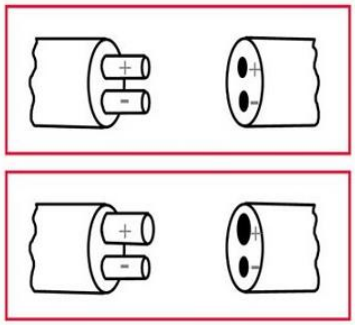
---

[67] Concept Arrangement Patterns, http://www.xbrlsite.com/mastering/Part02_Chapter05.I_ConceptArrangementPatterns.pdf

[68] Member Arrangement Patterns, http://www.xbrlsite.com/mastering/Part02_Chapter05.J_MemberArrangementPatterns.pdf

[69] Wikipedia, *Poka-yoke*, https://en.wikipedia.org/wiki/Poka-yoke

For example, consider the graphic[70] below. You want someone to plug the plug into the receptacle such that positive and negative match up; inadvertently reversing this would have catastrophic consequences. In the top graphic notice that it is possible to make a mistake but in the bottom a mistake would be impossible because of the size differences in the positive and negative receptacle and plug.



Smart design means less user errors. Poka-yoke techniques can be used to create software that is easier to use, can eliminate certain types of user mistakes, and can help guide users to put the Lego pieces together to get what they want.

Logical Glue

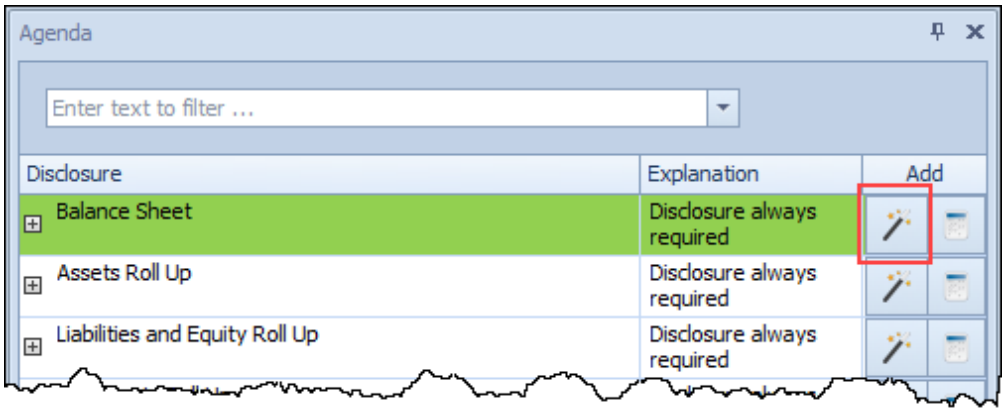
Each of the view are "glued together" logically. Business report model information is hard coded into the software application. The things that can go into the business report, the stuff that is provided within the report model (i.e. XBRL taxonomy, XBRL Formulas), must fit into that hard coded business report model. Each report model MUST be consistent with the report meta-model that is prescribed by the financial reporting scheme.

In our example we are using the PROOF financial reporting scheme meta-model[71]. Each financial reporting scheme[72] meta-model fits into the same business report model.

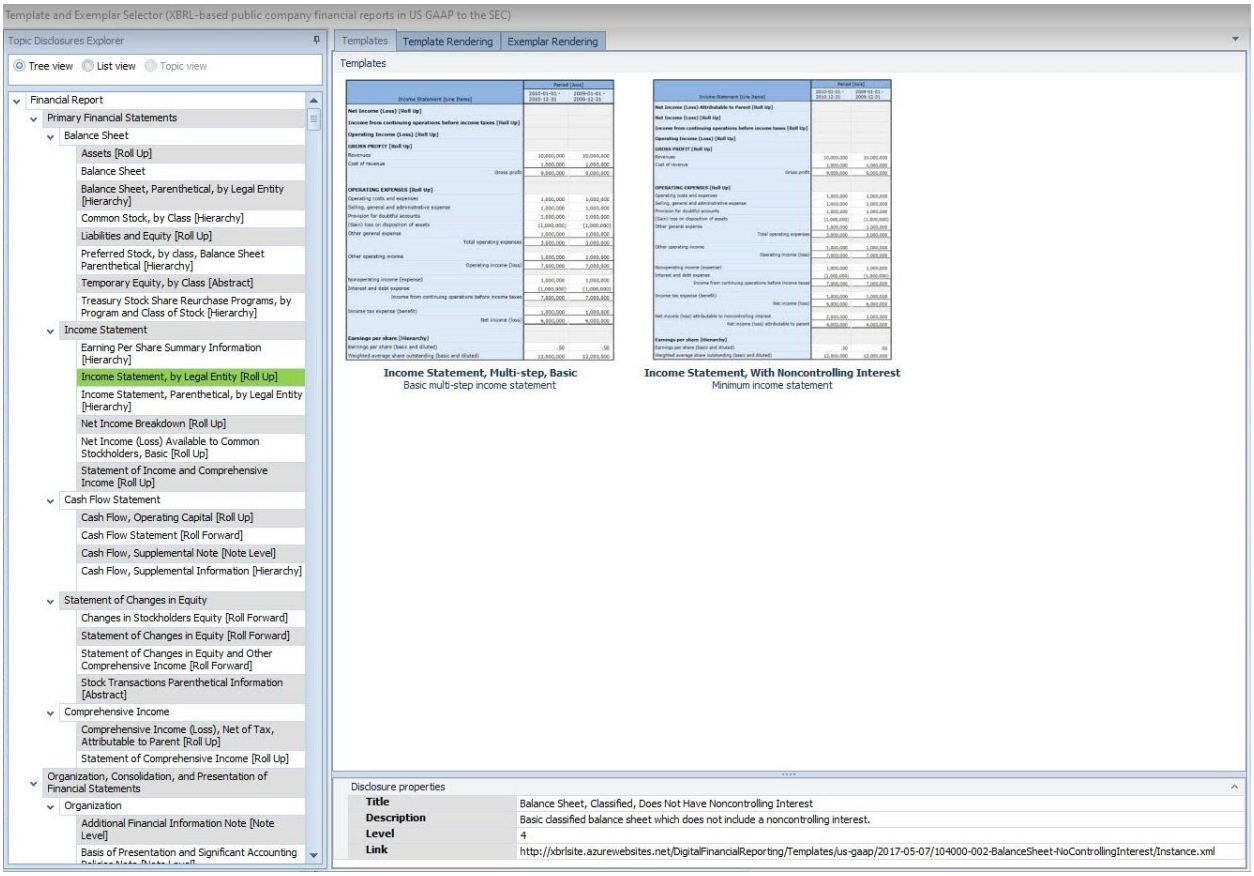
Templates

Templates are complete prototypes that are valid for a Disclosure. Templates can be imported. On the Agenda, you see a button for each Disclosure that is supposed to be in a report but does not yet exist:

[70] Process Exam, *Six Sigma Tools - Poka Yoke*, http://www.processexam.com/six-sigma-tools-poka-yoke
[71] PROOF Financial Reporting Scheme, http://www.xbrlsite.com/seattlemethod/proof/documentation/Index.html
[72] Financial Reporting Scheme Meta-Models, http://accounting.auditchain.finance/reporting-scheme/index.html



When you press that button, a set of templates appears and the templates for the disclosure that you are working with is selected and each template is shown. Below you see a prototype set of templates for US GAAP Disclosures[73]:



When you click on a template to select it for import, the template, which is represented in global standard XBRL, is imported, the expert system application makes any necessary adjustments to the template to make sure namespaces and prefixes match, and then the user can edit the template as the deem appropriate.

[73] Template selector, https://photos.google.com/share/AF1QipOfV4MNsTK3tekpIcb6HraFJ9WLIJH-UzyQp_CwlVQVt-45mHyhSpgNRCI7QOhoUw/photo/AF1QipOFYn1d9aqoSGBUWEYNo5OzoaTvhph604xzp-OB?key=aTZTUWQ1VDdVaVNldHotbkdoRGxNQVlQWkoxdHNn

Exemplars

The entire SEC EDGAR system is a repository of disclosure exemplars (a.k.a. examples). From a financial reporting logic perspective, the exemplars are generally excellent. From an XBRL point of view probably 80% are very good to excellent and 20% should be ignored. The quality of the exemplars will improve over time.

This is similarly true for the ESMA system; that too is a repository of disclosure exemplars that will improve over time.

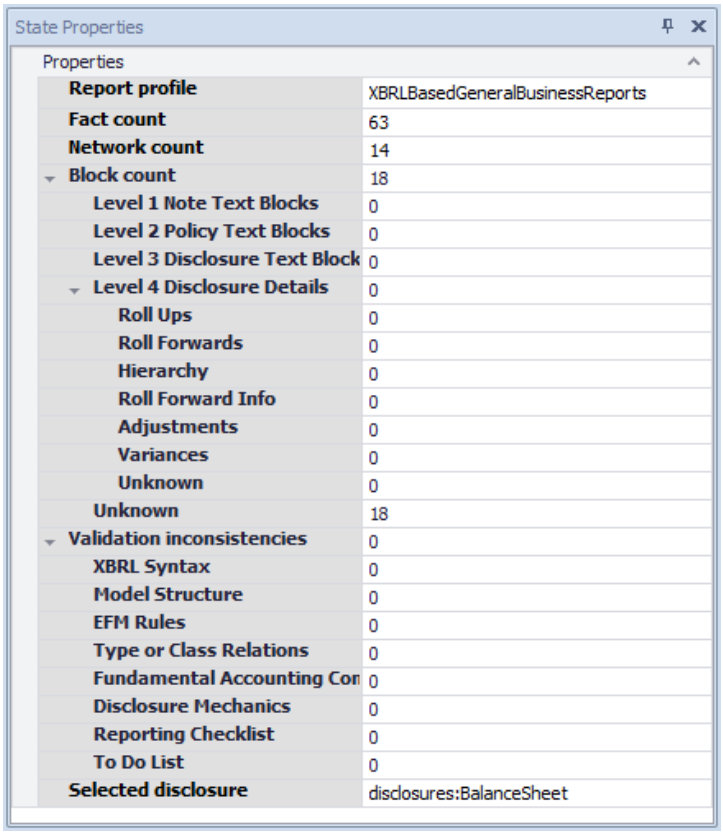
Those reports will be sliced and diced by industry, by entity type, and by other useful classifications so that you don't need to sift through thousands of reports, a recommender system will help you find and get the example you are looking for.

Similar to templates, exemplars will be imported directly into your report where it can be modified to meet your specific needs. Think "copy and paste" on steroids.

The metadata that will help you find the exemplar that you might be looking for will be created by accountants that contribute to the system which the expert system uses to drive its functionality.

State Machine

The expert system for creating financial reports acts more like a dynamic video game than a batch process. The expert system watches over what the user(s) are doing within the system and has access to information about the "state" of the application, a state machine[74].



[74] Wikipedia, *Finite State Machine*, https://en.wikipedia.org/wiki/Finite-state_machine

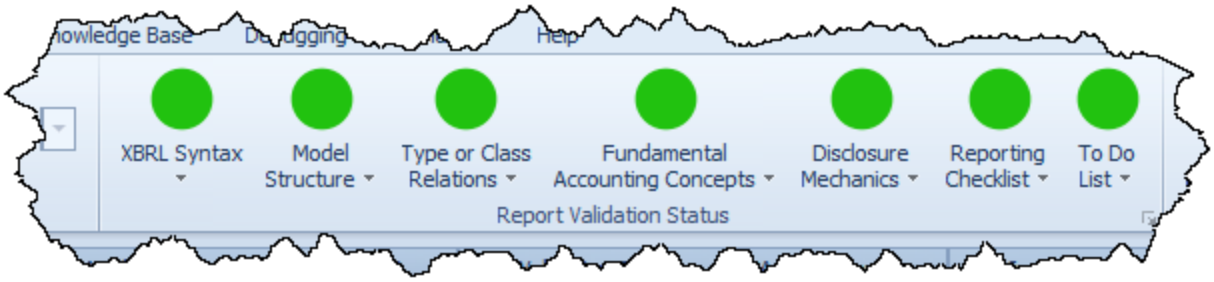
A good way to understand why a state machine is useful is by watching the video, State Machine Basics[75], that I put up on YouTube.com.

