Understanding Method

Best practices based method for creating a process control mechanism that consistently yields high-quality XBRL-based financial reports where the system can be "reshaped" or "altered" by report creators

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

March 7, 2021 (DRAFT)

> "I skate to where the puck is going to be, not where it has been." *Wayne Gretzky*, legendary Canadian hockey star

Executive summary:

- This document explains, at a high level, a proven, reliable, best practice method for implementing XBRL-based financial reporting following the forthcoming OMG Standard Business Report Model (SBRM).
- This method is specifically designed to address issues which come about when the extensibility features of XBRL are employed which allow report creators to "reshape" or "alter" or other such modifications.
- Report creator alterations must be controlled in order to maintain report quality, avoiding potential contradictions and inconsistencies.

Copyright (full and complete release of copyright)

This document explains at a high level a proven standard method of implementing a standard digital financial report using the XBRL technical syntax leveraging the extensibility features of XBRL which follow the forthcoming OMG *Standard Business Report Model* (SBRM)[1].

The intent of this document is to summarize know-how. This know-how, when documented in the form of a useful method, eliminates the need for others to re-invent the wheel. Rather than re-inventing the wheel; others can simply leverage a well-thought-through, world-class approach that has been designed, created, rigorously tested, and carefully engineered leveraging approaches that have been proven to work results.

These best practice approaches and techniques that has been generally demonstrated as superior to any known alternatives because the techniques produce results that are superior to those achieved by other means or because it has become a standard way of doing things are documented in this resource. It is anticipated that others will likely improve upon this method over time.

This method provides a process control mechanism that, when followed, will consistently yield high-quality XBRL-based digital financial reports. For full details of this method, please see *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*[2].

To understand this method it is critically important to understand certain specific background information, so that is where this document starts. To get started we will provide important understanding about how computers work and a grounding in artificial intelligence.

Computer Empathy and AI in a Nutshell

The following is a short summary of the document *Computer Empathy*[3] which explains that both computers and certain aspects of accounting work per the rules of mathematics.

If accountants can (1) improve their understanding of how computers work and (2) represent some of their accounting knowledge in a more formal way, this will lead to accountants and technology providers having much more productive conversations and path the way to computers being able to do some of the accountant's repetitive, mechanical manual work.

[1] OMG, *Standard Business Report Model* (SBRM), https://omgwiki.org/SBRM/doku.php
[2] *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*, http://www.xbrlsite.com/2020/Theory/SBRM-Method.pdf
[3] *Computer Empathy*, http://xbrlsite.azurewebsites.net/2018/Library/ComputerEmpathy.pdf

To understand how to get a computer to do work, it is important to understand the strengths of computers and the obstacles that get in the way which we will highlight now along with a few other important details.

Strengths of Computers

Computers seem to perform magic. How computers do what they do tends to be a mystery to many people. But computers are simple machines that follow very specific instructions. The strengths of computers can be summarised as follows. Computers can:

- store information
- retrieve information
- process stored information
- make information accessible to individuals or other machines or software

Obstacles – Communication & Understanding

The accounting profession is yet to fully leverage the strengths of computers mainly due to the following general obstacles that tend to get in the way:

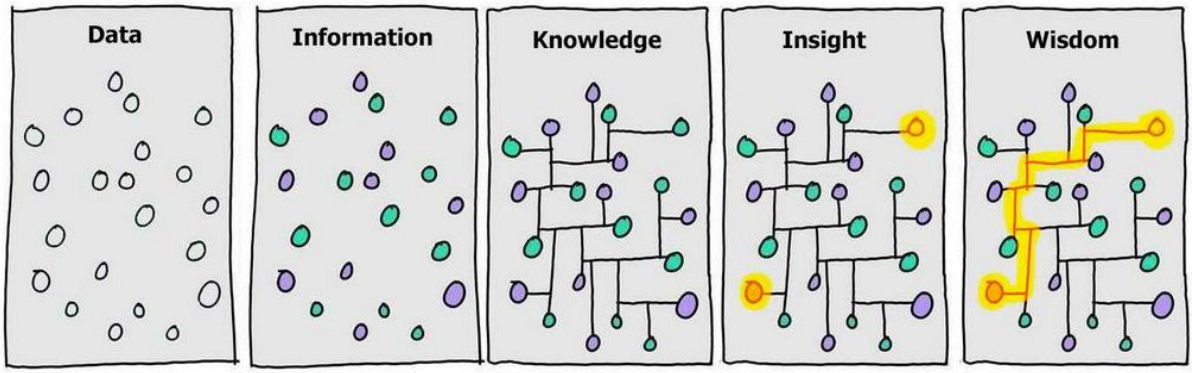
- accountants use different terminologies to refer to exactly the same thing
- accountants differ in their understanding and interpretation of accounting standards
- accountants don't understand technologies' limitations
- IT professionals use different technology stacks and languages to achieve the same result
- IT and business professionals have an oversimplified view of accounting

Data versus Information vs Knowledge

We are working with information, not data. The difference between data and information is that data is the raw facts and numbers where information is data in context. This is important to understand as most problems faced by accountants are an information problem, rather than a data problem. Getting data is easy. Knowing what that data represents and how the data fits together is difficult. Representing information in the form that a machine such as a computer can understand and use that information is difficult.

Knowledge is a set of data and information and a combination of skill, know-how, experience which can be used to improve the capacity to take action or support a decision making process by categorizing, collating, associating the data and information[4].

[4] YouTube.com; *Data, Information, Knowledge*; https://youtu.be/3NxN0OgVN2k



This is important to understand as most problems faced by accountants are an information problem, rather than a data problem. Getting data is easy. Knowing what that data represents and how the data fits together is difficult. Representing information in the form that a machine such as a computer can understand and use that information is difficult.

Standards

Standards can help overcome the obstacles above but won't eliminate them. Good examples of standards that have helped change the world are standard shipping containers, uniform product codes (barcodes) and standard electrical outlets. It is highly unlikely to get everyone to agree so providing options can be a good thing. XBRL is a global standard for business reporting and is an ontology-like thing (explained below) that can represent financial reports digitally. XBRL can be leveraged for automation of accounting, reporting, auditing and analysis processes and tasks. To do that, you use a knowledge system.

Knowledge Based Systems

The better the capability of a system to represent knowledge, the better the ability for a software application to read and process that knowledge and perform useful work for the user of the system.

- **A dictionary** would be a simple flat inventory of terms with no relations.
- **A thesaurus** would document some relations between broader and narrower terms. This is more useful than a simple dictionary.
- **A taxonomy** provides descriptions and a limited amount of structure generally in the form of one hierarchy. This is more useful than a thesaurus.
- **An ontology** is a model that tends to provide descriptions and multiple structures and therefore tends to have more than one hierarchy. For example, a set of taxonomies which explicitly differentiate types of relations or associations between terms could

- **A logical theory** is a set of **models** (ontology like things) that are consistent with the logical theory. A logical theory provides way of thinking about a domain by means of deductive reasoning to derive logical consequences of the theory.

I have created a logical theory that describes the mechanical aspects and dynamics of a financial report[5]. But to get a knowledge system to work, you have to put knowledge into that system.

Logical theory

In very simple terms, a logical theory is a set of models that are permissible per that logical theory. Those models are constructed by making logical statements which specify:

- **Terms** (things)
- **Relations** (associations between things) e.g. "type-of", "is-a" or "sub-class"
- **Structures** (sets of associations between things)
- **Assertions** (rules things and associations follow) such as if-then…
- **Facts** (values)
- World view (e.g. closed world assumption, unique naming assumption and negation as failure)

Financial Report Levels

To clearly and precisely understand XBRL-based digital financial reporting and the target level of this method, it helps to think of the spectrum of financial reports in terms of levels similar to how levels are helpful in understanding the capabilities of self-driving cars[6]. The term "self-driving" means different things to different people so it makes it difficult to have a precise conversation about that topic. But breaking the description into a spectrum of descriptions is very helpful to the communication process.

This is similarly true for the levels of an XBRL-based digital financial report. Below we will break down a financial report into helpful levels that will enable a precise and clear discussion. We will provide a very brief description, a little bit of information, and a link to specific examples that instantiate a report per each specific level. The marginal difference between each level is very helpful in providing the reader with a solid understanding of the different levels. Here is an overview of the levels related to financial reporting as I see them beginning with the least functional in terms of both human and machine use of the information from with a financial report.

---

[5] *Logical Theory Describing Financial Reports*, http://www.xbrlsite.com/2020/Theory/LogicalTheoryDescribingFinancialReport.pdf

[6] Truecar, The 5 Levels of Autonomous Vehicles, https://www.truecar.com/blog/5-levels-autonomous-vehicles/

- **Level 0**: Not machine readable. *An example of Level 0 is a clay tablet, papyrus, or paper as the report medium*.
- **Level 1**[7]: Machine readable, nonstandard, structured for presentation. *PDF, HTML, or XHTML are examples of Level 1*.
- **Level 2**[8]: Machine readable, nonstandard, structured for meaning, no taxonomy (a.k.a. dictionary), no rules, no report model. *An XBRL-based report without an XBRL taxonomy schema, without XBRL relations and resources, and without XBRL Formulas is an example of Level 2*.
- **Level 3**[9]: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary), incomplete rules, incomplete high-level report model. *An XBRL-based report with a XBRL taxonomy schema, with XBRL relations and resources, but without XBRL Formulas is an example of Level 3.*
- **Level 4**[10]: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary), complete set of rules provided, incomplete high-level report model. *An XBRL-based report with a XBRL taxonomy schema, with XBRL relations and resources, and with XBRL Formulas that completely describes the report is an example of Level 4*.
- **Level 5**[11]: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary), complete set of rules provided, complete global standard high-level report model, yields PROVEN properly functioning system and UNDERSTANDABLE report information. *An XBRL-based report with all the characteristics of Level 4, plus consistency cross checks, type-subtype relations, consistent modeling of XBRL presentation relations, information that describes the correct representation of every disclosure within the report, and a reporting checklist that describes all required disclosures is an example Level 5*.
- **Level 6**: All of Level 5 PLUS blockchain-anchored XBRL to increase trust. *An XBRL-based report with all the characteristics of Level 5, plus information within a digital distributed ledger that assures no one has tampered with the report is an example of Level 6*.
- **Level 7**: All of Level 6 PLUS blockchain-anchored transactions and events. *An XBRL-based report with all the characteristics of Level 6, plus information that indicates that assures no one has tampered with transactions is an example of Level 7*.

[7] Level 1 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level1/
[8] Level 2 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level2/
[9] Level 3 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level3/
[10] Level 4 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level4/
[11] Level 5 financial report example, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-level5/

The target of this method is Level 5 and above. Below Level 5 the functionality what we generally need from such reports in terms of quality and effective use of reported information in automated machine-based processes is not good enough.

Creating the knowledge to store in the system

There are two general approaches to creating knowledge to store in a knowledge base:

1. Let the computer work it out by using AI, machine learning or other approaches. This means, feed the computer a load of data and let it figure out the patterns. (pattern-based, machine learning)
2. Tell the computer what the knowledge is. Accountants and auditors are highly trained and have the knowledge in their head. All we need is a way of capturing that knowledge and storing it in an 'ontology' and a knowledge base of rules. (rule-based)

It is not an either-or question. But option 2 needs to be prioritised because it will provide the foundation for AI and machine learning to build on. Machine learning excels where there is a high tolerance for error. There is an extremely low tolerance for error in financial accounting, reporting, auditing, and analysis.

A knowledge based system draws upon the knowledge of human experts, i.e. accountants and auditors. The more knowledge in the knowledge base, the more the knowledge based system can do. The right information can literally supercharge what can be achieved using artificial intelligence.