The state machine information is available to recommender systems, project management information, workflow management, and so forth.

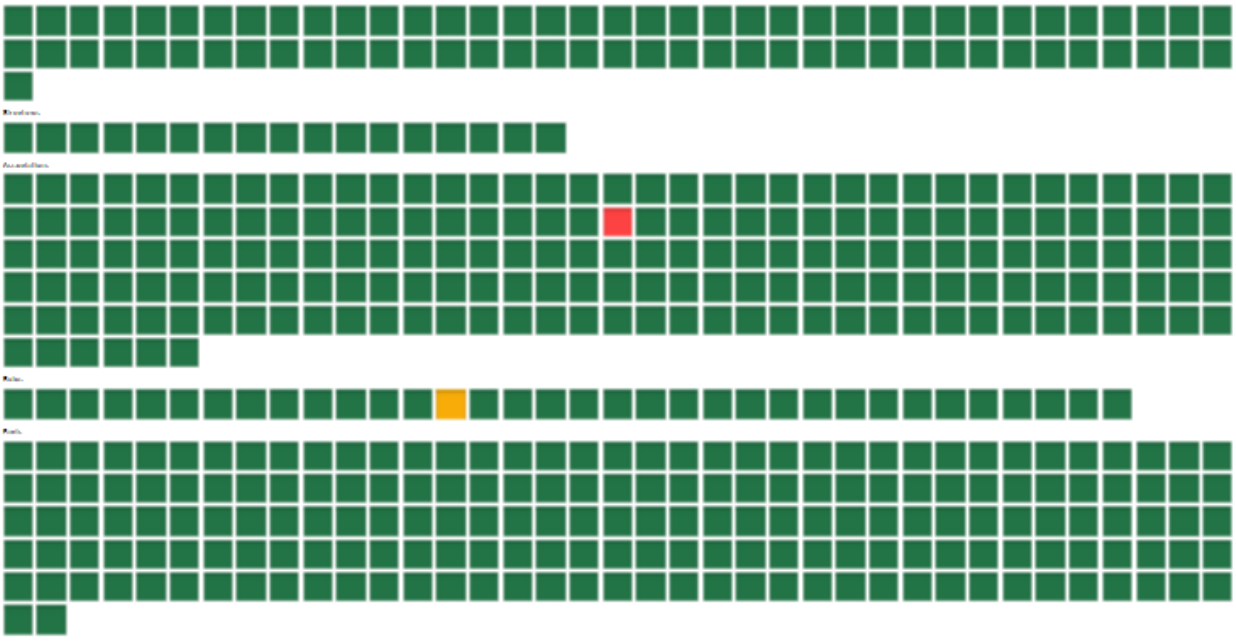
Project Management Information

The construction of a financial report is a project. If you have a team of people working on that report it becomes more complicated to manage that project. Project management is built into the expert system for creating financial reports. This includes "to do" lists that are manually maintained by the financial report creation team.

The project management information rolls up to a dashboard where the team working on the project can get a view of the big picture, for example this verification summary dashboard:



Cleverly designed interfaces[76] will help manage the big picture, the details, and every step in between:



[75] State Machine Basics, https://youtu.be/NuFGJHJ3jOM

[76] Example interface, http://xbrlsite.azurewebsites.net/2021/prototypes/verification/VerificationResult.html

Model Structure

A report model consists of the following categories of report elements per the Logical Theory Describing Financial Report: Networks, Hypercubes, Dimensions, Members, LineItems, Abstracts, and Concepts.

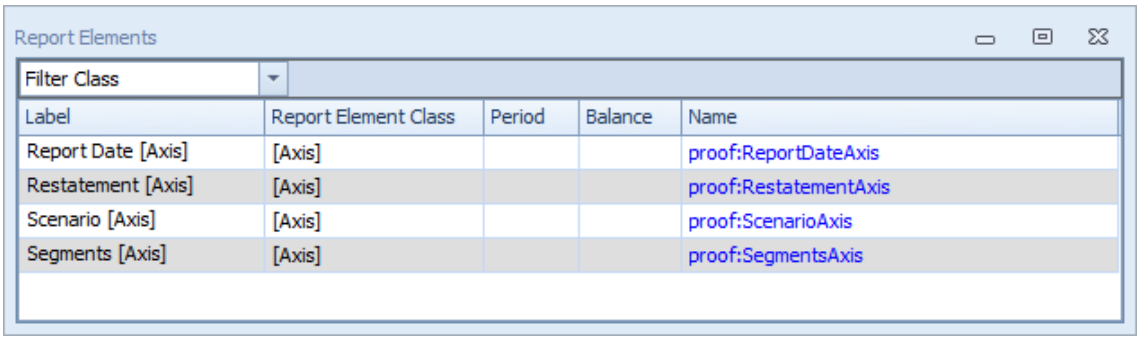
These report element categories have permitted and unpermitted associations with other categories of report elements. These permitted and unpermitted associations are defined in XBRL definition relations[77]. That information is presented in human readable form as follows:

		Parent						
		Network	Table	Axis	Member	Line Items	Abstract	Concept
Child	Network	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL
	Table	OK	Disallowed	Disallowed	Disallowed	Disallowed	OK	Disallowed
	Axis	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Member	Disallowed	Disallowed	OK	OK	Disallowed	Disallowed	Disallowed
	Line Items	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Abstract	OK	Disallowed	Disallowed	Disallowed	OK	OK	Disallowed
	Concept	Disallowed	Disallowed	Disallowed	Disallowed	OK	OK	Disallowed

An XBRL-based financial report can be valuated to be consistent or inconsistent with these model structure rules. Here is that evaluation for the PROOF report model structure as performed by Pesseract:

Child	Parent						
	Network	Table	Axis	Member	LineItems	Abstract	Concept
[Network]	0	0	0	0	0	0	0
[Table]	14	0	0	0	0	0	0
[Axis]	0	4	0	0	0	0	0
[Member]	0	0	4	9	0	0	0
[LineItems]	0	14	0	0	0	0	0
[Abstract]	0	0	0	0	15	2	0
[Concept]	0	0	0	0	3	55	0

When you click on any value you see the associations per the report model structure for that report:



Here is the same information provided by Pacioli[78]:

[77] Model Structure defined as XBRL definition relations, http://www.xbrlsite.com/seattlemethod/proof/model-structure/ModelStructure-rules-strict-def.xml

[78] Pacioli Model Structure Verification results, https://bafybeidv6e3zxuizutg5y4bo5eii5fwcjkdgzm6ht2kwk7xzbsqwz2mfdy.ipfs.infura-ipfs.io/modelStructure.html

Child	Parent						
	Network	Hypercube (a.k.a. Table)	Dimension (a.k.a. Axis)	Member	Line Items (a.k.a. Primary Items)	Abstract	Concept
Network	0	0	0	0	0	0	0
Hypercube (a.k.a. Table)	14	0	0	0	0	0	0
Dimension (a.k.a. Axis)	0	4	0	0	0	0	0
Member	0	0	4	9	0	0	0
Line Items (a.k.a. Primary Items)	0	14	0	0	0	0	0
Abstract	0	0	0	0	15	2	0
Concept	0	0	0	0	3	55	0

Note that the following verification script was used to generate this models structure result to make the result comparable to Pesseract. In the PROOF results with all of the validation tasks run, Pacioli considers all the associations in all the verification rules in addition to the report model structure, therefore different relationship counts result:

% Proof, only model structure verification, OK %
checkReport3("http://www.xbrlsite.com/seattlemethod/proof/reference-implementation/instance.xml",
['http://www.xbrlsite.com/seattlemethod/proof/model-structure/ModelStructure-rules-strict-def.xml'],
[saveToIPFS, extendedJSON, cacheValidity(3600)], Result, IPFSlink).

Note that the permitted and unpermitted model structure relations are configurable for an individual report, for an entire reporting scheme, or for all XBRL-based reports.

Model structure information is helpful to the expert system for creating reports, for locating XBRL taxonomy report elements, recommender systems, etc.

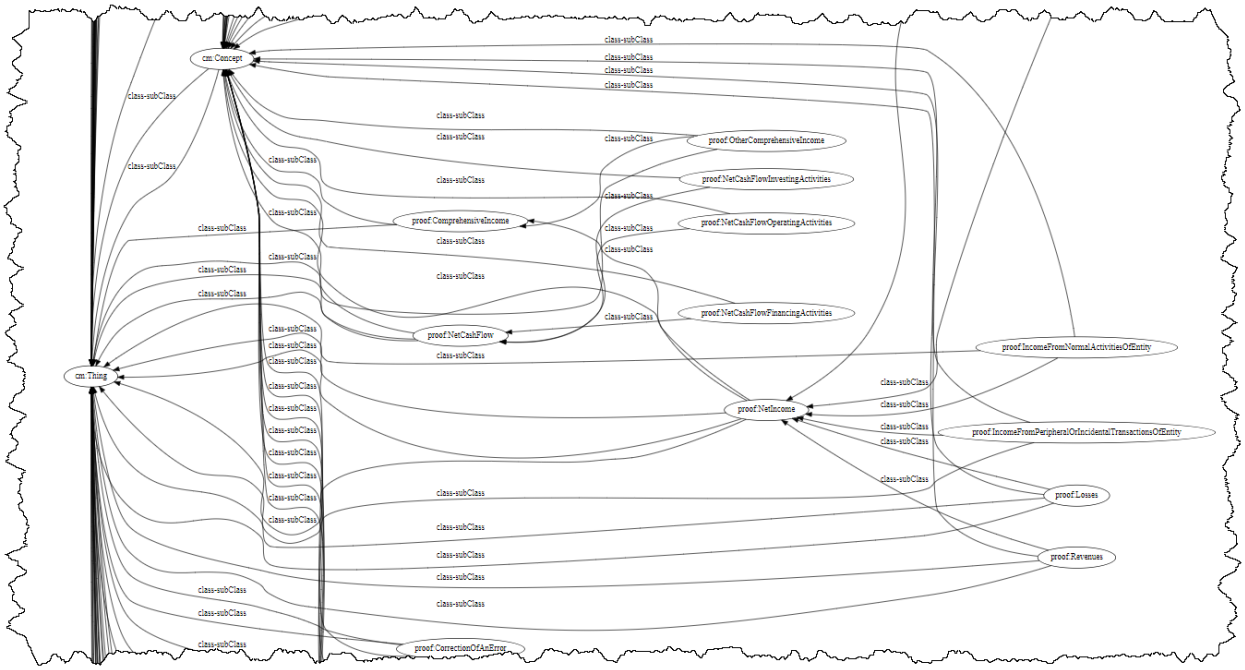
Note that the following table shows actual relations from the 2015 10-K financial reports of 6,000 public companies that submit their XBRL-based financial reports to the SEC. Notice the patters.