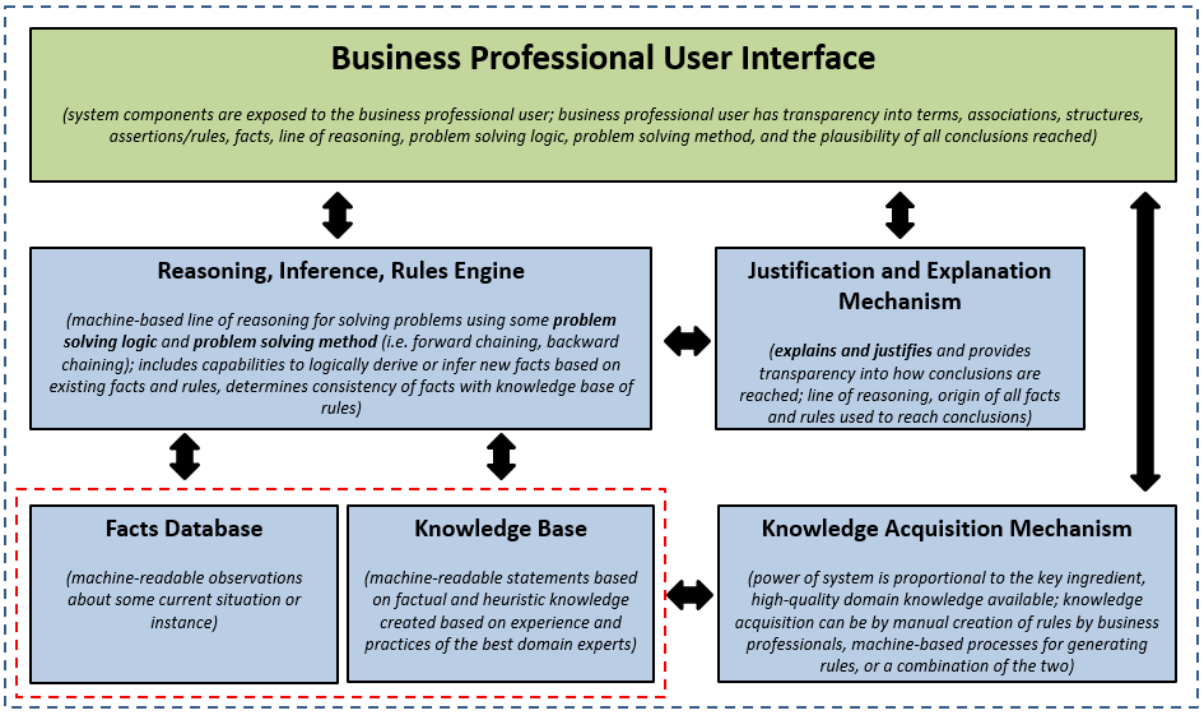
To understand the capabilities of a knowledge based system, it is important to understand the components of such knowledge based systems.

Components of a Knowledge Based System

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

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

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

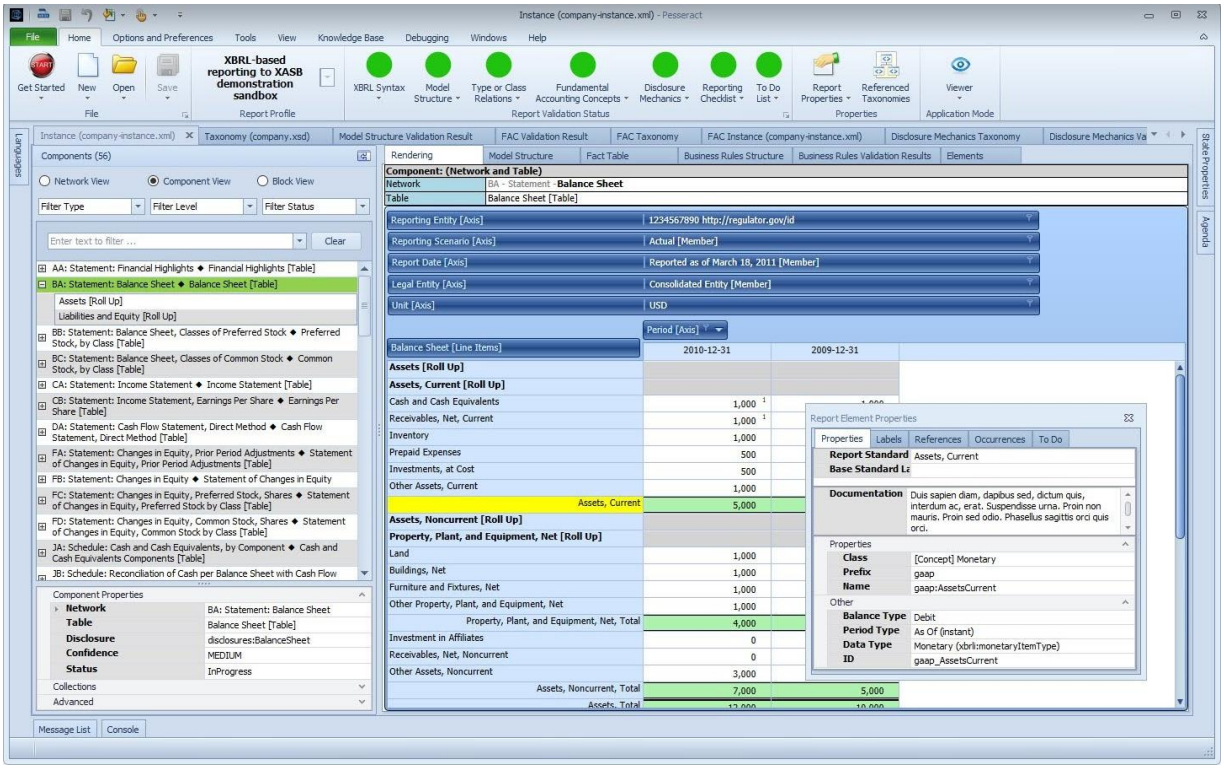


Business Professional User Interface

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

The following is one of a number of screen shots[12] of the working proof of concept software application Pesseract which provides an example of a user interface with which a business professional would likely interact:

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



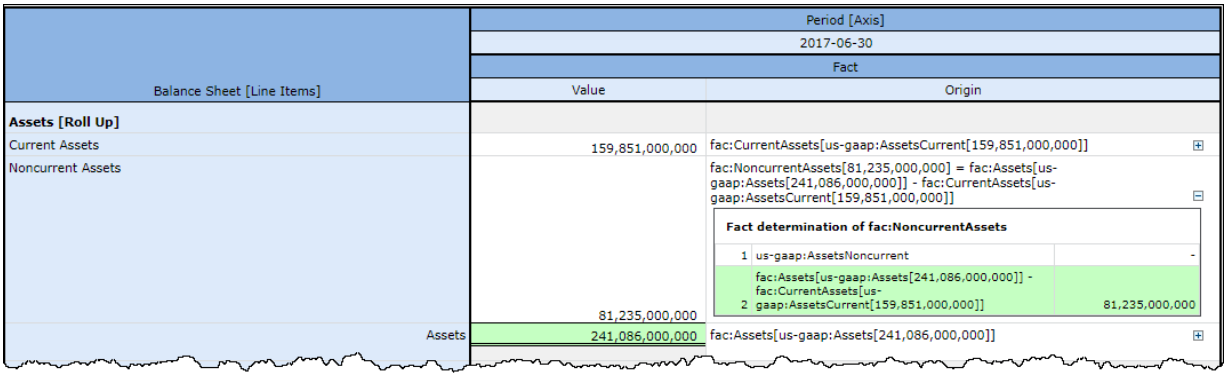
The user interface is non-technical requiring only business and accounting knowledge to effectively understand the software application and how to use it.

Justification and Explanation Mechanism

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

Below you see the fundamental accounting concept relations continuity cross check verification checks provided by XBRL Cloud's Evidence Package[13] which is a review tool that can be used to verify XBRL-based financial reports:

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



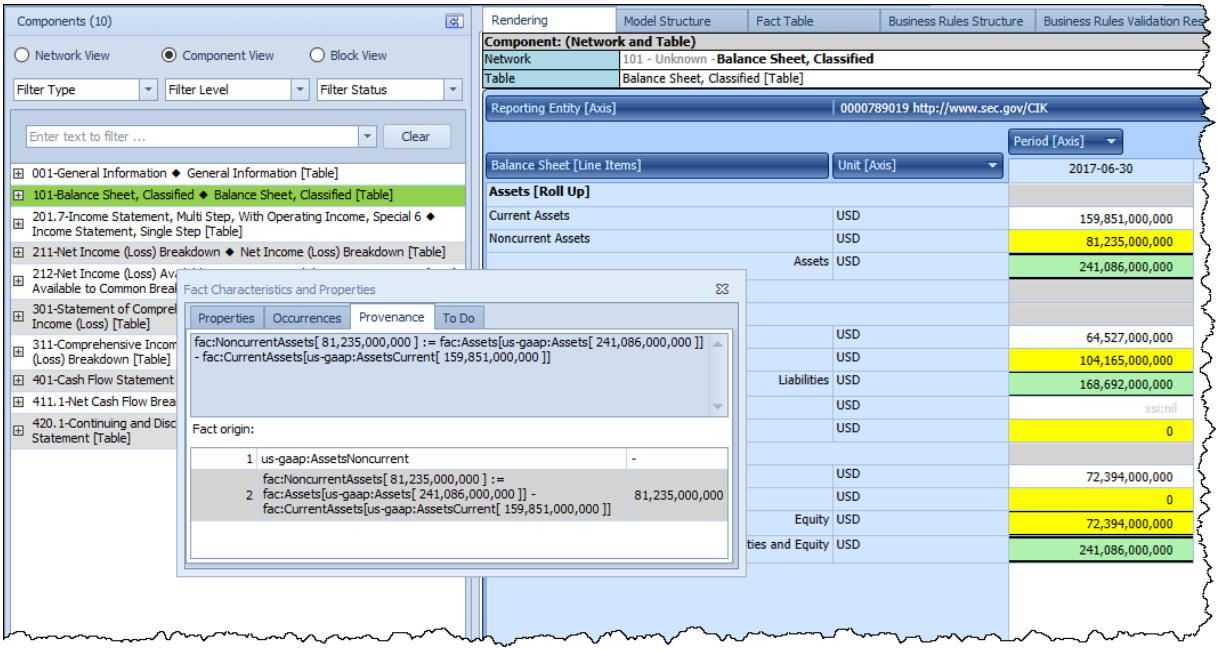
Balance Sheet [Line Items]	Period [Axis] 2017-06-30 Fact Value	Origin
Assets [Roll Up]		
Current Assets	159,851,000,000	fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]
Noncurrent Assets	81,235,000,000	fac:NoncurrentAssets[81,235,000,000] = fac:Assets[us-gaap:Assets[241,086,000,000]] - fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]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

Fact determination of fac:NoncurrentAssets

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

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.

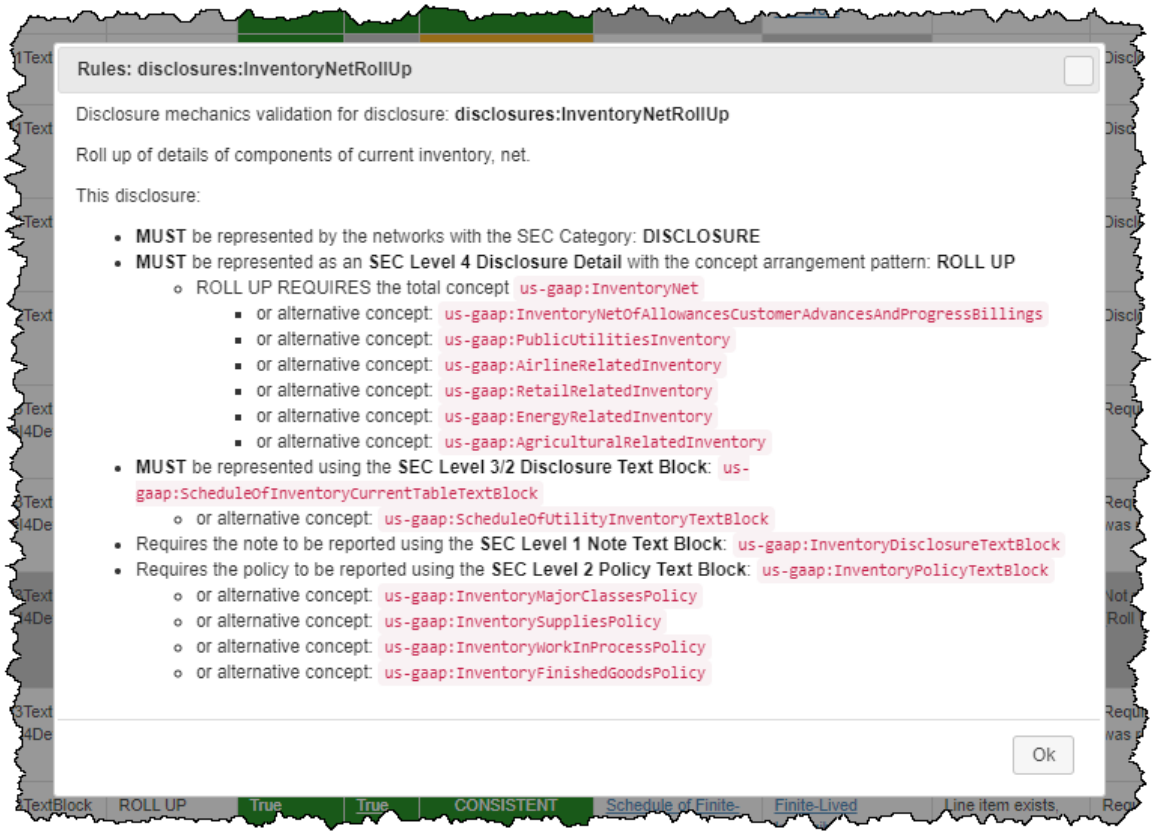
Pesseract provides similar functionality:



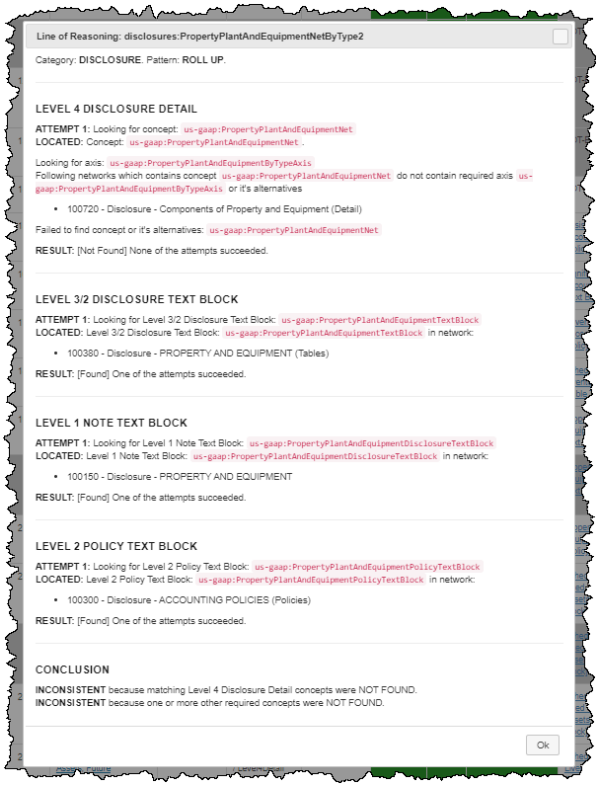
XBRL Cloud's Disclosure Mechanics and Reporting Checklist[14] provides the rules used, line of reasoning used, and conclusions reached for determining if a disclosure is structured consistent with its expected specification:

Disclosure mechanics rules:

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



Line of reasoning:



Conclusions reached:

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

Similar functionality is offered by Pesseract:

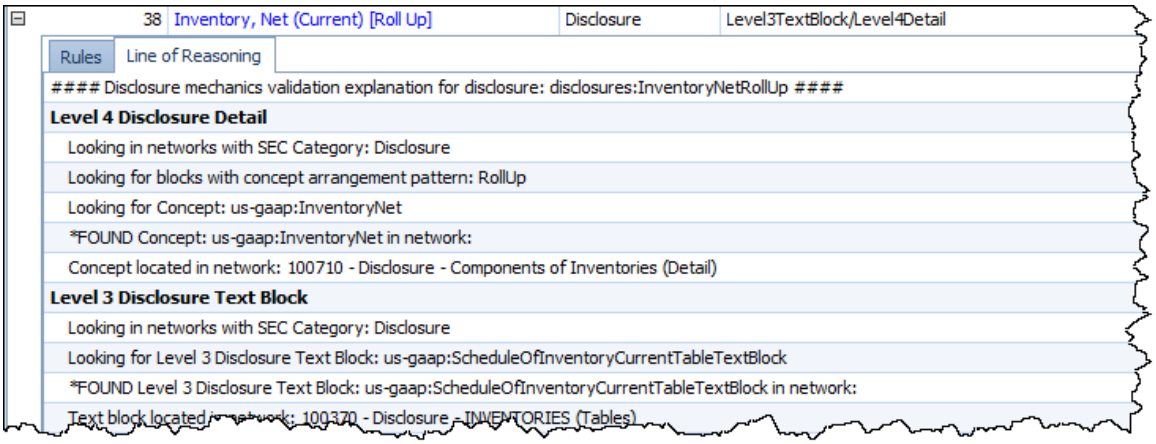
Disclosure mechanics rules:

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

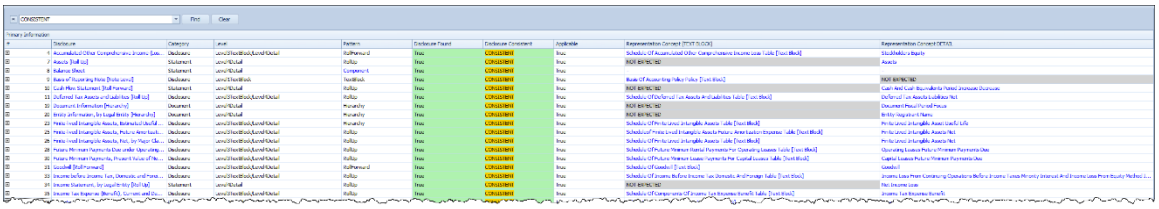
Rules | Line of Reasoning

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

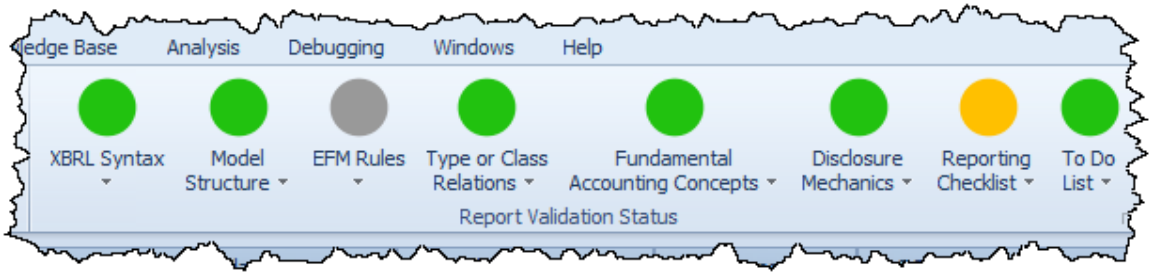
Line of reasoning:



Conclusions reached[15]:



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)[16]:



Reasoning, Inference, Rules Engine

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

[15] Pesseract disclosure mechanics verification of 94.8% of all 124 disclosures verified, http://xbrlsite.azurewebsites.net/2020/Prototype/Microsoft/Microsoft2017_Discovery.jpg
[16] SBRM Progress Report, http://xbrl.squarespace.com/journal/2020/1/30/sbrm-progress-report.html

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

- **Ontology + Rules**: For example, OWL[17] (or SWRL[18]) + SHACL[19] + RDF[20] (or N3[21]) provide sufficient fragments of first order logic. (Some call this Modern Symbolic AI[22])

- **Modern Prolog**: Prolog such as SWI Prolog[23] or Scryer Prolog[24] seem to have all of the necessary functionality. The up side is that there are a lot of Prolog implementations[25]. The down side is that none of these Prologs can call itself "the standard". Each has pros and cons. Prolog interoperates with relational (SQL) databases.

- **ISO Prolog**: ISO has created a standard Prolog[26]. ISO Prolog can be regarded as a subset of Full Prolog. There is solid motivation for implementations to support ISO Prolog as the international standard Prolog, many already do to one degree or another.

- **Datalog**: Datalog[27], or "function-free Horn Logic", is more tractable than Horn Logic[28] (Pure Prolog) and ISP Prolog (Full Prolog). RuleML.org points out[29], "Datalog is the language in the intersection of SQL and Prolog. It can thus be considered as the subset of logic programming needed for representing the information of relational databases, including (recursive) views." So Datalog interoperates with relational databases.

- **PSOA RuleML**: PSOA[30] (Positional-Slotted Object-Applicative) RuleML is a multi-paradigm, particularly graph-relational, data and rule language. PSOA interoperates with graph and relational databases. RuleML.org points out[31], "PSOA RuleML's databases (fact bases) generalize the instance level of Graph and Relational Databases; its knowledge bases complement facts by rules for deductive retrieval (extending the Datalog-level, function-free expressiveness of Deductive Databases to the Horn-logic

[17] W3C, *OWL*, https://www.w3.org/TR/owl2-overview/
[18] W3C, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, https://www.w3.org/Submission/SWRL/
[19] W3C, *SHACL*, https://www.w3.org/TR/shacl/
[20] W3C, RDF, https://www.w3.org/RDF/
[21] W3C, *Notation3 (N3): A readable RDF syntax*, https://www.w3.org/TeamSubmission/n3/
[22] Shawn Riley, *Modern Symbolic AI in 2020*, https://medium.com/@shawn.p.riley/modern-symbolic-ai-in-2020-dfcc27abbc5c
[23] SWI Prolog, https://www.swi-prolog.org/
[24] Scryer Prolog, https://github.com/mthom/scryer-prolog
[25] Wikipedia, *Comparison of Prolog Implementations*, https://en.wikipedia.org/wiki/Comparison_of_Prolog_implementations
[26] ISO, *ISO Prolog*, https://www.iso.org/standard/21413.html
[27] Wikipedia, *Datalog*, https://en.wikipedia.org/wiki/Datalog
[28] Wikipedia, *Horn Logic*, https://en.wikipedia.org/wiki/Horn_clause
[29] RuleML.org, http://ruleml.org/papers/Primer/RuleMLPrimer2012-08-09/RuleMLPrimer-p3-2012-08-09.html
[30] RuleML.org, *PSOA*, http://wiki.ruleml.org/index.php/PSOA_RuleML
[31] RuleML.org, *PSOA RuleML Bridges Graph and Relational Databases*, https://wiki.ruleml.org/index.php/PSOA_RuleML_Bridges_Graph_and_Relational_Databases

expressiveness of Logic Programming), interoperation, and reasoning, as well as for optionally emulating part of the schema level."

- **GQL/Cypher**: GQL[32] is an ISO project[33] to create a global standard query language (like SQL) for graph databases, graph query language. Open Cypher[34] which is based on Cypher is the query language of Neo4j.

- **SQL + More**: While it is proven[35] that you can store XBRL-based information in a relational database; you have to add functionality to process the information. Essentially, you have to construct a rules engine to process the information and prove the system is properly functioning. This is very possible but tends to not be very efficient.

- **XBRL + SBRM + More**: XBRL[36] is an open standard technical syntax published by XBRL International, SBRM[37] is a forthcoming standard to be published by OMG that formalizes a logical conceptualization of a business report. While XBRL provides the functionality to represent all that is needed to express knowledge and much of what is necessary to process that knowledge and prove the knowledge is represented correctly. However, certain specific processing is missing that must be supplemented to create a complete system. As such, that additional processing logic must be provided.

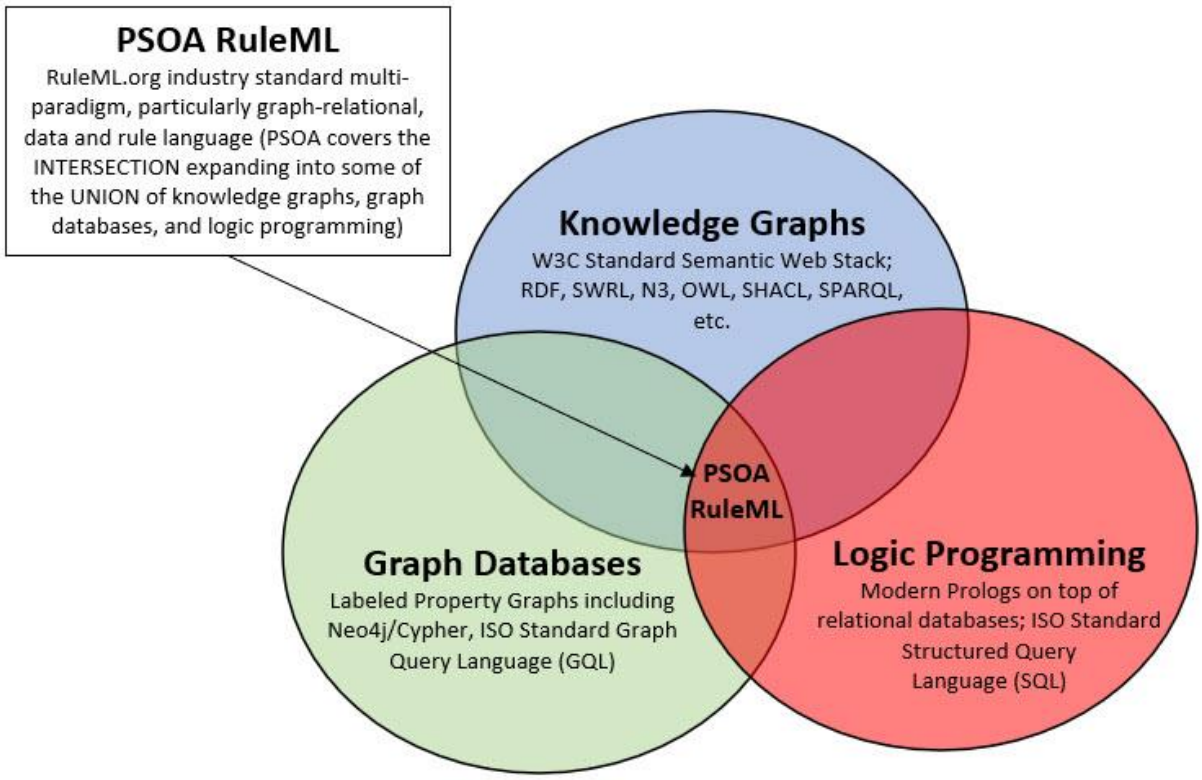
There are undoubtedly other logic engines that can be used to process XBRL-based digital financial reports. Other completely different approaches such as the decision model approach[38] could possibly be used but would need to include an ontology-type component. Any syntax used should be 100% convertible to all other syntaxes and be able to round tripped back into the original syntax. Then, you could switch between whatever approach you wanted.