		Network	Table	Axis	Member	LineItems	Abstract	Concept
		485,842	212,618	430,552	1,360,484	212,733	727,374	3,155,641
Child	Network	0	0	0	0	0	0	0
	Table	513	0	0	4	4	212,090	11
	Axis	0	430,549	0	0	0	3	0
	Member	0	0	503,078	857,390	3	13	0
	LineItems	29	212,570	0	0	30	104	0
	Abstract	483,334	18	0	2	101,932	141,774	314
	Concept	8	0	1	49	1,178,684	1,969,653	7,246

Type-Subtype Associations

Type-subtype associations, a.k.a. “wider-narrower” or “general-special” or other types of associations help accountants using report models understand the associations between such report elements. Here is an example of the associations as represented in Pacioli[79]:

[79] Pacioli Type-subtype associations, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/typeSubTypeGraph.html



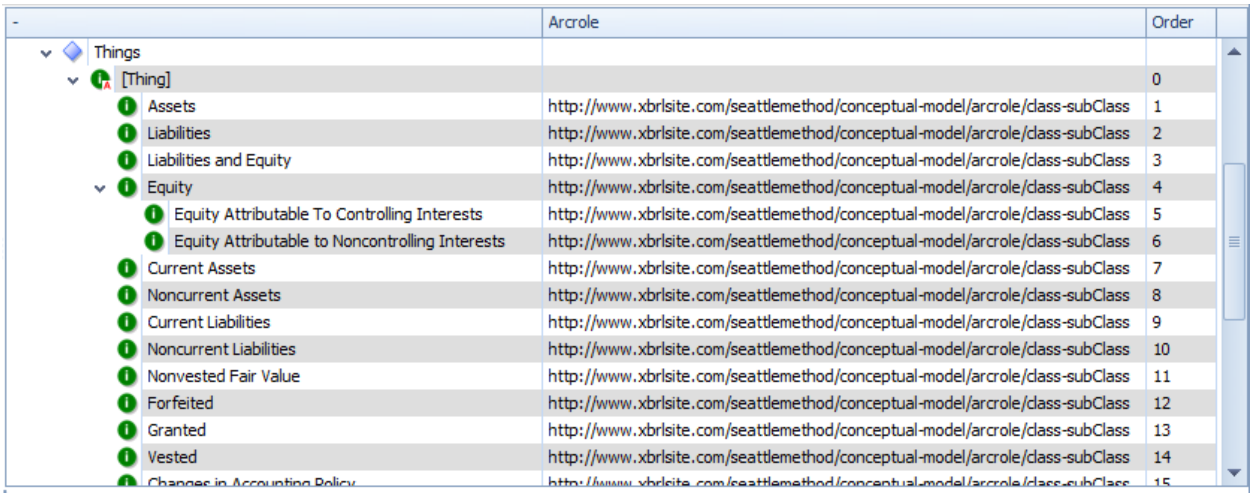
There are other approaches to representing this information for a user. As can be seen from the example above, the representations can become quite complex. But filtering by the type of association or other such things makes this information easier for a human to understand and digest.

An alternative representation is a table[80]:

#	Network	Type	Subtype
		cm:Concept	
18	Concept	cm:Concept	proof:Assets
19	Concept	cm:Concept	proof:BasisOfReportingTextBlock
20	Concept	cm:Concept	proof:ChangesInAccountingPolicy
21	Concept	cm:Concept	proof:ComprehensiveIncome
22	Concept	cm:Concept	proof:CorrectionOfAnError
23	Concept	cm:Concept	proof:CurrentAssets
24	Concept	cm:Concept	proof:CurrentLiabilities
25	Concept	cm:Concept	proof:DistributionsToOwners
26	Concept	cm:Concept	proof:Equity
27	Concept	cm:Concept	proof:EquityAttributableToControllingInterests
28	Concept	cm:Concept	proof:EquityAttributableToNoncontrollingInterests

Yet another view is a tree view as is shown below which comes from Pesseract:

[80] Pacioli Type-subtype associations, filterable table, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/typeSubTypeList.html



-	Arcrole	Order
Things		
[Thing]		0
Assets	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	1
Liabilities	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	2
Liabilities and Equity	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	3
Equity	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	4
Equity Attributable To Controlling Interests	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	5
Equity Attributable to Noncontrolling Interests	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	6
Current Assets	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	7
Noncurrent Assets	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	8
Current Liabilities	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	9
Noncurrent Liabilities	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	10
Nonvested Fair Value	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	11
Forfeited	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	12
Granted	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	13
Vested	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	14
Changes in Accounting Policy	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	15

Many other views are possible. This information and other information related to parts is useful in filtering down large XBRL taxonomies such as the US GAAP XBRL Taxonomy which has approximatley 17,000 report elements. Recommender systems that leverage the type-subtype and whole-part assocations can assist accountants using the expert system to do so effectively and effectently.

Fundamental Accounting Concepts Continuity Cross Checks

Financial reports are not forms. Rather, creators of financial reports are permitted to organize financial report information using different subtotals, different totals, completely different concepts based on which industry they are involved with, and certain common subtotals might be omitted from their financial report for one reason or another. Additionally, because of the way XBRL taxonomies are designed different reporting entities might use different XBRL taxonomy concepts to report the exact same line item.

Because of this flexibility the fundamental accounting concepts continuity cross checks are helpful, in fact essential, to sort financial reports out and figure out if the reports have been created correctly.

Data aggregators and others attempting to use reported information can use this same information to effectively extract information from financial reports.

The PROOF report does not provide insight into the extent that fundamental accounting concepts continuity cross checks are helpful so we will use a US GAAP example to show this. Below you see a part of the fundamental accounting concepts summary provided by XBRL Cloud for Microsoft[81]:

[81] Fundamental accounting concepts, Microsoft, http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/evidence-package/USFACRenderingSummary.html

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]] **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
Assets	241,086,000,000	fac:Assets[us-gaap:Assets[241,086,000,000]]
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	64,527,000,000	fac:CurrentLiabilities[us-gaap:LiabilitiesCurrent[64,527,000,000]]
Noncurrent Liabilities	104,165,000,000	fac:NoncurrentLiabilities[104,165,000,000] = fac:Liabilities[us-gaap:Liabilities[168,692,000,000]] - fac:CurrentLiabilities[us-gaap:LiabilitiesCurrent[64,527,000,000]]
Liabilities	168,692,000,000	fac:Liabilities[us-gaap:Liabilities[168,692,000,000]]
Commitments and Contingencies	0	fac:CommitmentsAndContingencies[us-gaap:CommitmentsAndContingencies[0]]
Temporary Equity	0	fac:TemporaryEquity[0] = fac:LiabilitiesAndEquity[us-gaap:LiabilitiesAndStockholdersEquity[241,086,000,000]] - ((fac:Liabilities[us-gaap:Liabilities[168,692,000,000]] + fac:Equity[72,394,000,000]) - fac:CommitmentsAndContingencies[us-gaap:CommitmentsAndContingencies[0]])
Equity [Roll Up]		
Equity Attributable to Parent	72,394,000,000	fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]]
Equity Attributable to Noncontrolling Interest	0	fac:EquityAttributableToNoncontrollingInterest[0] = fac:Equity[72,394,000,000] - fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]]
Equity	72,394,000,000	fac:Equity[72,394,000,000] = fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]]
Liabilities and Equity	241,086,000,000	fac:LiabilitiesAndEquity[us-gaap:LiabilitiesAndStockholdersEquity[241,086,000,000]]

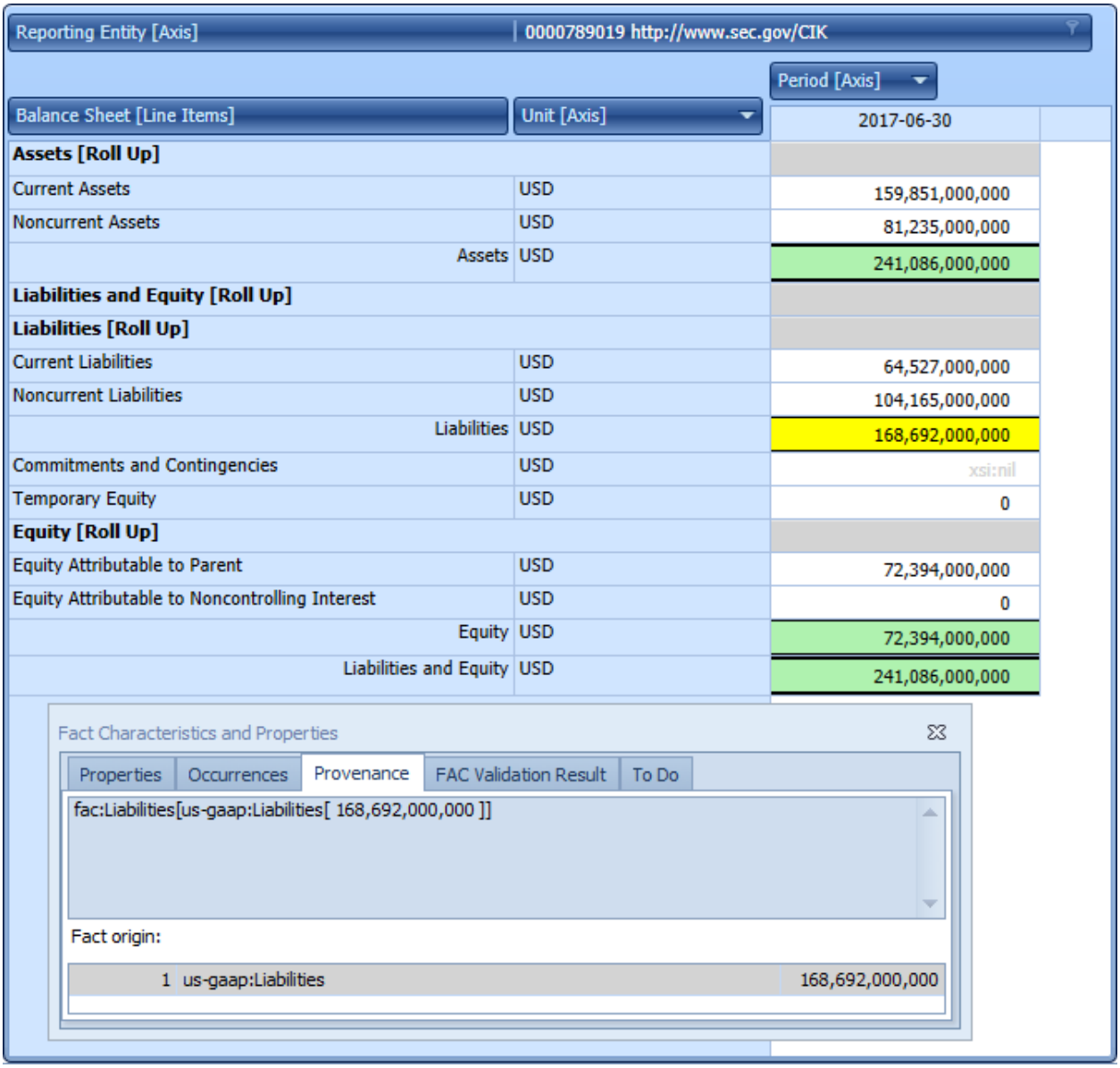
Pacioli provides similar information[82]:

Concept \ Period	2020-12-31	2019-12-31
Assets [Roll Up]		
Current Assets	500	0
Noncurrent Assets	3,000	0
Assets	3,500	0
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	0	0
Noncurrent Liabilities	0	0
Liabilities	0	0
Equity [Roll Up]		
Equity Attributable to Controlling Interests	3,000	0
Equity Attributable to Noncontrolling Interests	500	0
Equity	3,500	0
Liabilities and Equity	3,500	0

Pesseract provides another alternative approach which is more summarized:

Entity	Period	ID	Test	Result	Amount Of Inconsistency	Evaluation
0000789019	2017-FY	FAC_CONSISTENCY_1	fac:Equity = (fac:EquityAttributableToParent + fac:EquityAttributableToNoncontrollingInterest)	True	0	fac:Equity[72,394,000,000] = (fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]] + fac:EquityAttributableToNoncontrollingInterest[0])
0000789019	2017-FY	FAC_CONSISTENCY_10	fac:NetCashFlowFromInvestingActivities = (fac:NetCashFlowFromInvestingActivitiesContinuing + fac:NetCashFlowFromInvestingActivitiesDiscontinued)	True	0	fac:NetCashFlowFromInvestingActivities[(46,781,000,000)] = (fac:NetCashFlowFromInvestingActivitiesContinuing[us-gaap:NetCashProvidedByUsedInInvestingActivitiesContinuingOperations[(46,781,000,000)]] + fac:NetCashFlowFromInvestingActivitiesDiscontinued[0])
0000789019	2017-FY	FAC_CONSISTENCY_11	fac:NetCashFlowFromFinancingActivities = (fac:NetCashFlowFromFinancingActivitiesContinuing + fac:NetCashFlowFromFinancingActivitiesDiscontinued)	True	0	fac:NetCashFlowFromFinancingActivities[8,408,000,000] = (fac:NetCashFlowFromFinancingActivitiesContinuing[us-gaap:NetCashProvidedByUsedInFinancingActivitiesContinuingOperations[8,408,000,000]] + fac:NetCashFlowFromFinancingActivitiesDiscontinued[0])
0000789019	2017-FY	FAC_CONSISTENCY_12	fac:GrossProfit = (fac:Revenues - fac:CostOfRevenue)	True	0	fac:GrossProfit[us-gaap:GrossProfit[55,689,000,000]] = (fac:Revenues[us-gaap:SalesRevenueNet[89,950,000,000]] - fac:CostOfRevenue[us-gaap:CostOfRevenue[34,261,000,000]])
0000789019	2017-FY	FAC_CONSISTENCY_13	fac:OperatingIncomeLoss = (fac:GrossProfit - fac:OperatingExpenses)	True	0	fac:OperatingIncomeLoss[us-gaap:OperatingIncomeLoss[22,326,000,000]] = (fac:GrossProfit[us-gaap:GrossProfit[55,689,000,000]] - fac:OperatingExpenses[33,363,000,000])
0000789019	2017-FY	FAC_CONSISTENCY_15	fac:IncomeLossFromContinuingOperationsBeforeTax = (fac:OperatingIncomeLoss + fac:NonoperatingIncomePlusInterestAndDebtExpensePlusIncomeFromEquityMethodInvestments)	True	0	fac:IncomeLossFromContinuingOperationsBeforeTax[us-gaap:IncomeLossFromContinuingOperationsBeforeIncomeTaxesMinorityInterestAndIncomeLossFromEquityMethodInvestments[23,149,000,000]] = (fac:OperatingIncomeLoss[us-gaap:OperatingIncomeLoss[22,326,000,000]] + fac:NonoperatingIncomePlusInterestAndDebtExpensePlusIncomeFromEquityMethodInvestments[823,000,000])
0000789019	2017-FY	FAC_CONSISTENCY_16	fac:IncomeLossFromContinuingOperationsAfterTax = (fac:IncomeLossFromContinuingOperationsBeforeTax - fac:IncomeTaxExpenseBenefit)	True	0	fac:IncomeLossFromContinuingOperationsAfterTax[21,204,000,000] = (fac:IncomeLossFromContinuingOperationsBeforeTax[us-gaap:IncomeLossFromContinuingOperationsBeforeIncomeTaxesMinorityInterestAndIncomeLossFromEquityMethodInvestments[23,149,000,000]] - fac:IncomeTaxExpenseBenefit[us-gaap:IncomeTaxExpenseBenefit[1,945,000,000]])
0000789019	2017-FY	FAC_CONSISTENCY_17	fac:NetIncomeLoss = (fac:IncomeLossFromContinuingOperationsAfterTax + fac:IncomeLossFromDiscontinuedOperationsNetOfTax + fac:ExtraordinaryItemsOfIncomeExpenseNetOfTax)	True	0	fac:NetIncomeLoss[21,204,000,000] = (fac:IncomeLossFromContinuingOperationsAfterTax[21,204,000,000] + fac:IncomeLossFromDiscontinuedOperationsNetOfTax[0] + fac:ExtraordinaryItemsOfIncomeExpenseNetOfTax[0])
0000789019	2017-FY	FAC_CONSISTENCY_18	fac:NetIncomeLoss = (fac:NetIncomeLossAttributableToParent + fac:NetIncomeLossAttributableToNoncontrollingInterest)	True	0	fac:NetIncomeLoss[21,204,000,000] = (fac:NetIncomeLossAttributableToParent[us-gaap:NetIncomeLoss[21,204,000,000]] + fac:NetIncomeLossAttributableToNoncontrollingInterest[0])
0000789019	2017-FY	FAC_CONSISTENCY_19	fac:NetIncomeLossAvailableToCommonStockholdersBasic = (fac:NetIncomeLossAttributableToParent - fac:PreferredStockDividendsAndOtherAdjustments)	True	0	fac:NetIncomeLossAvailableToCommonStockholdersBasic[21,204,000,000] = (fac:NetIncomeLossAttributableToParent[us-gaap:NetIncomeLoss[21,204,000,000]] - fac:PreferredStockDividendsAndOtherAdjustments[0])
0000789019	2017-FY	FAC_CONSISTENCY_2	fac:Assets = fac:LiabilitiesAndEquity	True	0	fac:Assets[us-gaap:Assets[241,086,000,000]] = fac:LiabilitiesAndEquity[us-gaap:LiabilitiesAndStockholdersEquity[241,086,000,000]]
0000789019	2017-FY	FAC_CONSISTENCY_20	fac:ComprehensiveIncomeLoss = (fac:ComprehensiveIncomeLossAttributableToParent + fac:ComprehensiveIncomeLossAttributableToNoncontrollingInterest)	True	0	fac:ComprehensiveIncomeLoss[20,098,000,000] = (fac:ComprehensiveIncomeLossAttributableToParent[us-gaap:ComprehensiveIncomeNetOfTax[20,098,000,000]] + fac:ComprehensiveIncomeLossAttributableToNoncontrollingInterest[0])
0000789019	2017-FY	FAC_CONSISTENCY_21	fac:ComprehensiveIncomeLoss = (fac:NetIncomeLoss + fac:OtherComprehensiveIncomeLoss)	True	0	fac:ComprehensiveIncomeLoss[20,098,000,000] = (fac:NetIncomeLoss[21,204,000,000] + fac:OtherComprehensiveIncomeLoss[(1,106,000,000)])
0000789019	2017-FY	FAC_CONSISTENCY_3	fac:Assets = (fac:CurrentAssets + fac:NoncurrentAssets)	True	0	fac:Assets[us-gaap:Assets[241,086,000,000]] = (fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]] + fac:NoncurrentAssets[81,235,000,000])
0000789019	2017-FY	FAC_CONSISTENCY_4	fac:Liabilities = (fac:CurrentLiabilities + fac:NoncurrentLiabilities)	True	0	fac:Liabilities[us-gaap:Liabilities[168,692,000,000]] = (fac:CurrentLiabilities[us-gaap:LiabilitiesCurrent[64,527,000,000]] + fac:NoncurrentLiabilities[104,165,000,000])
0000789019	2017-FY	FAC_CONSISTENCY_5	fac:LiabilitiesAndEquity = (fac:Liabilities + fac:CommitmentsAndContingencies + fac:TemporaryEquity + fac:Equity)	True	0	fac:LiabilitiesAndEquity[us-gaap:LiabilitiesAndStockholdersEquity[241,086,000,000]] = (fac:Liabilities[us-gaap:Liabilities[168,692,000,000]] + fac:CommitmentsAndContingencies[us-gaap:CommitmentsAndContingencies[ ]] + fac:TemporaryEquity[0] + fac:Equity[72,394,000,000])
0000789019	2017-FY	FAC_CONSISTENCY_50	fac:NetCashFlow = (fac:NetCashFlowContinuing + fac:NetCashFlowDiscontinued + fac:ExchangeGainsLosses)	True	0	fac:NetCashFlow[us-gaap:CashAndCashEquivalentsPeriodIncreaseDecrease[1,153,000,000]] = (fac:NetCashFlowContinuing[1,134,000,000] + fac:NetCashFlowDiscontinued[0] + fac:ExchangeGainsLosses[us-gaap:EffectOfExchangeRateOnCashAndCashEquivalents[19,000,000]])
0000789019	2017-FY	FAC_CONSISTENCY_6	fac:NetCashFlow = (fac:NetCashFlowFromOperatingActivities + fac:NetCashFlowFromInvestingActivities + fac:NetCashFlowFromFinancingActivities + fac:ExchangeGainsLosses)	True	0	fac:NetCashFlow[us-gaap:CashAndCashEquivalentsPeriodIncreaseDecrease[1,153,000,000]] = (fac:NetCashFlowFromOperatingActivities[39,507,000,000] + fac:NetCashFlowFromInvestingActivities[(46,781,000,000)] + fac:NetCashFlowFromFinancingActivities[8,408,000,000] + fac:ExchangeGainsLosses[us-gaap:EffectOfExchangeRateOnCashAndCashEquivalents[19,000,000]])
0000789019	2017-FY	FAC_CONSISTENCY_7	fac:NetCashFlowContinuing = (fac:NetCashFlowFromOperatingActivitiesContinuing + fac:NetCashFlowFromInvestingActivitiesContinuing + fac:NetCashFlowFromFinancingActivitiesContinuing)	True	0	fac:NetCashFlowContinuing[1,134,000,000] = (fac:NetCashFlowFromOperatingActivitiesContinuing[us-gaap:NetCashProvidedByUsedInOperatingActivitiesContinuingOperations[39,507,000,000]] + fac:NetCashFlowFromInvestingActivitiesContinuing[us-gaap:NetCashProvidedByUsedInInvestingActivitiesContinuingOperations[(46,781,000,000)]] + fac:NetCashFlowFromFinancingActivitiesContinuing[us-gaap:NetCashProvidedByUsedInFinancingActivitiesContinuingOperations[8,408,000,000]])
0000789019	2017-FY	FAC_CONSISTENCY_8	fac:NetCashFlowDiscontinued = (fac:NetCashFlowFromOperatingActivitiesDiscontinued + fac:NetCashFlowFromInvestingActivitiesDiscontinued + fac:NetCashFlowFromFinancingActivitiesDiscontinued)	True	0	fac:NetCashFlowDiscontinued[0] = (fac:NetCashFlowFromOperatingActivitiesDiscontinued[0] + fac:NetCashFlowFromInvestingActivitiesDiscontinued[0] + fac:NetCashFlowFromFinancingActivitiesDiscontinued[0])
0000789019	2017-FY	FAC_CONSISTENCY_9	fac:NetCashFlowFromOperatingActivities = (fac:NetCashFlowFromOperatingActivitiesContinuing + fac:NetCashFlowFromOperatingActivitiesDiscontinued)	True	0	fac:NetCashFlowFromOperatingActivities[39,507,000,000] = (fac:NetCashFlowFromOperatingActivitiesContinuing[us-gaap:NetCashProvidedByUsedInOperatingActivitiesContinuingOperations[39,507,000,000]] + fac:NetCashFlowFromOperatingActivitiesDiscontinued[0])

[82] Pacioli, Fundamental Accounting Concepts, https://bafybeicptuxhaahe7lt6sok7ibztzdtfkrymfmfijyzurbto67dgighx5a.ipfs.infura-ipfs.io/f84790a25ff59191a37a.html#38df70550bbd781f3247



Reporting Entity [Axis]	0000789019 http://www.sec.gov/CIK	
		Period [Axis]
Balance Sheet [Line Items]	Unit [Axis]	2017-06-30
Assets [Roll Up]		
Current Assets	USD	159,851,000,000
Noncurrent Assets	USD	81,235,000,000
Assets	USD	241,086,000,000
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	USD	64,527,000,000
Noncurrent Liabilities	USD	104,165,000,000
Liabilities	USD	168,692,000,000
Commitments and Contingencies	USD	xsi:nil
Temporary Equity	USD	0
Equity [Roll Up]		
Equity Attributable to Parent	USD	72,394,000,000
Equity Attributable to Noncontrolling Interest	USD	0
Equity	USD	72,394,000,000
Liabilities and Equity	USD	241,086,000,000

Normalized View of Report Information

Note that the fundamental accounting concept continuity cross check information provides an approach to normalizing and comparing information reported in XBRL-based financial reports. Period comparisons (same entity, different periods) and entity comparisons (different entity, same period) are easily generated using the mappings, derivation rules, fundamental accounting concepts, and consistency rules.