Converting between these logics is very possible. For example, converting between RDF and labeled property graphs is possible[39]. Converting from RDF to SWI Prolog is possible[40]. But 100% conversion is limited to the least common denominator, the set of logic that each alternative possesses.

---

[32] GQL Standards.org, *GQL Standard*, https://www.gqlstandards.org/
[33] Wikipedia, *GQL Graph Query Language*, https://en.wikipedia.org/wiki/GQL_Graph_Query_Language
[34] OpenCypher.org, *Open Cypher*, https://www.opencypher.org/
[35] Proof representation, http://xbrlsite.azurewebsites.net/2020/master/proof/index.html
[36] XBRL International, https://www.xbrl.org/
[37] OMG, SBRM, https://www.omg.org/intro/SBRM.pdf
[38] Wikipedia, *Decision Model*, https://en.wikipedia.org/wiki/Decision_model
[39] Neo4j, Jesús Barrasa, *RDF Triple Stores vs. Labeled Property Graphs: What's the Difference?*, https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/
[40] Samuel Lampa, *SWI-Prolog as a Semantic Web Tool for semantic querying in Bioclipse: Integration and performance benchmarking*, https://www.diva-portal.org/smash/get/diva2:398839/FULLTEXT01.pdf

It is unlikely that every enterprise will use the same approach. This graphic shows how these different problem solving paradigms relate to one another[41]:



The following are examples of the sorts of reasoning, inference, and rules engine which could be used[42]:

[41] Primary Problem Solving Logic Paradigms, http://xbrl.squarespace.com/journal/2020/9/15/primary-problem-solving-logic-paradigms.html
[42] Rules Engine Comparison, http://xbrlsite.azurewebsites.net/2020/Library/RulesEngineComparison.jpg

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

Currently, while no one single rules engine can process 100% of what is required to be processed[43] a financial report or an accounting process automation workflow[44]; the above processors can be combined to achieve 100% of the capabilities which are necessary[45].

Fact Database

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

There are many approaches to storing facts within a database[46]. Each approach has a set of PROS and CONS; no approach is 100% the best or 100% the worst. What appear to be the most viable alternatives include:

- **SQL database**: These are the most pervasive and the most popular today.
- **RDF triple store**: These are popular for working with the W3C Semantic Web Stack. These are sometimes implemented within a SQL database.

[43] Charles Hoffman, CPA, *Chain of Capabilities Necessary to Automate Accounting Processes*, http://xbrlsite.azurewebsites.net/2018/Library/ChainOfCapabilities.pdf
[44] Charles Hoffman, CPA, et. al., *Understanding Digital*, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingDigital.pdf
[45] Continuous Accounting Workflow Prototype, http://xbrlsite.azurewebsites.net/2020/master/continuous-accounting/index.html
[46] Understanding Database/Query Options (Part 2), http://xbrl.squarespace.com/journal/2014/4/27/understanding-databasequery-options-part-2.html

- **Graph database**: Graph databases such as Neo4j[47] are increasing in popularity, standard query languages are being developed like Cypher[48].
- **NOSQL databases**: NOSQL databases such as MondoDB are increasing in popularity because they require no schema which can be a feature or a bug depending upon whether you desire a database schema.
- **DATOMIC**: Datomic[49] is a fact database or cell store[50] that has a built in DATALOG rules engine.

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

Knowledge Base

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

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

- US GAAP[51]
- IFRS[52]
- IPSAS[53]
- FRF for SMEs[54]
- US GAAP Not-for-Profit[55]

Other testing, prototype, and other such XBRL-based financial reporting schemes were represented in order to collect information which could yield information useful to create on

[47] Neo4j, http://xbrl.squarespace.com/journal/2020/7/7/neo4j.html
[48] Cypher, https://www.opencypher.org/
[49] Datomic Cloud, https://www.datomic.com/
[50] Ghislain Fourny, PhD, *Cell Stores*, https://arxiv.org/pdf/1410.0600.pdf
[51] US GAAP financial reporting scheme, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/documentation/Index.html
[52] IFRS financial reporting scheme, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ifrs/documentation/Index.html
[53] IPSAS financial reporting scheme (prototype), http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ipsas/documentation/Index.html
[54] FRF for SMEs financial reporting scheme, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/frf-sme/documentation/Index.html
[55] US GAAP Not-for-Profit financial reporting scheme, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/nfp/documentation/Index.html

framework for representing all financial reporting schemes. That information is summarized in *Mastering XBRL-based Digital Financial Reporting*[56].

In essence, it is possible to represent any financial reporting scheme[57] using the notion of profiles[58] to adjust for any minor differences between how each financial reporting scheme chooses to implement XBRL-based digital financial reporting.

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

Knowledge Acquisition Mechanism

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

There are three approaches to acquiring knowledge:

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

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

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

For example, information about disclosures[59] can be used to learn how to create algorithms for identifying other such disclosures simply by probing existing XBRL-based financial reports submitted to financial regulators such as the SEC and ESMA. That machine-readable

---

[56] *Mastering XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2020/master/
[57] *Comparison of Financial Reporting Schemes High Level Concepts*, http://xbrlsite.azurewebsites.net/2018/Library/ReportingSchemes-2018-12-30.pdf
[58] *XBRL-based Digital Financial Reporting Profiles and General Business Reporting Profile*, http://xbrlsite.azurewebsites.net/2018/Library/Profiles-2018-10-22.pdf
[59] Disclosure Best Practices, http://xbrlsite-app.azurewebsites.net/DisclosureBestPractices/DisclosureBestPractices.aspx?DisclosureName=IncomeStatement

information along with humans to guide and tweak the process can be used to identify rules for other unknown disclosures by looking for specific known patterns.

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

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

Understanding the SBRM Meta-Meta Model

A meta-models is a model whose purpose is to describe and process models that subscribe to that meta-model. Models and meta-models both prescribe and describe what is permissible and what is not permissible per some model or meta-model.

Utility of Method

One reason for this is to be sure each model is consistent with the specification provided by the meta-model. The following is a summary of the utility of this method:

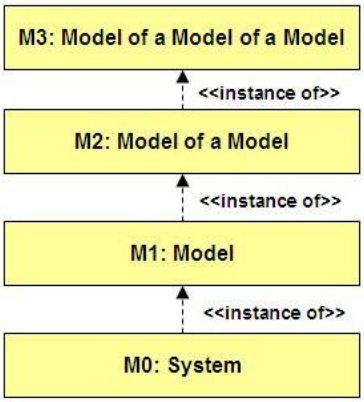
- Clear scope and purpose providing a framework and theory for thinking about financial reports[60].
- Semantic rigor. The model of a business report and financial report is thorough.
- Formally documented and unambiguous specification; XBRL for syntax and SBRM for semantics.
- Open, freely available, global standard XBRL and SBRM.
- Verified and tested leaving no stone unturned, no question unanswered, or argument about how the framework and theory work[61].

No Need to Reinvent the Wheel

Standard meta-models are critically important so individual implementations don't have to reinvent the wheel. Object Management Group (OMG) publishes something called the **Meta Object Facility** (MOF)[62]. Basically, the MOF explains the distinction between an "Object", a "Model", a "Meta Model", and a "Meta-meta Model". These ideas are commonly confused, are

[60] Charles Hoffman, CPA, *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/
[61] *Mastering XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2020/master/
[62] Wikipedia, *Meta Object Facility*, https://en.wikipedia.org/wiki/Meta-Object_Facility

not generally understood by business professionals, often not even understood by technical people, but are **CRITICALLY IMPORTANT to getting business professionals what they really want/need**. A model is essentially a pattern in a system.



What we don't need is every individual regulator, standards setter, and/or enterprise creating their own "meta model" when one common model will do. What we need is for, say, ESMA and the SEC and other regulators and others to **use the SAME META-META MODEL** where possible. If they could, why would they not?

SBRM

Generally, the answer to that question is ignorance as to the benefits of a common meta-meta model. All this is why OMG is so interested in what I have done with my framework and method and why OMG quickly understood it and creates the ***Standard Business Report Model*** (SBRM)[63] which is a syntax independent logical conceptualization of a business report.

OMG calls SBRM a "model" but it is actually a "meta-meta model" in my view. An economic entity defines objects, puts the objects into a model, which needs to conform to some regulator model (meta model) so that the regulator can collect models from many, many economic entities and be able to compare each different economic entity's information. All regulator models could fit into one common meta-meta model, SBRM. Whether a regulator chooses to take this approach is up to the regulator.

XBRL

XBRL provides the technical syntax format which physically transports the information, US GAAP or IFRS or other financial reporting scheme provides the meaning, that meaning is represented using the XBRL syntax, SBRM provides the one common report model that all reporting economic entities use.

[63] OMG, Standard Business Report Model (SBRM), https://www.omg.org/intro/SBRM.pdf

The relationships go like this:

1. An economic entity creates a report, they put their "Objects" into the report which forms the report "Model" for that economic entity.
2. The economic entity can create their own "Objects" (extension Objects) and associate them with some "Model" financial reporting scheme such as the US GAAP or IFRS taxonomies (base Objects).
3. Every economic entity creating their report "Model" must fit into some "Meta model", today ESMA and the SEC have very similar "Meta Models", but they are slightly different (unnecessarily).
4. Rather than each regulator and/or economic entity or others creating their own "Meta model"; I have created a "Meta Model" that anyone can simply pick up and use which (a) is consistent with both ESMA and the SEC (b) adds more information to make sure the reports are properly functioning (consistent, complete, precise), and (c) adds information that is specific to financial reporting.
5. My "Meta-Model" fits into the OMG Standard Business Report Model (SBRM) which is a "Meta-meta Model" for BUSINESS reporting. A financial report is a specialization of the more general business report.

So, XBRL does has a "business report meta-meta-model". It is just that XBRL International explains it rather poorly. The *Open Information Model 1.0*[64] is an attempt to define a business report meta-meta model. Further, I expanded the SBRM adding additional metadata related specifically to financial reporting using the *Logical Theory Describing Financial Report*[65].

Testing

And so, every XBRL-based financial report submitted to the SEC using US GAAP or IFRS, the ESMA field tests[66], and all my prototypes[67] fit into my one common meta-meta-model. How do I know this? Because I loaded 100% of US GAAP and IFRS XBRL-based reports, ESMA field tests, and all my prototypes into my model[68] to be certain that my model was correct. Here are the results from one of those tests:

[64] XBRL International, *Open Information Model 1.0*, https://specifications.xbrl.org/work-product-index-open-information-model-open-information-model.html
[65] *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/
[66] ESMA Field Tests, http://xbrl.squarespace.com/journal/2018/7/10/esma-field-test-information-great-information-for-testing.html
[67] *Mastering XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2020/master/
[68] *Understanding Digital*, Page 46, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingDigital.pdf#page=46

So, the actual average size of the pieces of a report are quite small. Information is skewed a bit by the relatively large number of Level 1, Level 2, and Level 3 text blocks. Of the **754,430** fact sets there are:

- **Text Blocks**: 407,392 (54%) are text blocks (Level 1 Notes, Level 2 Policies, Level 3 Disclosures)
- **Sets**: 181,063 (24%) are sets (or hierarchies, no mathematical computations)
- **Roll Ups**: 120,708 (16%) are roll ups
- **Roll Forwards**: 37,721 (5%) are roll forwards
- Other (including **Roll Forward Info**, **Adjustment**, **Variance**): 7,546 (1%) are Roll Forward Infos or something else

What is more, which we will get to elsewhere in these documents, each of the concept arrangement patterns can be associated with a specific disclosure. See this

I then created what I call my PROOF representation which includes every aspect of my model (i.e. the SBRM model) [69].

And so, you should be able to see the connections between the information from the 754,430 fact sets (a.k.a. Blocks) from above and the "Pattern" column in the PROOF representation which is used to TEST the meta-model to see if SBRM provides everything that is necessary below. (Note that "Hierarchy" and "Set" are synonyms). Every fragment of the report is represented in the rows. Every ROW is a STRUCTURE of a report that fits into the overall meta-model. Every COLUMN is a common PROPERTY of the STRUCTURE.

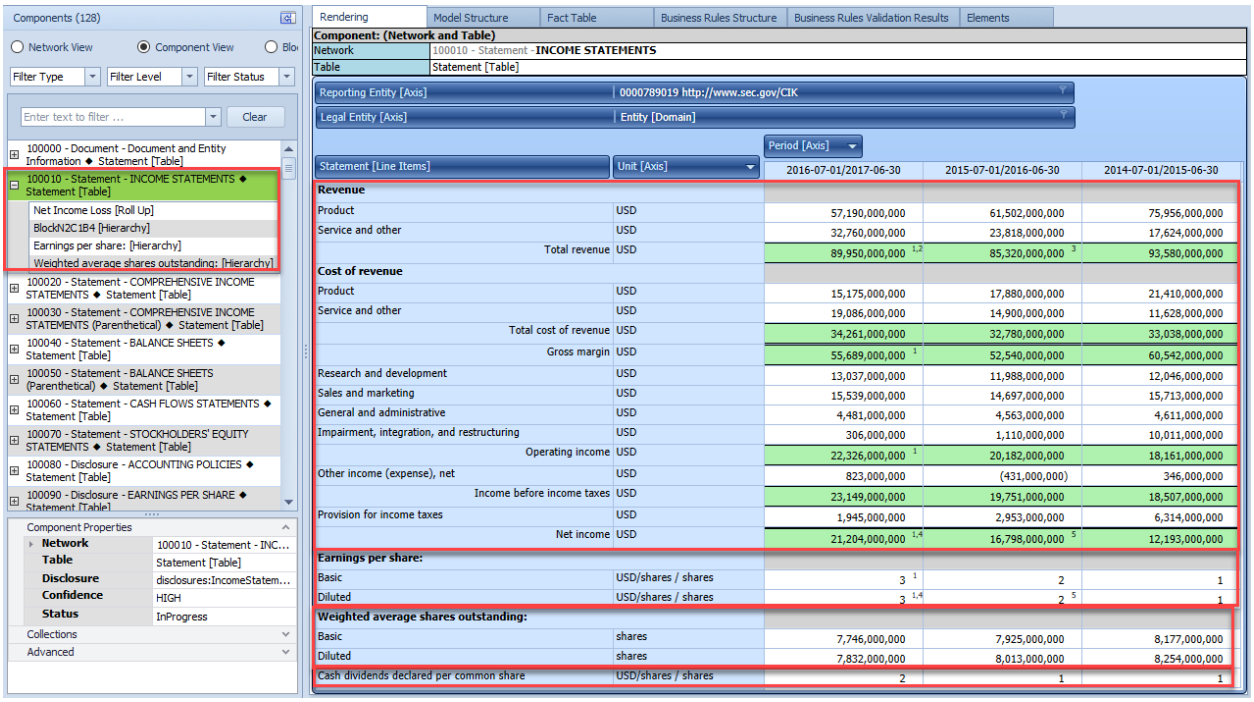
Primary Information

#	Disclosure	Category	Level	Pattern	Disclosure Fo…	Disclosure Co…	Applicable	Representation Concept [TEXT BLOCK]	Representation Concept DETAIL
1	Balance Sheet	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT-EXPECTED	Assets
2	Basis of Reporting	Unknown	Level1TextBlock	TextBlock	True	CONSISTENT	True	Basis of Reporting [Text Block]	NOT-EXPECTED
3	Changes in Equity	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Equity
4	Financial Highlights	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT-EXPECTED	Revenues
5	Income Statement	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Comprehensive Income
6	Nature of Operations	Unknown	Level1TextBlock	TextBlock	True	CONSISTENT	True	Nature of Operations [Text Block]	NOT-EXPECTED
7	Prior Period Errors	Unknown	Level4Detail	Adjustment	True	CONSISTENT	True	NOT-EXPECTED	Equity
8	Revenue Recognition Policy	Unknown	Level1TextBlock	TextBlock	True	CONSISTENT	True	Revenue Recognition Policy [Text Block]	NOT-EXPECTED
9	Segment Revenues	Unknown	Level4Detail	Hierarchy	True	CONSISTENT	True	NOT-EXPECTED	Revenues
10	Stock Plan Activity	Unknown	Level4Detail	RollForwardInfo	True	CONSISTENT	True	NOT-EXPECTED	Nonvested Fair Value
11	Variance Analysis	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Comprehensive Income

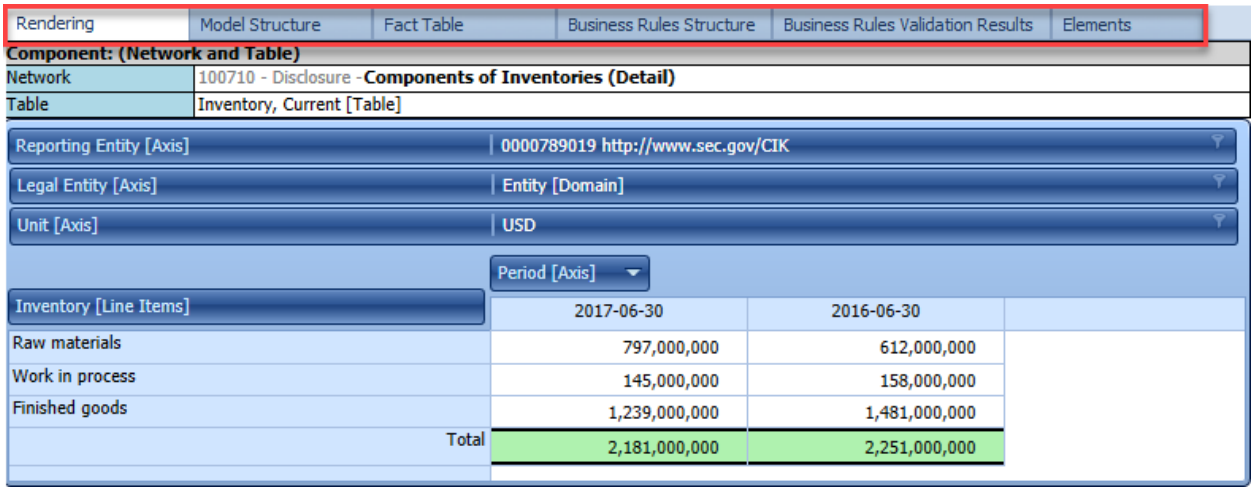
The SET of properties is the META-META MODEL of SBRM which is the SAME for EVERY economic entity that creates a report. This is likewise the SAME for every financial reporting scheme.

ROWs can be combined to create all of the arbitrary fragments of a report that are used to represent the complete report. For example, the Microsoft 10-K contains an income statement fragment[70]. That income statement fragment has FOUR structures:

[69] Proof representation, http://xbrlsite.azurewebsites.net/2020/master/proof/index.html
[70] Microsoft Income Statement fragment, http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/evidence-package/#Rendering-StatementINCOMESTATEMENTS-us_gaap_StatementTable.html



Each **structure** is described in machine-readable terms using XBRL presentation, XBRL calculation, XBRL definition, and XBRL formula relations and resources. Structures can be examined using features implemented in software applications including a "Rendering" view which is human readable, a "Model Structure" which explains the model, a "Fact Table" which is a raw set of the facts included in the structure, "Business Rules Structure" which defines the mathematical rules, "Elements" which is a list of the elements included in the model structure.



Every structure for every report from any reporting scheme works exactly this same way. The Logical Theory Describing Financial Report[71] provides all the details.

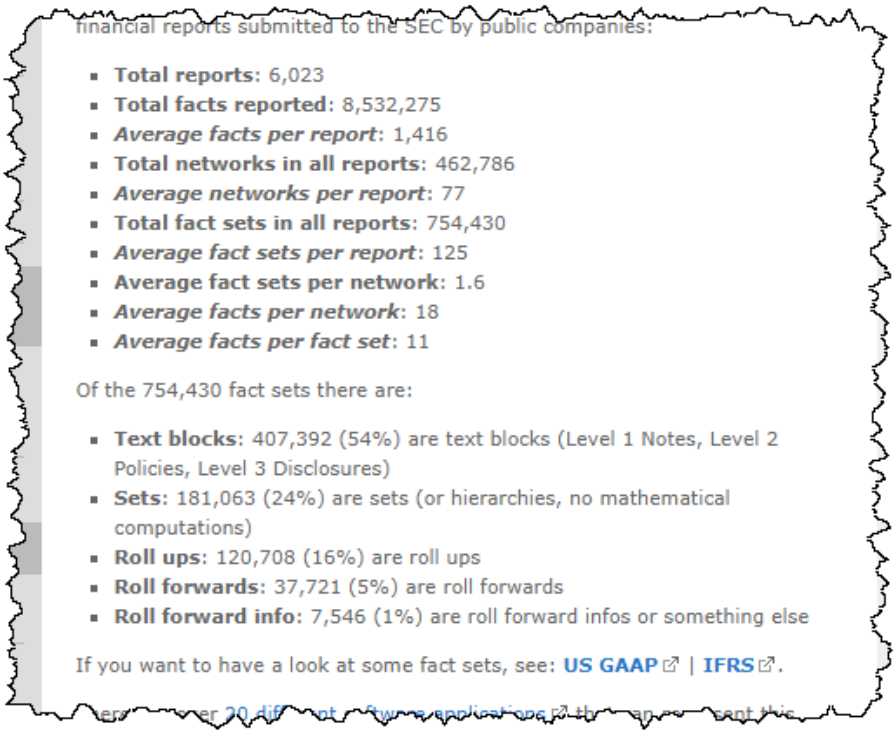
[71] *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/

Machine readable rules of the meta-meta-model both (a) describe what is permissible, (b) can be used to verify that models created are consistent with the meta-meta-model, and (c) can be used to extract information from created reports with verification that the information is consistent with what is permissible.

Process Control

This provides a **CONTROL MECHANISM** to keep quality extremely high. Rules provide the control. More rules can be added to the meta-model to expand the control mechanisms; but no set of rules can be removed from the system because if something is removed then the system is "incomplete" and errors can then slip through the system undetected.

The PROOF[72] example model can be expanded to include all the objects of the Microsoft 10-K financial report[73]. Likewise, this model works with Apple, Amazon, Google/Alphabet, Facebook, Salesforce per another analysis[74]. Likewise, this model works with every one of the 6,023 10-K financial reports analyzed which submits information using XBRL to the SEC using the US GAAP financial reporting scheme[75].



[72] *Understanding Proof*, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingProof.pdf
[73] Microsoft Analysis, http://xbrl.squarespace.com/journal/2020/4/13/microsoft-xbrl-based-report-analysis.html
[74] Software Companies Prototype, http://xbrl.squarespace.com/journal/2020/4/2/software-companies-prototype.html
[75] Breaking down the pieces, http://xbrl.squarespace.com/journal/2019/4/9/breaking-down-the-pieces-of-an-xbrl-based-digital-financial.html

Similar results were achieved with another earlier analysis of 6,751 US GAAP based 10-K financial reports[76].

Similar results were obtained from an analysis of 406 IFRS based XBRL-based financial reports submitted to the SEC.

If you strip out the "double entry accounting model" and "accounting equation" from my meta-model you get something that can be used to create high-quality general business reports (a.k.a. semantic workbook, see *Understanding Semantic Spreadsheets*[77]).

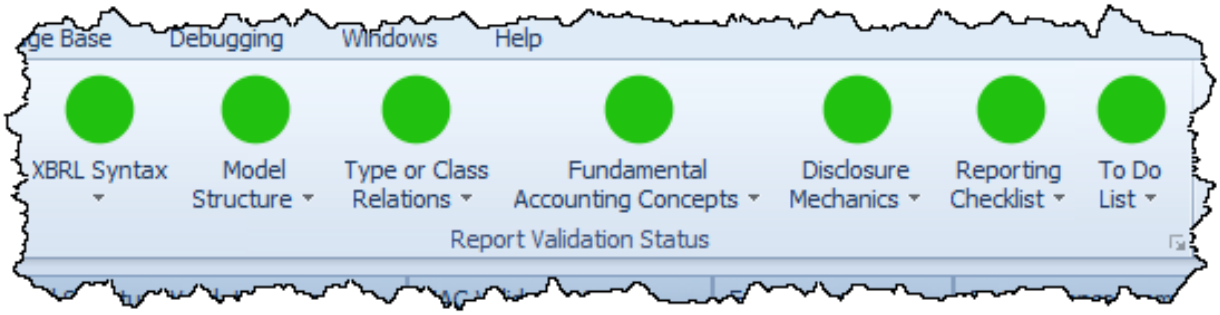
When you put all these details together it helps you understand important patterns and a method for leveraging those patterns to control report creation processes.

Overview of Method Enabled by SBRM

The following is a brief overview of the *Method of Implementing a Standard Financial Report Using the XBRL Syntax*[78].

Thorough

This graphic provides somewhat of a "dashboard" for understanding this method:



The dashboard has seven categories that are explained as follows:

1. **XBRL Syntax**: This category of rules is provided by XBRL International in the form of a machine-readable set of rules referred to as a conformance suite[79]. This conformance suite is 100% automatable via computer-based processes and used to be sure the XBRL technical format is consistent with the expectations of the XBRL Technical specification. The XBRL conformance

[76] Analysis of 6,751 XBRL-based Public Company 10-Ks Submitted to SEC, http://www.xbrlsite.com/mastering/Part05_Chapter08.F_AnalysisOf675110Ks.pdf
[77] *Understanding Semantic Spreadsheets*, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingSemanticSpreadsheets.pdf
[78] *Method of Implementing a Standard Financial Report Using the XBRL Syntax*, http://www.xbrlsite.com/2020/Theory/SBRM-Method.pdf
[79] XBRL International, XBRL 2.1, https://specifications.xbrl.org/work-product-index-group-base-spec-base-spec.html

suite has helped software vendors get their XBRL technical syntax consistent and today about 99.99% of all XBRL-based financial reports are consistent with expectation. But, this checks only the information FORMAT, not the MEANING conveyed by the information.