The following are a number of examples generated using Pesseract:

Period comparison:

Component: (Network and Table)					
Network	001 - Unknown - **General Information**				
Table	General Information [Table]				
Reporting Entity [Axis]	0000789019 http://www.sec.gov/CIK				
	Period [Axis]				
General Information [Line Items]	2016-07-01/2016-12-31	2016-07-01/2016-09-30	2015-07-01/2016-06-30	2015-07-01/2016-03-31	2015-07-01/2015-12-31
General Information [Hierarchy]					
Entity Registrant Name	MICROSOFT CORPORATION	MICROSOFT CORPORATION	MICROSOFT CORPORATION	MICROSOFT CORPORATION	MICROSOFT CORPORATION
Entity Central Index Key	0000789019	0000789019	0000789019	0000789019	0000789019
Entity Filer Category	Large Accelerated Filer	Large Accelerated Filer	Large Accelerated Filer	Large Accelerated Filer	Large Accelerated Filer
Trading Symbol	MSFT	MSFT	MSFT	MSFT	MSFT
Fiscal Year End	--06-30	--06-30	--06-30	--06-30	--06-30
Fiscal Year Focus	2017	2017	2016	2016	2016
Fiscal Period Focus	Q2	Q1	FY	Q3	Q2
Document Type	10-Q	10-Q	10-K	10-Q	10-Q
Balance Sheet Date	2016-12-31	2016-09-30	2016-06-30	2016-03-31	2015-12-31

Entity comparison:

Component: (Network and Table)					
Network	001 - Unknown - **General Information**				
Table	General Information [Table]				
Drop Filter Fields Here					
	Period [Axis]	Reporting Entity [Axis]			
	2016-07-31/2017-01-28	2016-09-25/2016-12-31	2016-07-01/2016-12-31	2016-04-01/2016-12-31	2015-11-28/2016-12-02
General Information [Line Items]	0000858877 http://www.sec.gov/CIK	0000320193 http://www.sec.gov/CIK	0000789019 http://www.sec.gov/CIK	0000880807 http://www.sec.gov/CIK	0000796343 http://www.sec.gov/CIK
General Information [Hierarchy]					
Entity Registrant Name	CISCO SYSTEMS, INC.	APPLE INC	MICROSOFT CORPORATION	AMERICAN SUPERCONDUC...	ADOBE SYSTEMS INC
Entity Central Index Key	0000858877	0000320193	0000789019	0000880807	0000796343
Entity Filer Category	Large Accelerated Filer	Large Accelerated Filer	Large Accelerated Filer	Accelerated Filer	Large Accelerated Filer
Trading Symbol	CSCO	AAPL	MSFT	AMSC	0
Fiscal Year End	--07-29	--09-30	--06-30	--03-31	--12-02
Fiscal Year Focus	2017	2017	2017	2016	2016
Fiscal Period Focus	Q2	Q1	Q2	Q3	FY
Document Type	10-Q	10-Q	10-Q	10-Q	10-K
Balance Sheet Date	2017-01-28	2016-12-31	2016-12-31	2016-12-31	2016-12-02

Entity comparison, normalized income statement:

Component: (Network and Table)					
Network	201.7 - Unknown - **Income Statement, Multi Step, With Operating Income, Special 6**				
Table	Income Statement, Single Step [Table]				
Drop Filter Fields Here					
	Period [Axis]	Reporting Entity [Axis]			
	2016-07-31/2017-01-28	2016-09-25/2016-12-31	2016-07-01/2016-12-31	2016-04-01/2016-12-31	2015-11-28/2016-12-02
Income Statement [Line Items]	0000858877 http://www.sec.gov/CIK	0000320193 http://www.sec.gov/CIK	0000789019 http://www.sec.gov/CIK	0000880807 http://www.sec.gov/CIK	0000796343 http://www.sec.gov/CIK
Net Income (Loss) [Roll Up]					
Income (Loss) from Continuing Operations After Tax [Roll Up]					
Income (Loss) from Continuing Operations Before Tax [Roll Up]					
Operating Income (Loss) [Roll Up]					
Gross Profit [Roll Up]					
Revenues	23,932,000,000	78,351,000,000	44,543,000,000	59,000,000	5,854,430,000
Cost of Revenue	8,772,000,000	48,175,000,000	17,745,000,000	50,992,000	819,908,000
Gross Profit	15,160,000,000	30,176,000,000	26,798,000,000	8,008,000	5,034,522,000
Operating Expenses	9,390,000,000	6,817,000,000	15,396,000,000	28,562,000	3,540,920,000
Operating Income (Loss)	5,770,000,000	23,359,000,000	11,402,000,000	(20,554,000)	1,493,602,000
Nonoperating Income (Loss) Plus Interest and Debt Expense Plus Income (Loss) from Equity Method Investments	146,000,000	821,000,000	286,000,000	1,142,000	(58,464,000)
Income (Loss) from Continuing Operations Before Tax	5,916,000,000	24,180,000,000	11,688,000,000	(19,412,000)	1,435,138,000
Income Tax Expense (Benefit)	1,246,000,000	6,289,000,000	1,798,000,000	1,036,000	266,356,000
Income (Loss) from Continuing Operations After Tax	4,670,000,000	17,891,000,000	9,890,000,000	(20,448,000)	1,168,782,000
Income (Loss) from Discontinued Operations, Net of Tax	0	0	0	0	0
Extraordinary Items of Income (Expense), Net of Tax	0	0	0	0	0
Net Income (Loss)	4,670,000,000	17,891,000,000	9,890,000,000	(20,448,000)	1,168,782,000

Entity comparison, normalized cash flow statement:

Component: (Network and Table)					
Network	401 - Unknown - **Cash Flow Statement**				
Table	Cash Flow Statement [Table]				
Drop Filter Fields Here	Period [Axis] / Reporting Entity [Axis]				
	2016-07-31/2017-01-28	2016-09-25/2016-12-31	2016-07-01/2016-12-31	2016-04-01/2016-12-31	2015-11-28/2016-12-02
Cash Flow Statement [Line Items]	0000858877 http://www.sec.gov/CIK	0000320193 http://www.sec.gov/CIK	0000789019 http://www.sec.gov/CIK	0000880807 http://www.sec.gov/CIK	0000796343 http://www.sec.gov/CIK
Net Cash Flow [Roll Up]					
Net Cash Flow from Operating Activities [Roll Up]					
Net Cash Flow from Operating Activities, Continuing	6,502,000,000	27,056,000,000	17,842,000,000	(10,535,000)	2,199,728,000
Net Cash Flow from Operating Activities, Discontinued	0	0	0	0	0
Net Cash Flow from Operating Activities	6,502,000,000	27,056,000,000	17,842,000,000	(10,535,000)	2,199,728,000
Net Cash Flow from Investing Activities [Roll Up]					
Net Cash Flow from Investing Activities, Continuing	(4,978,000,000)	(19,122,000,000)	(33,221,000,000)	357,000	(960,033,000)
Net Cash Flow from Investing Activities, Discontinued	0	0	0	0	0
Net Cash Flow from Investing Activities	(4,978,000,000)	(19,122,000,000)	(33,221,000,000)	357,000	(960,033,000)
Net Cash Flow from Financing Activities [Roll Up]					
Net Cash Flow from Financing Activities, Continuing	1,743,000,000	(12,047,000,000)	17,345,000,000	(3,657,000)	(1,090,706,000)
Net Cash Flow from Financing Activities, Discontinued	0	0	0	0	0
Net Cash Flow from Financing Activities	1,743,000,000	(12,047,000,000)	17,345,000,000	(3,657,000)	(1,090,706,000)
Exchange Gains (Losses)	0	0	(8,000,000)	(432,000)	(14,234,000)
Net Cash Flow	3,267,000,000	(4,113,000,000)	1,958,000,000	(14,267,000)	134,755,000

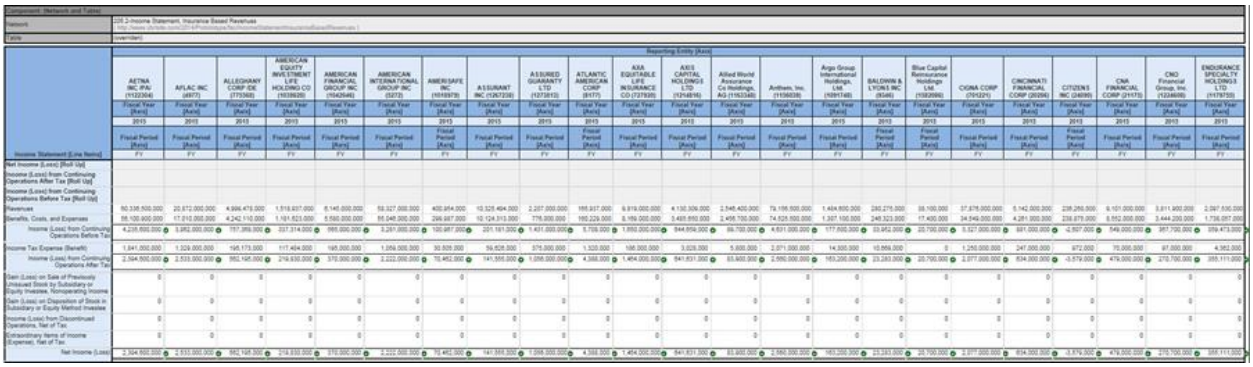
Entity comparison, normalized balance sheet:

Component: (Network and Table)					
Network	101 - Unknown - **Balance Sheet, Classified**				
Table	Balance Sheet, Classified [Table]				
Drop Filter Fields Here	Period [Axis] / Reporting Entity [Axis]				
	2017-01-28	2016-12-31			2016-12-02
Balance Sheet [Line Items]	0000858877 http://www.sec.gov/CIK	0000320193 http://www.sec.gov/CIK	0000789019 http://www.sec.gov/CIK	0000880807 http://www.sec.gov/CIK	0000796343 http://www.sec.gov/CIK
Assets [Roll Up]					
Current Assets	83,392,000,000	103,332,000,000	144,949,000,000	64,260,000	5,839,774,000
Noncurrent Assets	42,856,000,000	227,809,000,000	79,661,000,000	45,981,000	6,867,340,000
Assets	126,248,000,000	331,141,000,000	224,610,000,000	110,241,000	12,707,114,000
Liabilities and Equity [Roll Up]					
Liabilities [Roll Up]					
Current Liabilities	22,708,000,000	84,130,000,000	70,787,000,000	38,364,000	2,811,635,000
Noncurrent Liabilities	39,722,000,000	114,621,000,000	85,014,000,000	8,084,000	2,470,644,000
Liabilities	62,430,000,000	198,751,000,000	155,801,000,000	46,448,000	5,282,279,000
Commitments and Contingencies	xsi:nil	xsi:nil	xsi:nil	xsi:nil	xsi:nil
Temporary Equity	0	0	0	0	0
Equity [Roll Up]					
Equity Attributable to Parent	63,811,000,000	132,390,000,000	68,809,000,000	63,793,000	7,424,835,000
Equity Attributable to Noncontrolling Interest	7,000,000	0	0	0	0
Equity	63,818,000,000	132,390,000,000	68,809,000,000	63,793,000	7,424,835,000
Liabilities and Equity	126,248,000,000	331,141,000,000	224,610,000,000	110,241,000	12,707,114,000

The examples above only scratch the surface. Using the full set of machine readable rules, the expert system can compare disclosures, compare across financial reporting schemes by creating the correct mappings, compare information and pull the accounting policies information and show that with the quantitative information, provide "drill down", links between parts of an analysis.

Here is a comparison of 23 insurance companies[83]. This is achieved by simply providing the CIK number of the entity, the fiscal period desired, and the reporting style code. From that information, the following rendering is generated:

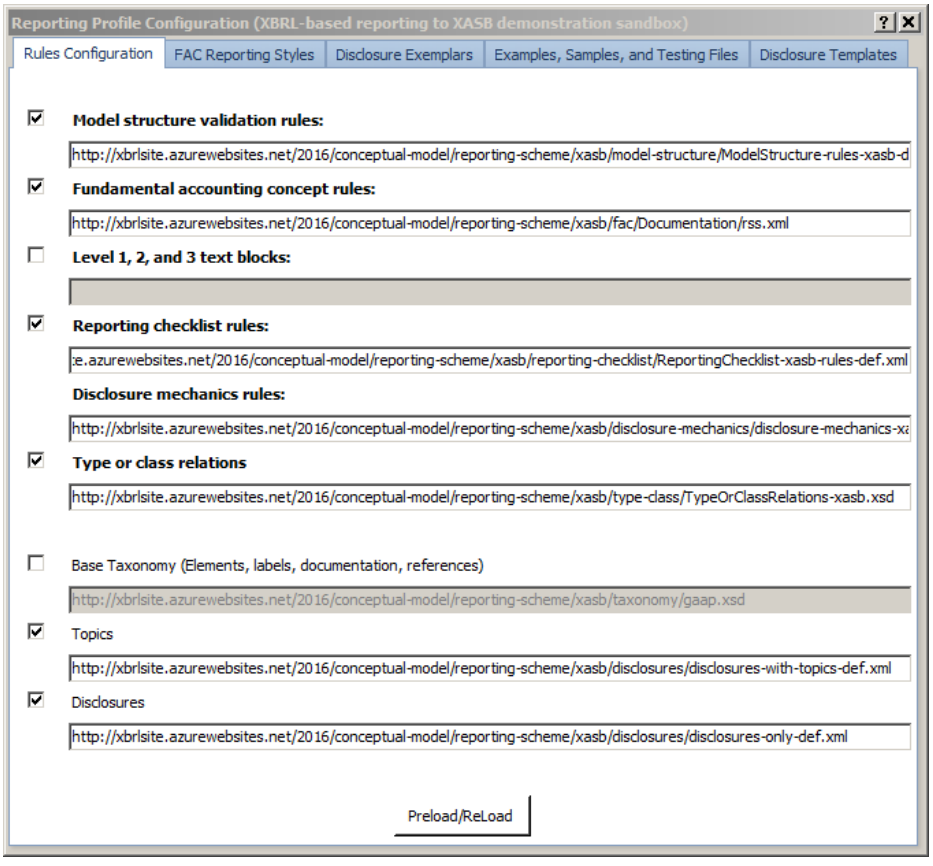
[83] Insurance company comparison, http://www.xbrlsite.com/2016/fac/v3/Examples/IncomeStatementInsuranceBasedRevenues.jpg



Additional normalized comparison examples[84].