2. **Model Structure**: This category of rules overcomes missing information related to the relationship between the categories of report elements that are used to structure a financial report model. While the permissible sorts of XBRL calculation relations and XBRL definition relations and certain aspects of XBRL presentation relations are specified by the XBRL technical specification; information about the permissible associations between the categories of report elements as shown by the matrix below are not specified. The model structure rules simply explicitly specify these rules:

		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

3. **Type or class relations** (a.k.a. type-subtype relations): This category of rules specifies allowed subtype relations for each type defined in an XBRL taxonomy. Other terms for this are "is-a" relations or "general-special" relations or "wider-narrower" relations. An example would be a type-subtype rule that specifies that "Accounts Payable" is a sub type of the "Current Liabilities" type. This prevents the inadvertent use of "Accounts Payable" as a part of "Noncurrent Liabilities" or "Equity", etc.
4. **Fundamental accounting concepts:** This category of rules specifies information that helps detect common inconsistencies and contradictions within a financial report[80]. Consistency cross checks are created[81] against expectation. There are many examples of the types of errors that have been known to commonly occur[82]. For example, for US GAAP XBRL-based financial reports submitted to the SEC a common error was to use the concept "us-gaap:NoncurrentAssets" to represent information for which the concept "us-gaap:AssetsNoncurrent" should have been used.
5. **Disclosure mechanics:** This category of rules is used to specify the permissible representations of each specific disclosure. For example, the disclosure "Components of Inventories" would be specified to be a "roll up" mathematical relation which uses the concept "us-gaap:InventoryNet" or a permissible alternative to represent that total. Disclosure mechanics rules likewise specify

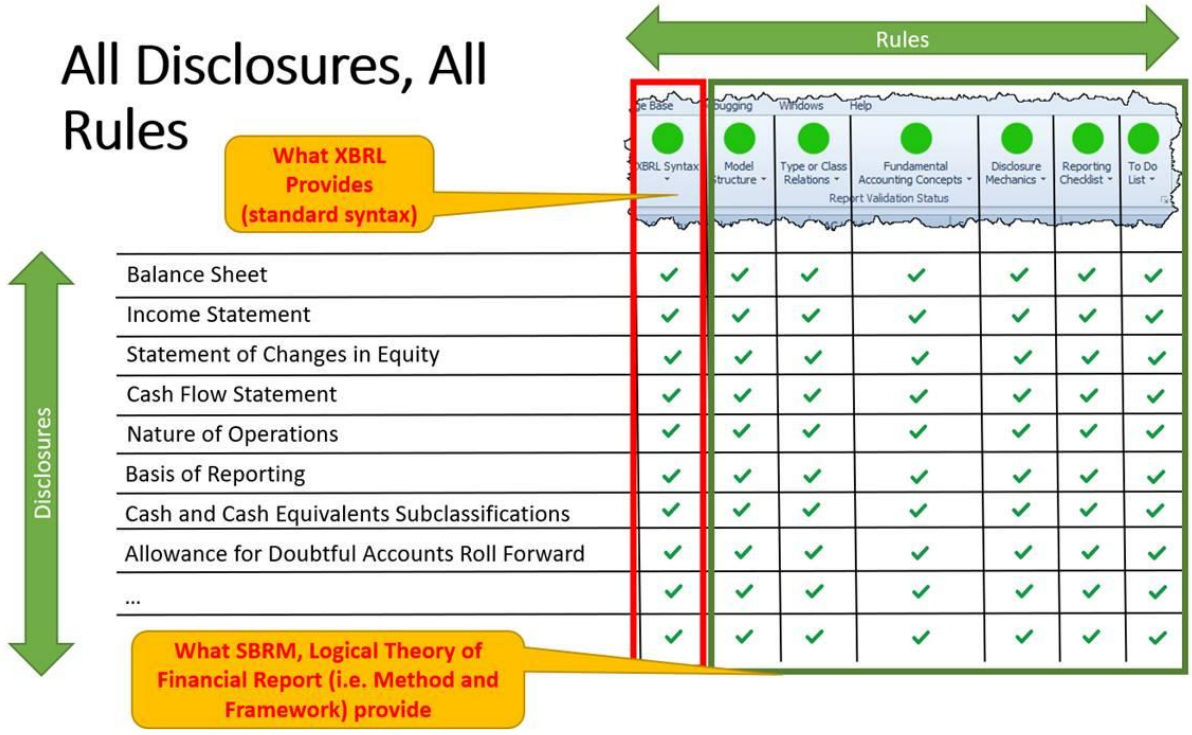
[80] Quarterly XBRL-based Public Company Financial Report Quality Measurement (March 2019), http://xbrl.squarespace.com/journal/2019/3/29/quarterly-xbrl-based-public-company-financial-report-quality.html
[81] Consistency cross check rules, http://xbrlsite.azurewebsites.net/2019/Library/Signals_2019-03-31.jpg
[82] High-quality examples of errors, http://xbrl.squarespace.com/journal/2017/4/29/high-quality-examples-of-errors-in-xbrl-based-financial-repo.html

that the concept “us-gaap:ScheduleOfInventoriesTextBlock” should be used to represent the Level 3 disclosure text block disclosure.

6. **Reporting checklist:** This category of rules is used to specify the permissible sets of disclosures that are required to exist within a financial report. For example, the fact that a balance sheet is always required to be included can be specified, as would be the case for an income statement, statement of cash flow, or statement of changes in equity. The fact that a combined statement of comprehensive income and income could be used as an alternative can be specified. Finally, if a specific line item such as “Inventories” is provided on the balance sheet, the fact that an inventories policy and inventories disclosure must be provided can be specified.
7. **To do list:** This category of rules is for cases where either (a) a rule CANNOT be specified in machine-readable terms because the rules language used is not expressive enough to represent the rule or (b) a rule COULD have been represented but it simply HAS NOT been represented and therefore manual work is necessary to verify report logic that could have been automated.

Complete

And so, XBRL Syntax validation provides only a small subset of what can be verified to be correctly represented within an XBRL-based financial report. Categories 2 through 6 must either be (a) also represented using machine-readable rules and verified using automated processes or (b) verified using manual processes which are less reliable and therefore more prone to error.



Further, if 100% of the rule categories are specified for 100% of the disclosures that exist within an XBRL-based report; then a control mechanism is provided to verify that the financial information

conveyed within an XBRL-based report is consistent with specified statutory and regulatory rules and other structural, mechanical, mathematical, and logical rules.

Additional columns of rules can be added, but no columns can be removed. For example, if one desired to add the XBRL US "Data Quality Checks", then a new column is simply added. Want to add a "spell checker"? That can be added also.

If such a control mechanism is provided; then as pointed out in *Understanding Digital*[83], accounting, reporting, auditing, and analysis tasks and processes can be automated to the degree that such rules exist to enable such automation. Humans (a) deal with exceptions and (b) manual checks. Further, when such processes leverage Lean Six Sigma philosophies and techniques[84], financial report quality can be controlled (as contrast to making mistakes and then spending hours and hours of human effort to detect and correct errors).

The best way to understand the need for this method and the process control mechanisms that it provides is to understand the impediments to creating a properly functioning logical system. We do that next by looking at the impediments to properly functioning logical systems.

Process Automation

Because the method is through and complete, processes can be effectively automated. How exactly can you be sure your financial report is a true and fair representation of the financial position and financial performance of your entity without testing it to be sure the report is working effectively? If you cannot measure it, you cannot control it.

Risk Reduction

Process automation and automated verification reduces the risk of noncompliance. Hope and chance are not good strategies for complying with statutory and regulatory reporting rules.

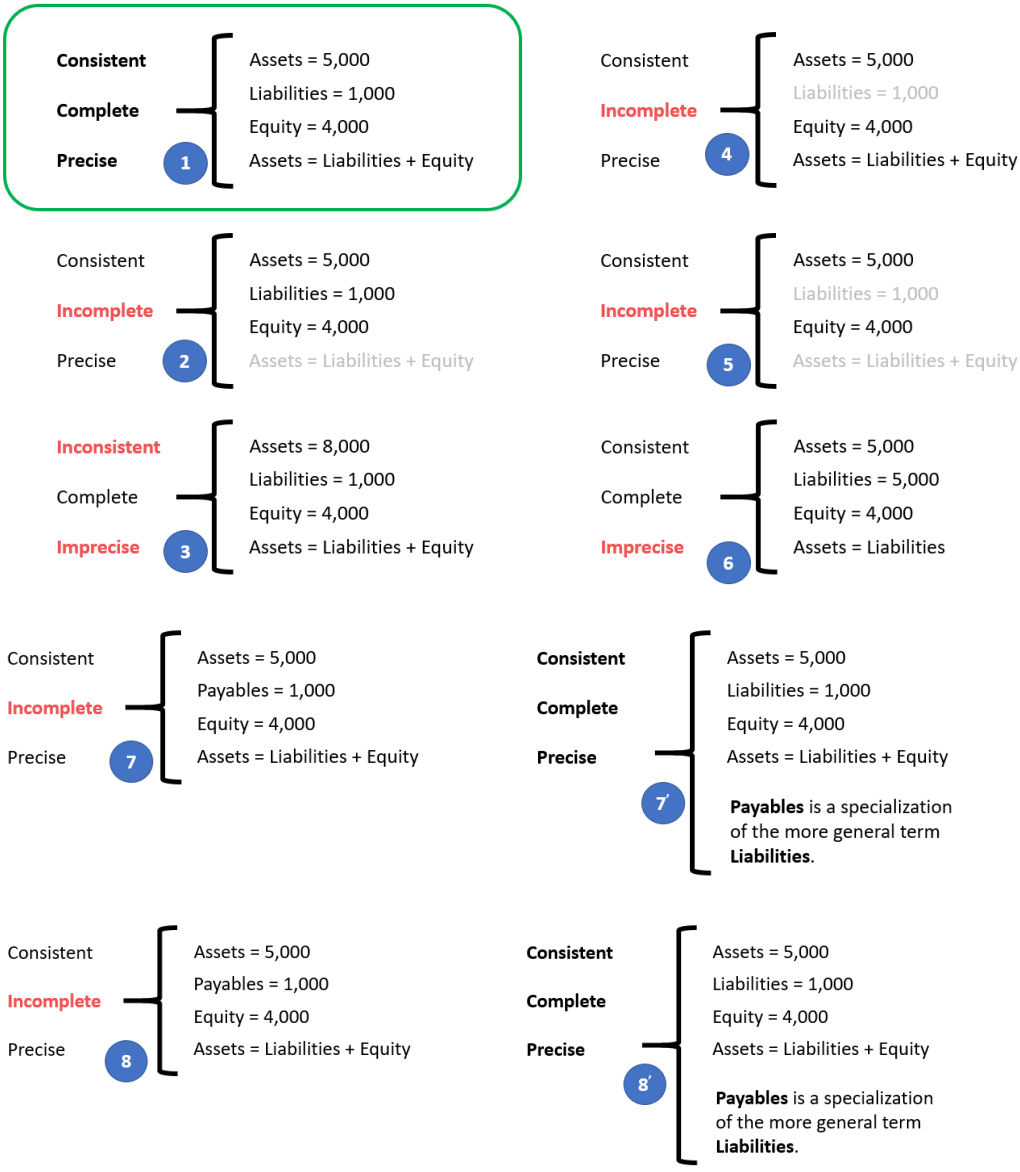
Social Cooperation and Benefit

Being able to effectively exchange information between processes which enables the automation of those processes provides social benefit. Among those benefits are cost reduction, process quality improvement, ability to provide new products/services, and improved functioning of capital markets resulting from these process improvements.

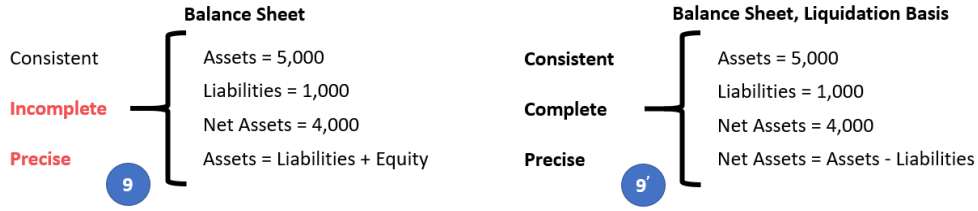
[83] Understanding Digital, http://xbrlsite.azurewebsites.net/2020/Library/UnderstandingDigital.pdf
[84] Lean Six Sigma, http://www.xbrlsite.com/mastering/Part01_Chapter02.K_LeanSixSigma.pdf

Understanding Errors that Can Occur which Method Detects and Prevents (Comparison of States)

The following is a comparison of 9 states of the same simple financial report logical system, the accounting equation[85]. The point of using such a simple financial report logical system is to explain specific things that can go wrong so that a reader can understand why each of the categories of rules are necessary. Here is a summary of all nine states:



[85] Accounting equation, http://xbrlsite.azurewebsites.net/2020/master/ae/index.html



In the following sections I want to make some adjustments to the logical system which make the logical system either inconsistent, incomplete, or imprecise and explain why the system is then not a properly functioning logical system. I made videos that explain each of these impediments to a properly functioning logical system which you can see in this video playlist, *Understanding the Financial Report Logical System*[86].