Reporting Schemes

There are many different financial reporting schemes[85]. Each financial reporting scheme has a set of knowledge expressed in machine readable global standard XBRL for that specific financial reporting scheme[86]. An expert system for creating financial reports would connect to that machine readable information. Pesseract, for example, points to that machine readable information:



[84] Additional normalized comparison examples, http://www.xbrlsite.com/2016/fac/v3/Examples/Index.html

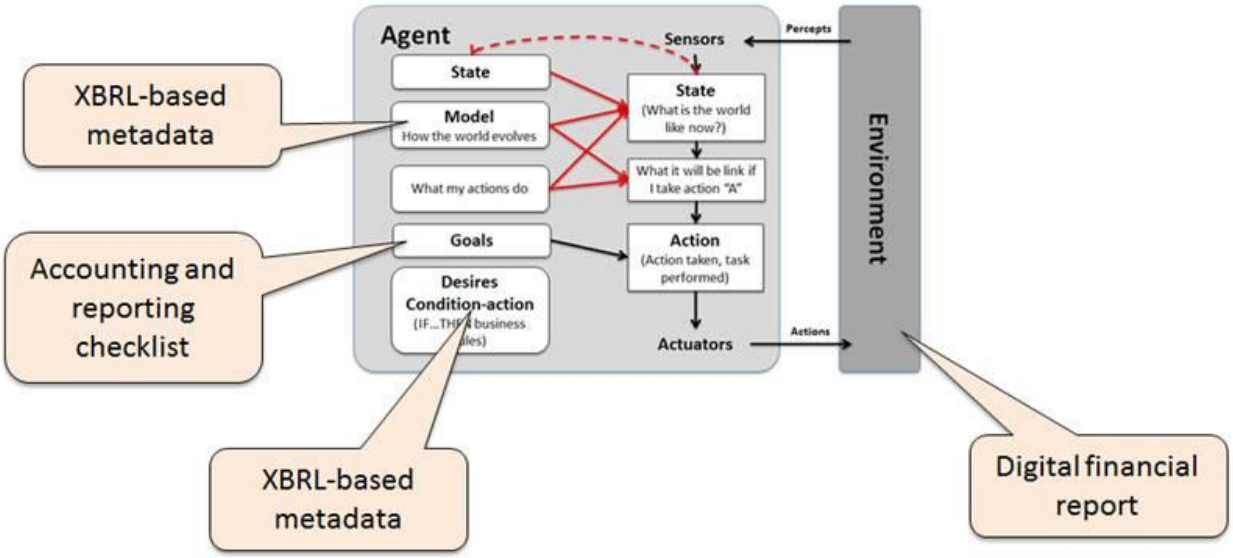
[85] *Comparison of Elements of Financial Statements*, http://xbrlsite.azurewebsites.net/2020/master/ElementsOfFinancialStatements.pdf

[86] *Financial Reporting Schemes XBRL-based Knowledge Graph*, http://accounting.auditchain.finance/reporting-scheme/index.html

Daemons and Intelligent Software Agents

A daemon[87] is a program that continuously runs in the background and wakes up to handle periodic service requests, which might come from some other process that needs something done.

An agent is an entity capable of sensing the state of its environment and acting upon it based on a set of specified rules. An agent performs specific tasks on behalf of another. In the case of software, an agent is a software program. Intelligent software agents[88] can work behind the scenes or in the foreground to perform tasks for users of the software.



Some examples of how daemons and intelligent software agents might help accountants in the creation of XBRL-based digital financial reports includes:

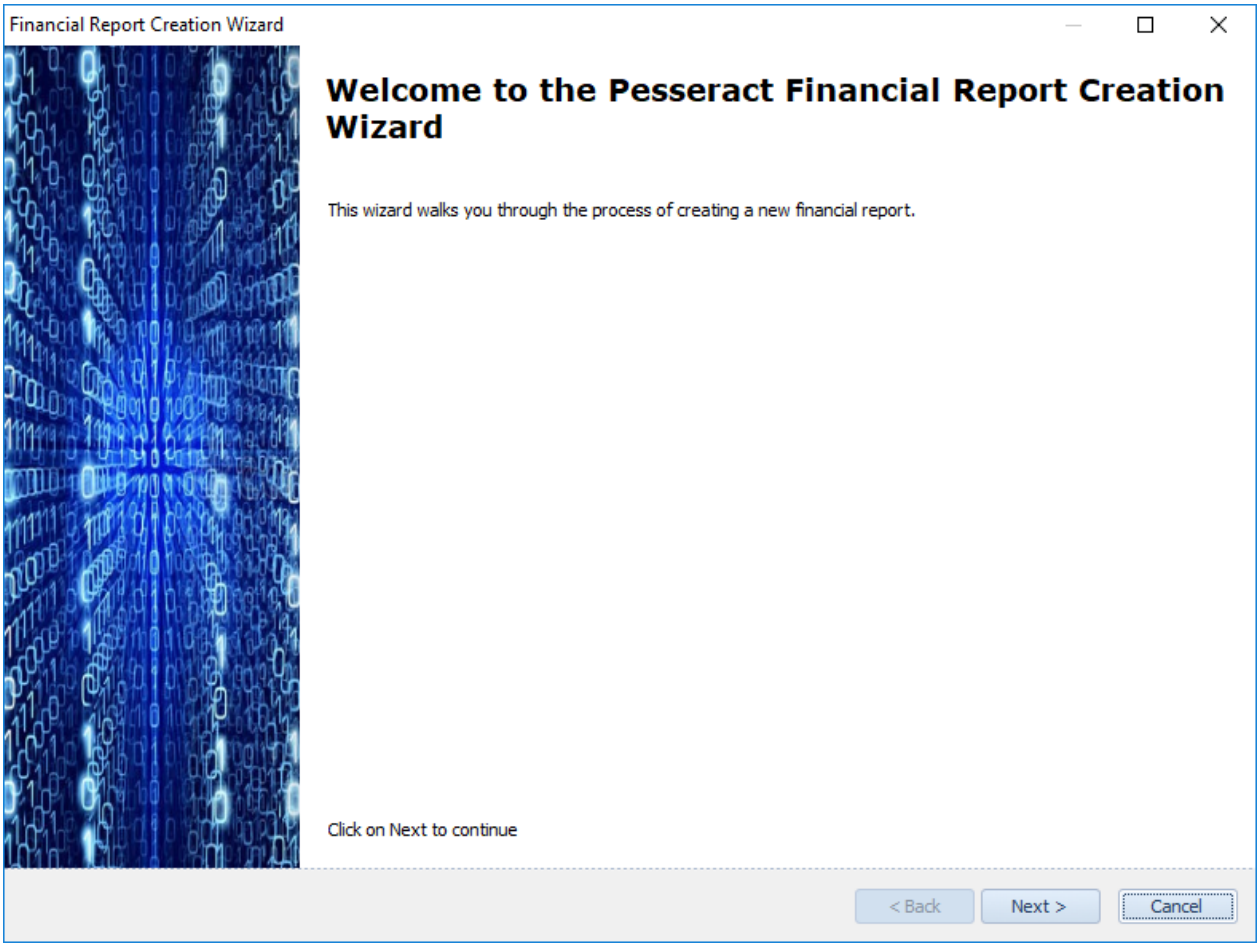
- Watching over the creation of the report model structure, assuring that report element associations stay within specified permitted boundaries.
- When a user is editing model structure the software senses where the accountant is editing the report and providing a list of the report elements to the user based on the context of where the user is currently editing.
- Leveraging type-subtype association rules to help users pick the correct concept to add to a model structure and to associate concepts within permitted rules.
- Verifying in the background that the mathematics of a roll up or roll forward and other such mathematical associations "foots" and "cross casts" correctly while a report is being created.
- Watching over a report and provide information to the user of software to indicate any inconsistencies or contradictions within a report to known fundamental accounting concept associations.

[87] TechTarget, *What is a daemon?*, https://www.techtarget.com/whatis/definition/daemon
[88] *Intelligent Software Agents*, http://www.xbrlsite.com/mastering/Part01_Chapter02.G_IntelligentSoftwareAgents.pdf

- Being sure that the logical and accounting relations within a disclosure are consistent with provided disclosure mechanics rules which specify permitted approaches to creating financial disclosures.

Wizards

A wizard is simply an approach to walking a user of an application through a series of steps. For example, Pesseract has a "Financial Report Creation Wizard" that walks the user through a series of steps that help them create a financial report:



A wizard is a type of recommender system. The wizard queries the user of the application for the following information:

- Which **financial reporting scheme** are they using to create the report? This will help provide the user with the correct machine readable information.
- What is the **legal form** of the reporting economic entity? This helps to application to understand what types of XBRL taxonomy concepts to use, which templates might be used, which disclosures apply, etc.
- What is the main **accounting activity** of the reporting economic entity? Again, this helps with the selection of machine readable metadata.
- What is the **industry sector** of the economic entity?

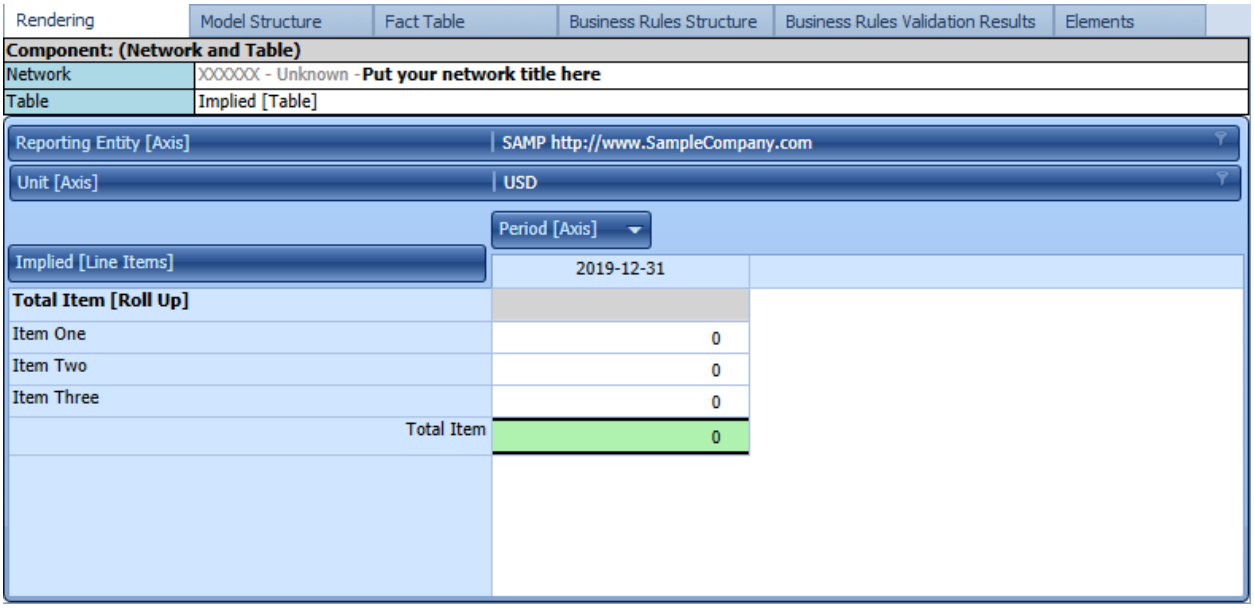
- What is the **reporting style** of the economic entity?
- Other report information.
- Other user preferences.
- Uses the collected information to suggest a list of **templates** the user can use to populate the report.

The answers to the questions are stored in the **state machine**. Then, the application uses that information to pre-filter machine readable knowledge, point the user in the right direction when performing a task, etc.

Jigs

A jig is somewhat like a mini template. The idea of jigs in the creation of XBRL-based reports was introduced in Pesseract[89]. What a jig does is it hides the technical complexity of constructing an information Block. Below you see the "jig" for the creation of a roll up. Rather than configuring the roll up; the user can simply import the jig and then modify the pieces of the jig in order to create a fragment of a financial report. A jig is simply a technique for making things easier.

Pesseract has jigs that can be used to create any information model.



Constructing a Report

Pulling all these pieces together to construct a report is summarized in the document *Financial Report Creation Proof*[90]. Fundamentally, creating a report involves (a) creating each individual fragment of the report properly and (b) making sure there are no contradictions or inconsistencies between report fragments. An agenda helps you understand what fragments are required, which have been completed, and provide templates for each fragment to assist in the fragment creation process[91]. The information for each fragment can be viewed in

[89] Pesseract, *Jigs*, http://xbrlsite.azurewebsites.net/2016/conceptual-model/jigs/rss.xml
[90] Financial Report Creation Proof, http://xbrlsite.com/2022/master-dynamic/FinancialReportCreationProof.pdf
[91] Report Creation, http://xbrlsite.com/seattlemethod/pictures/Index.html

different helpful ways[92]. Detailed lists of logical report artifacts are provided by software applications[93].

Logical Data Model

Key to creating a proper, elegant expert system for creating financial reports is the logical data model that is used to store data used by the expert system, enforce a logical scheme to keep the information quality high, and flexibly query and use that information within the interfaces of the expert system.

My background is with relational databases. As such in a prototype application that I created I used a Microsoft Access Database[94] to store data. The problem with that is that enforcing the logical schema and querying information is challenging.

Another software engineer used a Microsoft SQL Server database to create Luca which is a skeleton of an expert system for creating financial reports[95]. The point of Luca was to help the software engineer understand the logical data model that is necessary.

A third software engineer use a MongoDB database to create a cloud-based version of Luca which, likewise, is a skeleton of an expert system for creating financial reports[96].

This experimentation yielded a set of Excel spreadsheets that contained the logical data model that could be used to import information into Luca and cloud-based Luca. This Excel import spreadsheet set was completed when we created Pacioli. With Pacioli, you can import 100% of the PROOF into Pacioli which will then generate a complete XBRL-based financial report for the PROOF[97].

That same set of Excel files, excluding several rule formats not yet supported by the cloud-based Luca application (adjustment, variance, derivation, nonstandard), can be imported into both Pacioli and the cloud-based version of Luca. Very, very similar files can be copy and pasted into the local version of Luca and my Microsoft Access database application.

The point is that the Excel files represent the logical data model that would underly the expert system for creating financial reports.

Those Excel import files[98] are supplemented with an additional set of Excel files[99] that contain the machine readable information of the PROOF financial reporting scheme[100]. What is the point here? The point is that if you look at the additional supplemental files information, it all fits into the same logical model as the "skeleton" logical data model that is used to import the report model and report information.

---

[92] Information Block Views, http://xbrlsite.com/seattlemethod/pictures2/Index.html
[93] Detailed Lists (Logical), http://xbrlsite.com/seattlemethod/pictures3/Index.html
[94] *Free Open Source Tool for Creating Quality XBRL-based Digital Financial Reports*, http://xbrl.squarespace.com/journal/2020/12/8/free-open-source-tool-for-creating-quality-xbrl-based-digita.html
[95] Luca, http://xbrl.squarespace.com/journal/2020/9/15/luca.html
[96] Cloud-based Luca, http://xbrl.squarespace.com/journal/2021/8/31/cloud-based-luca.html
[97] Pacioli Import of Proof, http://xbrlsite.azurewebsites.net/2022/Library/PacioliImportOfProof.pdf
[98] Excel import files, http://accounting.auditchain.finance/library/proof-import-excel-2022-02-23.zip
[99] Excel import files supplement, http://accounting.auditchain.finance/library/proof-import-excel-2022-02-23_Supplement.zip
[100] Proof Financial Reporting Scheme Knowledge Graph, http://www.xbrlsite.com/seattlemethod/proof/documentation/index.html

The bottom line here is that the Luca and Pacioli import files show you the logical data model that is used to create the expert system for creating financial reports. There were a couple of short cuts that were taken related to the XBRL reference linkbase. But, that is essentially the logical data model.

But, the storage format for the data should be a graph or graph database of some sort with a logical schema enforcing the integrity of the information. While information can be effectively stored in a relational database, enforcing the logical rules is significantly more challenging when using a relational database.

To understand the logical data model of the expert system for creating financial reports it is suggested that you import the Excel[101] files for PROOF into Luca[102]. Any XBRL-based report submitted to the SEC could be imported using exactly the same technique and would fit into the logical data model.

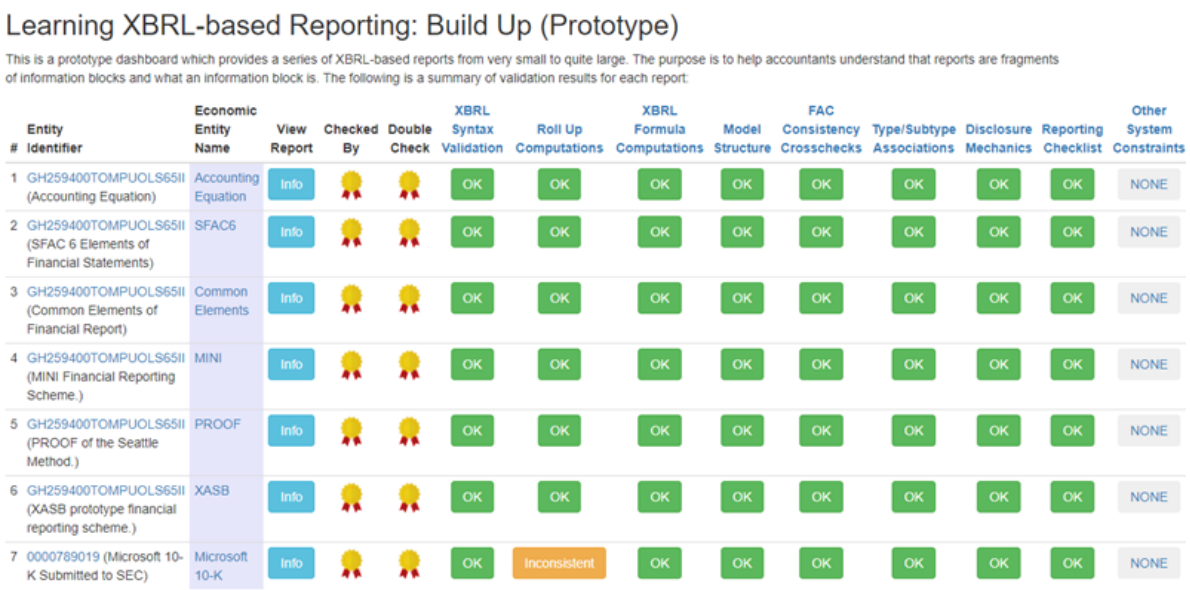
Additional work is necessary to figure out any modifications which might be necessary when the additional metadata for a financial reporting scheme[103] is added to the logical data model. While the metadata stored is likely not 100% complete, it does provide the information necessary to understand the moving parts of an expert system for creating financial reports. The PROOF tutorial for the locally installed version of Luca[104] is helpful in understanding the logical data model as a supplement to using cloud-based Luca tutorials[105] to understand the model.

The ”skeleton” applications, Luca, cloud-based Luca, and my Microsoft access database, are very helpful in understanding how to start creating an expert system for creating financial reports.

Examples

The following set of examples works up incrementally from a very tiny report to a complete 10-K of a public company submitted to the SEC[106]. The examples start with the accounting equation, to SFAC 6, to Common Elements of Financial Report, to a MINI financial reporting scheme, to the PROOF we have previously mentioned, to XASB which is a prototype financial reporting scheme, to US GAAP financial reporting by Microsoft in an 10-K.

[101] Excel files for Proof, http://accounting.auditchain.finance/library/proof-import-excel-2022-02-23.zip
[102] Luca, http://luca.yaxbrl.com/
[103] Additional supplemental metadata, http://accounting.auditchain.finance/library/proof-import-excel-2022-02-23_Supplement.zip
[104] Luca tutorials (locally installable version), http://xbrl.squarespace.com/journal/2020/9/15/luca.html
[105] Cloud-based Luca tutorials, http://xbrl.squarespace.com/journal/2021/8/31/cloud-based-luca.html
[106] Dashboard, Buildup, http://xbrlsite.azurewebsites.net/2022/Prototypes/buildup/Dashboard.html



Learning XBRL-based Reporting: Build Up (Prototype)

This is a prototype dashboard which provides a series of XBRL-based reports from very small to quite large. The purpose is to help accountants understand that reports are fragments of information blocks and what an information block is. The following is a summary of validation results for each report:

#	Entity Identifier	Economic Entity Name	View Report	Checked By	Double Check	XBRL Syntax Validation	Roll Up Computations	XBRL Formula Computations	Model Structure	FAC Consistency Crosschecks	Type/Subtype Associations	Disclosure Mechanics	Reporting Checklist	Other System Constraints
1	GH259400TOMPUOLS65II (Accounting Equation)	Accounting Equation	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE
2	GH259400TOMPUOLS65II (SFAC 6 Elements of Financial Statements)	SFAC6	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE
3	GH259400TOMPUOLS65II (Common Elements of Financial Report)	Common Elements	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE
4	GH259400TOMPUOLS65II (MINI Financial Reporting Scheme.)	MINI	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE
5	GH259400TOMPUOLS65II (PROOF of the Seattle Method.)	PROOF	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE
6	GH259400TOMPUOLS65II (XASB prototype financial reporting scheme.)	XASB	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE
7	0000789019 (Microsoft 10-K Submitted to SEC)	Microsoft 10-K	Info			OK	Inconsistent	OK	OK	OK	OK	OK	OK	NONE

Testing Resources

This ZIP archive[107] has a plethora of examples that can be used for testing.

This XBRL-based digital financial reporting conformance suite[108] helps you make sure you can detect many mistakes that are made when creating XBRL-based digital financial reports.

Learning About XBRL-based Digital Financial Reporting[109] has a multitude of reports that can be used for testing.

Mastering XBRL-based Digital Financial Reporting[110] is a series of examples.

My blog has a plethora[111] of documented examples that can be used for testing.

Library of Pesseract Screen Shots, Videos, Other

The library of Pesseract screen shots help professional accountants and software engineers building an expert system for constructing financial reports to communicate and discuss details[112]. The following video playlists help you see Pesseract in operation[113]. Other helpful information can be found on my blog post *Try Pesseract, Get a Glimpse of the Future of Financial Reporting*[114].

---

[107] Pacioli Examples, http://xbrlsite.azurewebsites.net/2022/Library/PacioliExamples.zip
[108] XBRL-based digital financial report conformance suite, http://xbrlsite.com/2020/conformance-suite/index.xml
[109] *Learning About XBRL-based Digital Financial Reporting*, http://xbrl.squarespace.com/journal/2022/2/9/learning-xbrl-based-digital-financial-reporting.html
[110] *Mastering XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2020/master/
[111] My blog, http://xbrl.squarespace.com/blog-archive/
[112] Pesseract, Screenshots, https://photos.app.goo.gl/cWeZYaMBEbmSSm7v8
[113] Pesseract, Video Playlists, https://www.youtube.com/channel/UCRIbipm3f0DaGPuLK51rvHA/playlists
[114] *Try Pesseract, Get a Glimpse of the Future of Financial Reporting*, http://xbrl.squarespace.com/journal/2020/4/3/try-pesseract-get-a-glimpse-of-the-future-of-financial-repor.html

Phases of Financial Report Creation Capabilities

There are several phases of financial report creation. This section walks you through those phases.