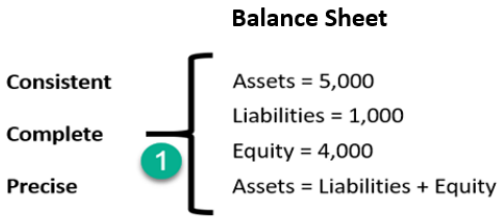
Before we get to the improperly functioning logical systems, let's take one final look at the properly functioning logical system so that you can use that as a baseline for comparing and contrasting the properly functioning and improperly functioning logical systems so that you can understand they sorts of errors that can occur.

State 1: Properly Functioning Logical System

For completeness, I want to start by mentioning again our properly functioning logical system which is consistent, complete, and precise. It can be helpful to contrast other states to this state to understand the difference between properly functioning logical systems and improperly functioning systems.

Balance Sheet [Abstract]	Period [Axis]
	2020-12-31
Balance Sheet [Abstract]	
Assets	5,000
Liabilities	1,000
Equity	4,000

Result	Rule
Pass	$Assets = $Liabilities + $Equity



Again, this is considered a properly functioning logical system because (a) all the statements within the system are **consistent**; (b) the set of statements that describe the system is **complete**; and (c) the information conveyed by the system is **precise** in its representation of reality. Further, we are formally declaring this "reality"[87] to be our base understanding.

Also, we need to be explicit. We defined three terms "Assets", "Liabilities", and "Equity".

[86] Understanding the Financial Report Logical System, https://www.youtube.com/playlist?list=PLqMZRUzQ64B7EWamzDP-WaYbS_W0RL9nt
[87] YouTube, *Reality*, https://youtu.be/eq2Jw6waaCI

Now, you may know what those three terms are; but a computer does not. You have to define what you work with relative to something that you know. Imagine our system defines four terms, "fac:Assets", "fac:Liabilities", "fac:Equity", and "fac:LiabilitiesAndEquity"[88]. You understand your system but you have to map every external system into your system[89]. Your internal system understands more that the accounting equation system (i.e. you have LiabilitiesAndEquity). You have to be able to compute that value based on some other system's information[90]. It is perfectly reasonable for our system to create a concept LiabilitiesAndEquity and compute that value even though the accounting equation logical system does not have that explicit value.

The point is that different economic entities have different models; but all models of a financial reporting scheme are reconcilable from/to one another in some manner[91].

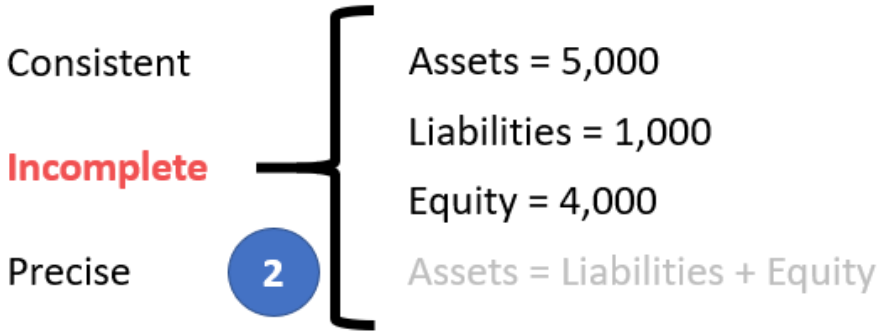
State 2: Incomplete Coverage by Rules

The logical system #2 below is intended to show exactly the same information as our #1 properly functioning logical system, except that #2 leaves out the rule "Assets = Liabilities and Equity" which is showed as grayed out (i.e. because it is assumed to be missing from the logical system.

Coverage is a measure of how well you do or can represent a domain of information within a logical system. "Do" is about using the tools you have correctly and effectively. "Can" is about the capabilities of the tools you are using to represent the rule.

For example, if your logical system neglects to include the rule "Assets = Liabilities + Equity" or if your tools don't provide the capabilities to allow you to represent that rule; then there is the possibility that the facts being represented to be represented incorrectly and the system will not detect the inconsistency. As such, that logical system has **incomplete coverage**.

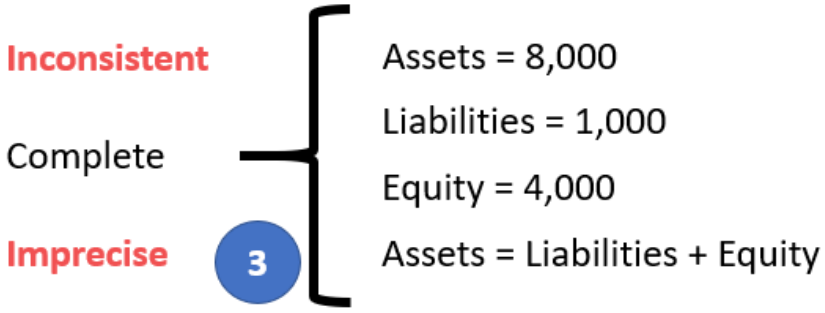
[88] Fundamental accounting concepts, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac.xsd
[89] Mapping from accounting equation to fundamental accounting concepts in our system, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml
[90] XBRL Formula to derive the value for LiabilitiesAndEquity, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-ImputeRule-LiabilitiesAndEquity-formula.xml
[91] Charles Hoffman, CPA, *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*, http://xbrl.squarespace.com/journal/2019/12/30/special-theory-of-machine-based-automated-communication-of-s.html



While this specific state #2 does have the Assets, Liabilities, and Equity facts consistent with the absent rule; the system is still incomplete because the coverage can be improved by adding the missing rule. If that missing rule is added, then the logical system can be considered complete again.

State 3: Inconsistent and Imprecise

All the statements in the system must be consistent for the logical system to be considered properly functioning. If statements are inconsistent, the logical system is not is not properly functioning. In this system #3, the values for Assets, Liabilities, and Equity are inconsistent with the rule "Assets = Liabilities + Equity". From looking at the information provided, it is impossible to know exactly which of the three facts are incorrect; it is only possible to understand that the statements made within the logical system is inconsistent. It could be the case that the rule is incorrect.

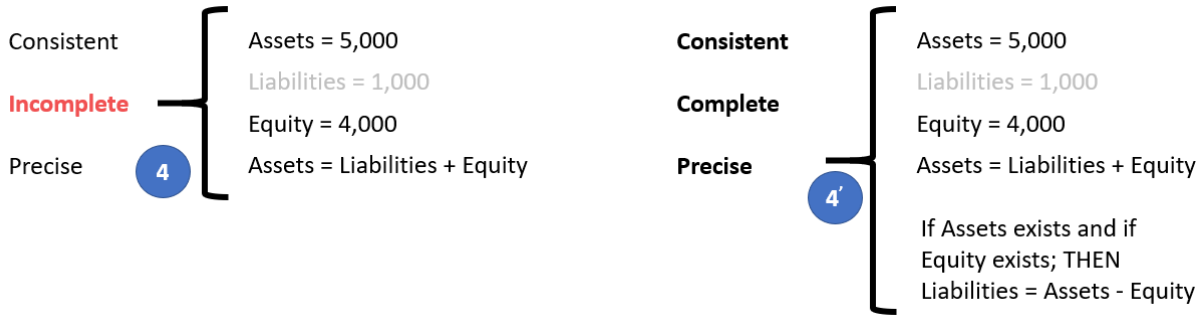


However, given that we know from state #1 that the value for Assets is 5,000 and not 8,000; the facts in this system is imprecise because the fact for Assets does not reflect reality.

State 4: Unreported Facts

In state #4, the situation is that the economic entity representing information in their report neglected to include the fact for Liabilities. Whether it is the case that a fact can, or cannot, be left unreported is a decision that can be made by the stakeholders of the system.

If it is the case that it is decided that the fact "Liabilities" can be omitted if both Assets and Equity are reported; then you must provide a rule to derive the value of Liabilities when that fact is not reported. Below you see that the system has been adjusted in state #4' to add the rule "IF Assets exists and if Equity exists; THEN Liabilities = Assets - Equity"[92]. *(NOTE that this rule should actually be "IF Assets exists and if Equity exists and if not(exists) Liabilities; THEN Liabilities = Assets – Equity")*



If it were likewise true that either Assets[93] or Equity[94] could also be left unreported, similarly derivation rules could be created for each of those facts. Note that XBRL Formula chaining[95] can be used to physically derive unreported facts if any one of these three facts remain unreported. Note that it is impossible to derive missing information if any two of the facts remain unreported. Adding the derivation rule makes the system complete.

Allowing certain line items of a report to go unreported specifies the need to create rules to derive missing information. Or saying this another way, omitting the possibility of unreported facts negates the need for creating derivation rules.

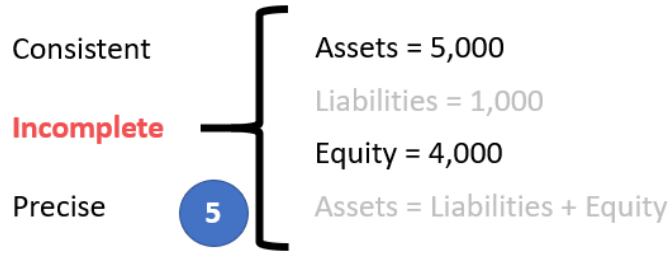
A second downside of allowing unreported facts is that you lose the parity check or cross check if facts can go unreported. Said another way, it would be considered best practice to not leave important high-level financial report line items to go unreported.

State 5: Incomplete

Similar to state #4, in state #5 the logical system is incomplete because both (a) the fact Liabilities is unreported and also (b) the consistency rule "Assets = Liabilities + Equity" is missing

[92] Here is the impute or derivation rule that would be added to the accounting equation logical system for this situation, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-1-Code-BS-Impute-01-formula.xml
[93] XBRL Formula rule for deriving Assets, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-3-Code-BS-Impute-03-formula.xml
[94] XBRL Formula rule for deriving Equity, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-2-Code-BS-Impute-02-formula.xml
[95] Deriving Facts Using XBRL Formula Chaining (Example), http://xbrl.squarespace.com/journal/2019/4/24/deriving-information-using-xbrl-formula-chaining-example.html

from the logical system. Because both a fact and the rule are missing from the logical system, it would be impossible to deduce the value of Liabilities in this case. There is not enough information in the logical system to allow Liabilities to be derived. At a minimum, a consistency crosscheck rule[96] plus the derivation rule to impute Liabilities[97] would be necessary.

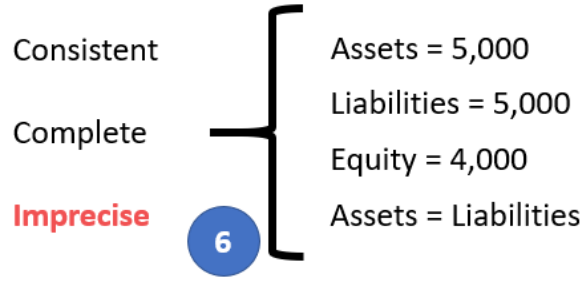


Again, consistent with state #4; Assets and Equity would require similar rules and there is no parity check of reported information.

State 6: Imprecise

A logical system is a true and fair representation of some agreed upon realism. **Precision** is a measure of how precisely you do or can represent the information of a domain within a logical theory. The reality that we formalized in state #1 indicates that "Assets = Liabilities + Equity". Yet, in the state #6 example, the rule "Assets = Liabilities" was provided. Further, the values of Assets and Liabilities are, in fact, consistent with the rule that has been provided.

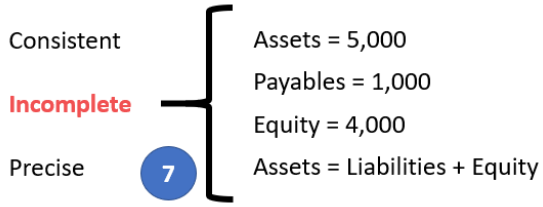
Remember that in state #1 we formalized our truth to be that "Assets = Liabilities + Equity". As such, this logical system can be described as being imprecise. To make this logical system precise, all that needs to be done is to fix the rule.