Phase 1: The objective of phase 1 is to understand the logical model of a digital financial report correctly and to have basic capabilities to create reports. The application at phase 1 can be used to create reports manually by keying information into a very basic interface similar to Luca[115], import 100% of report information using Excel import spreadsheets, and use an API to populate the software application. Once the information is in the application, XBRL is generated for the information within the application. In phase 1, information is entered directly into a form that looks very much like a database table. A high priority of this phase is getting information entry verification dialed in correctly.

StructureType	NetworkIdentifier	AssociationFromName	AssociationRole	AssociationToName	CalculationPolarity	PreferredLabelRole	Sequence
Presentation	BalanceSheet	proof:BalanceSheetHypercube	Parent-Child	proof:BalanceSheetLineItems			1
Presentation	BalanceSheet	proof:BalanceSheetLineItems	Parent-Child	proof:BalanceSheetSet			2
Presentation	BalanceSheet	proof:BalanceSheetSet	Parent-Child	proof:Assets			3
Presentation	BalanceSheet	proof:BalanceSheetSet	Parent-Child	proof:Liabilities			4
Presentation	BalanceSheet	proof:BalanceSheetSet	Parent-Child	proof:Equity			5
Definition	BalanceSheet	proof:BalanceSheetLineItems	RootMember-Member	proof:Assets			1
Definition	BalanceSheet	proof:BalanceSheetLineItems	Member-Member	proof:Liabilities			2
Defin	BalanceSheet	proof:BalanceSheetLineItems	Member-Member	proof:Equity			3

In phase 1, XBRL can be output but the application does not directly verify the XBRL output. But, the application does generate the correct XBRL output which is then verified by a separate tool that can perform all the necessary XBRL verification tasks.

Phase 2: The objective of phase to is to provide an improved GUI/UX for manually entering information into the report. In the prior phase for example, associations where entered into a database table as such.

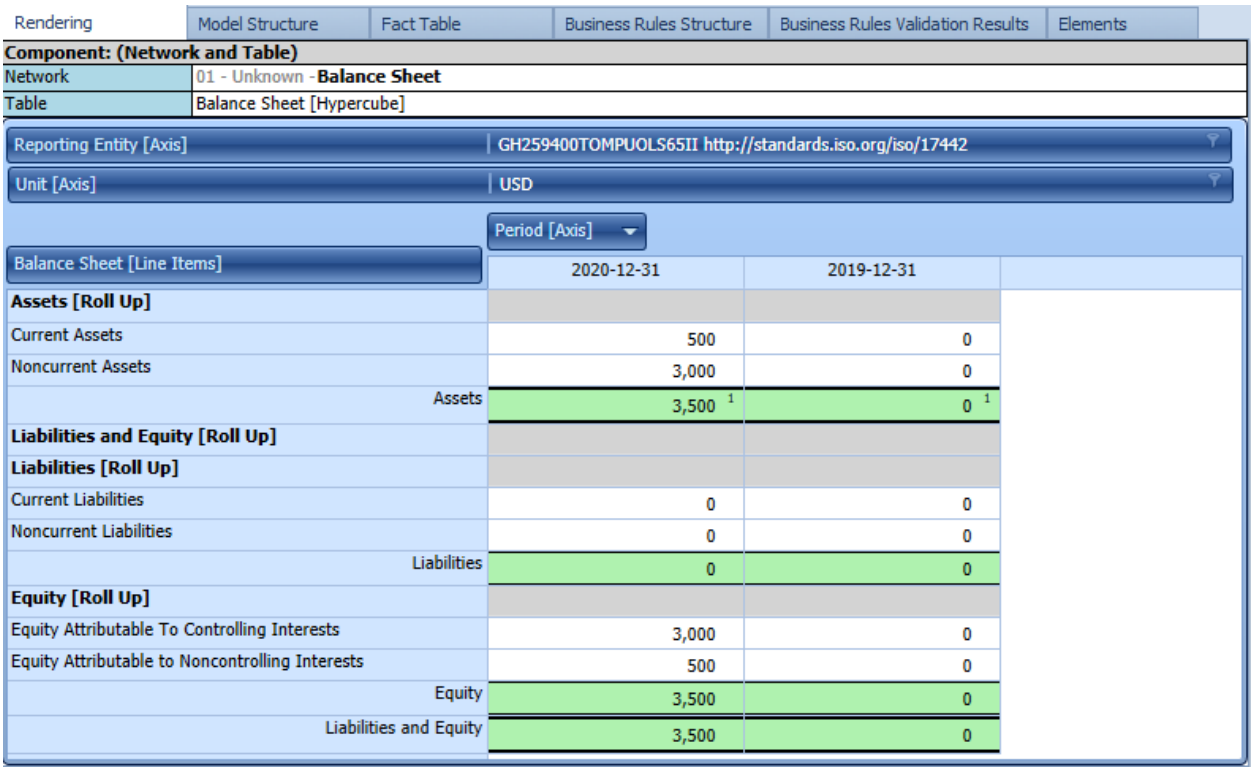
In the phase 2 interfaces will be changed to be more functional, for example rather than the database table for entering information as in phase 1, in phase 2 the accusations would be entered within an interface similar to the following: (i.e. editing associations using a tree view control or such)

-	Element Type	Balance	Period	Name	Order
Presentation View					
01-Balance Sheet	Extended Link				
Balance Sheet [Hypercube]	Table	na	duration	proof:BalanceSheetHypercube	0
Balance Sheet [Line Items]	Abstract	na	duration	proof:BalanceSheetLineItems	1
Assets [Roll Up]	Abstract	na	duration	proof:AssetsRollUp	2
Current Assets	Element	debit	instant	proof:CurrentAssets	3
Noncurrent Assets	Element	debit	instant	proof:NoncurrentAssets	4
Assets	Element	debit	instant	proof:Assets	5
Liabilities and Equity [Roll Up]	Abstract	na	duration	proof:LiabilitiesAndEquityRollUp	6
Liabilities [Roll Up]	Abstract	na	duration	proof:LiabilitiesRollUp	7
Current Liabilities	Element	credit	instant	proof:CurrentLiabilities	8
Noncurrent Liabilities	Element	credit	instant	proof:NoncurrentLiabilities	9
Liabilities	Element	credit	instant	proof:Liabilities	10
Equity [Roll Up]	Abstract	na	duration	proof:EquityRollUp	11
Equity Attributable To Controlling Interests	Element	credit	instant	proof:EquityAttributableToControllingInterests	12
Equity Attributable to Noncontrolling Interests	Element	credit	instant	proof:EquityAttributableToNoncontrollingInterests	13
Equity	Element	credit	instant	proof:Equity	14
Liabilities and Equity	Element	credit	instant	proof:LiabilitiesAndEquity	15

This type of interface is used in phase 2 because it is relatively easy to create and it is very usable. But, there is an even better way which will be created in Phase 3.

[115] Luca which is a Phase 1 report creation tool, http://luca.yaxbrl.com/

Phase 3: In phase three when developers are familiar with the rendering, model structure, fact table, rules, verification, and report elements the real interface we want will be created. This interface for editing the report is the rendering itself:



Rendering | Model Structure | Fact Table | Business Rules Structure | Business Rules Validation Results | Elements

Component: (Network and Table)

Network	01 - Unknown - **Balance Sheet**
Table	Balance Sheet [Hypercube]

Reporting Entity [Axis]	GH259400TOMPUOLS65II http://standards.iso.org/iso/17442
Unit [Axis]	USD

Period [Axis]

Balance Sheet [Line Items]	2020-12-31	2019-12-31
Assets [Roll Up]		
Current Assets	500	0
Noncurrent Assets	3,000	0
Assets	3,500 [1]	0 [1]
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	0	0
Noncurrent Liabilities	0	0
Liabilities	0	0
Equity [Roll Up]		
Equity Attributable To Controlling Interests	3,000	0
Equity Attributable to Noncontrolling Interests	500	0
Equity	3,500	0
Liabilities and Equity	3,500	0

In phase 3, the user will be able to edit an information Block (or any report fragment preferably) from within the rendering. The rendering, which is just a table, can have rows and/or columns added or removed using the notion of slots mentioned earlier. The user can still edit the report within the fact table, model structure, elements, and so forth…but they will also be able to edit the report by directly editing the rendering.

Also, part of phase 3 is understanding and using the logical "glue" that is used to hold an information block together. As part of this phase, software engineers gain an intimate understanding of the logical model and the use of the logical model to help the business professional that uses the software. This prepares the software engineer for the next phase.

Phase 4: In phase 4 the focus will be on recommender systems, wizards, templates, exemplars, and other artificial intelligence related features which help the user be more efficient and effective.

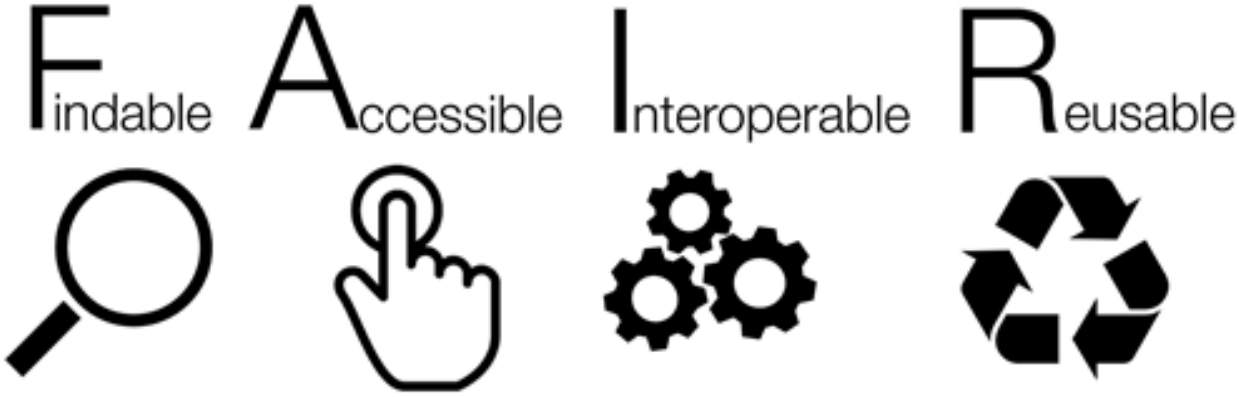
GUI/UX

One of the biggest challenges in creating a proper expert system for creating financial report will be to create a graphical user interface (GUI) and user experience (UX) that is (a) as close to a desktop application as possible and (b) as good as interface and as easy to use as something like Microsoft Excel. Syncfusion Software GUI/UX[116] is an example of the bar that needs to be hit.

[116] *Syncfusion Software GUI/UX*, http://xbrl.squarespace.com/journal/2022/3/11/syncfusion-software-guiux.html

F.A.I.R. Principles

Data and metadata should follow F.A.I.R. principles[117]. These guidelines help make data and metadata (digital assets): Findability, Accessibility, Interoperability, and Reuse. The FAIR principles emphasize machine-actionability[118] because humans increasingly rely on computational support to deal with data as a result of the increase in volume, complexity, and creation speed of data



Conclusion

Hard problems call for great efforts. Great effort was undertaken to attempt to bury complexity, achieve elegance, such that the millions of professional accountants that will use this system will benefit from that great effort.

If an expert system for creating financial reports can be built, which I believe it can based on existing software and working proof of concepts; that software will be useful, it will be novel, and it will offer a new method for creating financial reports.

As was said at the beginning of this document:

"Great things are done by a series of small things brought together." *Vincent Van Gogh*

This information is made available to help accountants and software engineers get the foundations of such a system right. Software vendors should compete with each other in terms of the value add of the functionality that they provide. Fundamental functionality should simply just work.

[117] Go-fair.org, *FAIR Principles*, https://www.go-fair.org/fair-principles/
[118] Wikipedia, *FAIR Data*, https://en.wikipedia.org/wiki/FAIR_data