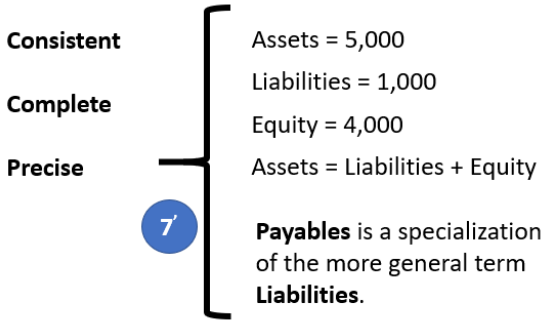


[96] XBRL Formula consistency crosscheck rule Assets = Liabilities + Equity, http://xbrlsite.azurewebsites.net/2020/core/master-ae/Consistency-5-Code-BS01-formula.xml
[97] XBRL Formula derivation rule to impute Liabilities, http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-1-Code-BS-Impute-01-formula.xml

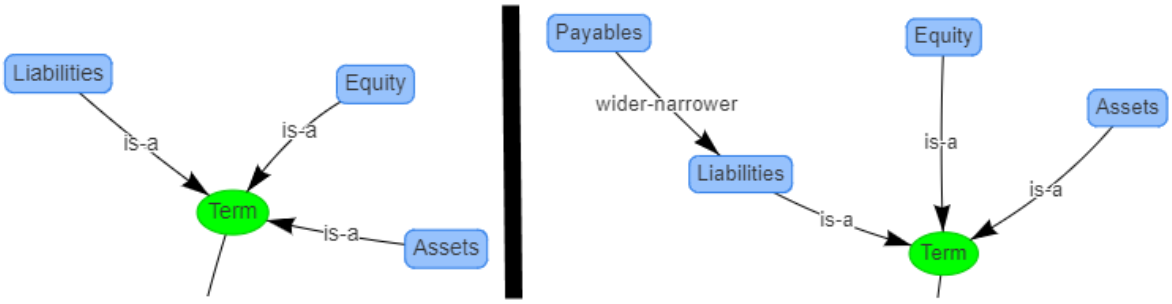
State 7: Extension Concept

In state #7 on the left, what we are trying to convey is that the economic entity reported the fact for Liabilities using the extension concept "Payables" that it had created. If a fact is represented using an extension concept created by a reporting entity; then a "general-special" or "wider-narrower" or "class-equivalentClass" association must be created to indicate to software applications of the relationship so that information can be used correctly. State #7' on the right, the rule "Payables is a specialization of the more general term Liabilities" has been added to the logical system which allows the system to operate effectively[98].





And so, the graphic below shows a fragment of the knowledge graph on page 9 above before and after the information that "Payables is a specialization of the more general term Liabilities," was added. On the left you see State 7, the taxonomy before the information was added and on the right you see "Payable" being added as an extension concept indicating that there is a "wider-narrower" relationship between Payables and Liabilities. Therefore, a machine based process can utilize the information per State 7' because the process understands Liabilities in the taxonomy, understands the "wider-narrower" relationship therefore knowing that "Payables" is a type of Liability.



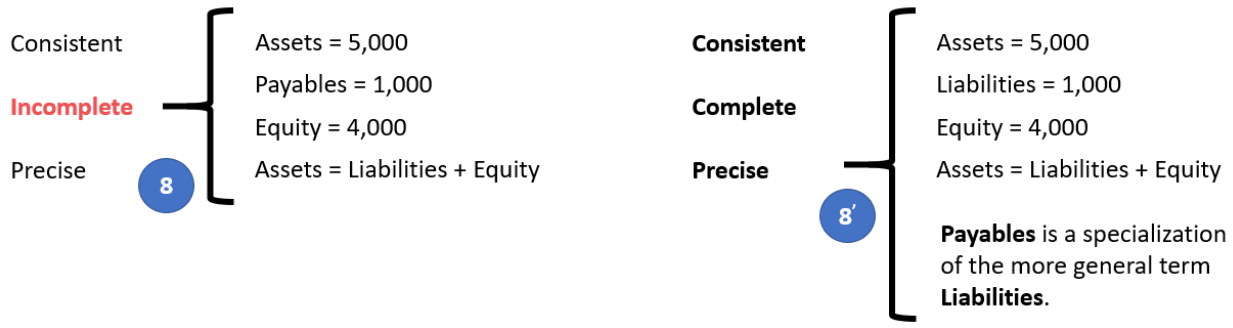
[98] XBRL Definition relations showing example of a mapping rule, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml

State 8: Base Taxonomy Wider/Narrower Concept Use

State #8 on the left below is similar to state #7 in that a different concept is used to report a fact; but while state #7 focuses on using an extension concept; state #8 points out that using a wider or narrower base taxonomy concept gives exactly the same result.

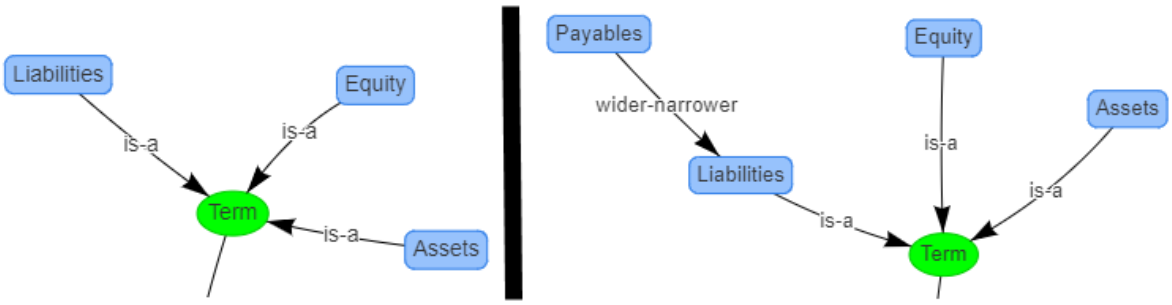
Now, our base state #1 does not have the concept "Payables"; but let's assume for a moment that it does have the concept "Payables". Also suppose that there was no information in the base logical system indicating the relationship between "Payables" and any other concept. If a fact is represented using a BASE TAXONOMY CONCEPT by a reporting entity; then a "general-special" or "wider-narrower" or "class-equivalentClass" association must exist in that base taxonomy to indicate that some concept is a permissible alternative for some other concept.

State #8' on the right adds the rule "Payables is a specialization of the more general term Liabilities"[99].



And so, the graphic below shows a fragment of the knowledge graph on page 9 above before and after the information that "Payables is a specialization of the more general term Liabilities," was added. On the left you see State 8, the base taxonomy before the information was added and on the right you see "Payable" being added as an extension concept indicating that there is a "wider-narrower" relationship between Payables and Liabilities. Therefore, a machine based process can utilize the information per State 8' because the process understands Liabilities in the base taxonomy, understands the "wider-narrower" relationship therefore knowing that "Payables" is a type of Liability.

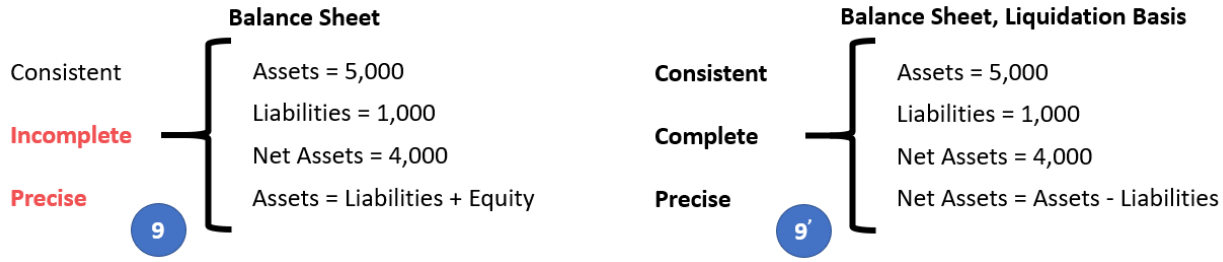
[99] XBRL Definition relations showing example of a mapping rule, http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml



State 9: Defining a Completely New Structure

State #9 below on the left focuses on the structure as contrast all the prior examples which focused on the terms and rules. If a new structure is created, the new structure must be referenced to the base taxonomy and the new structure needs to be explained using machine-readable rules[100]. Even base taxonomy structures need to be defined in order to be referred to[101]. When you say "Balance Sheet" you know what that means. But a machine does not know.

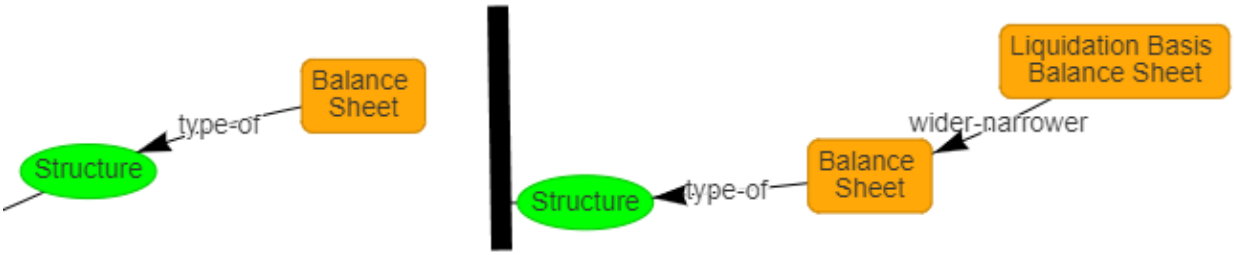
A base taxonomy should (a) provide all necessary structures separately, not intermingle different models in the same set of associations and (b) define what each structure must look like. Remember, computers are like babies and need to be led by the hand in order to understand the details you need them to understand.



Finally, in our case we have only one disclosure, the Balance Sheet. In our case, the Balance Sheet is always required to be reported per this logical system. As such, that rule is stated in a machine-readable reporting checklist[102]. Other logical systems with more disclosures will have more rules relating to when a disclosure is required to be provided in a report.

[100] XBRL Definition relations used to represent structure rules, http://xbrlsite.azurewebsites.net/2020/core/master-ae/dm-1355-rules-def.xml
[101] XBRL taxonomy schema used to define "Balance Sheet", http://xbrlsite.azurewebsites.net/2020/core/master-ae/disclosures.xsd
[102] XBRL Definition relations used to represent a reporting checklist or disclosure rules, http://xbrlsite.azurewebsites.net/2020/core/master-ae/reporting-checklist-rules-def.xml

Similar to how "Payables" was added as an extension of the terms in the logical system; we can extend the structures to include a "Liquidation Basis Balance Sheet" structure which is a specialization of a Balance Sheet:



And such, an automated process will be able to understand the new structure because it is related to an existing structure. Other structures could be added and only identified as a type of structure.

Examining Errors in Actual Submitted Reports

As explained, a very good way to understand how to create reports correctly is to examine errors in reports that others have created. Here are three documents that provide well documented examples of undisputed errors:

- Issues in XBRL-based Digital Financial Reports[103]
- More Issues in XBRL-based Digital Financial Reports[104]
- Accounting errors[105]

Finally, this blog post[106] (see toward the bottom) provides 24 sets of well documented errors in XBRL-based reports.

Note that all of these errors were discovered using automated processes which leverage this method. If reports were verified using this method prior to submitting the report to a regulator, all such errors could have been detected and corrected. The result would be higher quality XBRL-based reports.

[103] Issues in XBRL-based Digital Financial Reports, http://xbrlsite.azurewebsites.net/2020/master/100IssuesOfPublicCompanyReports.pdf
[104] More issues in XBRL-based Digital Financial Reports, http://xbrlsite.azurewebsites.net/2020/master/About50MoreIssuesOfPublicCompanyReports.pdf
[105] Accounting Errors, http://xbrlsite.azurewebsites.net/2018/Library/AccountingErrorsFoundDuringValidation.pdf
[106] High Quality Examples of Errors in XBRL-based Financial Reports, http://xbrl.squarespace.com/journal/2017/4/29/high-quality-examples-of-errors-in-xbrl-based-financial-repo.html

Best Practices

A **best practice** is a method or technique that has been generally accepted as superior to any other known alternatives because it produces results that are superior to those achieved by other means or because it has become a standard way of doing things.

Best practices (or good practices) are techniques that have produced outstanding results in other situations, inside or outside of a particular organization and which can be validated, codified, and shared with others and recommended as models to follow[107].

To understand how to represent XBRL-based financial reports at Level 5 please start with *Essentials of XBRL-based Digital Financial Reporting*[108].

Conclusion

As is said, "If you cannot measure it, you cannot control it." The method we are providing an overview of in this document provides the measurements necessary to create repeatable, reliable process control mechanism which yields high-quality XBRL-based financial reports.

[107] European Guide to good Practice in Knowledge Management - Part 5: KM Terminology, page 3 (PDF page 9), http://arielsheen.com/wp-content/uploads/2019/10/CEN-CWA14924-05-2004-Mar.pdf#page=9
[108] Charles Hoffman, CPA, *Essentials of XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2021/essentials/EssentialsOfXBRLBasedDigitalFinancialReporting.pdf