Seattle Method

Proven, industrial strength, good practices, standards-based pragmatic approach to creating provably high quality XBRL-based general purpose financial reports explained in simple terms

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

Last Revised – April 4, 2024 (DRAFT)

http://xbrlsite.com/seattlemethod/

"For, if he does not know how to put each thing in its own place, he will find himself in great trouble and confusion as to all his affairs, according to the familiar saying, Where there is not order, there is confusion." *Luca Pacioli*

Executive summary:

- A best practice or good 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.
- Financial reports are fundamentally based on the double entry accounting model, the accounting equation, and are intentionally designed to have innate characteristics such as mathematical interrelationships to achieve the notion of articulation which is where one report element is intentionally defined on the bases of other elements in order to achieve the interconnectedness of the four primary financial statements.
- Financial reports are knowledge graphs.
- XBRL-based digital financial reports can be proven to be properly functioning logical systems that are consistent, precise, and complete using automated machine-based processes that take into account the inherent variability of financial reports.
- The focus of the *Seattle Method* is financial reporting using financial reporting schemes such as US GAAP, IFRS, UK GAAP, and other schemes where the preparer of a financial report is permitted to modify the report model. Because modification of the report model is allowed, those modifications must be controlled to keep the modifications within permitted boundaries.
- Without control, there can be no automation, no repeatable processes. Rules provide control. Control leads to high quality. High quality leads to effective automation.

Acknowledgements:

While I did physically create the information in this resource, I could have not done so without the gracious help of a number of people, directly and indirectly, over the years. I see myself as merely a custodian of this important information, nurturing it along for the benefit of all, condensing countless discussions into something hopefully useful for the common good.

I would like to specifically thank these contributors: Walter Hamscher, PhD, Geoff Shuetrim, PhD; David vun Kannon; Josef Macdonald, CA; Jim Richards; Roger Debreceny; Jeff Naumann, CPA; David Prather, Alan Teixeira, CA; Hugh Wallis; Allyson Ugarte; Colm O hAonghusa; Giancarlo Pellizzari; Yossef Newman, CPA; Rob Blake; Mark Creemers; Marc van Hilvoorde; Herman Fischer; Ignacio Hernandez-Ros; Dean Ritz; Timothy Randle; Cliff Binstock; David Scott Stokes; Masatomo Goto; Paul Warren; Mark Goodhand; Campbell Pryde, CPA; Michele Romanelli; Maciej Piechocki, PhD; Victor Morilla; Mike Rowling; Joe Ryba, CPA; Matthias Brantner; Dennis Knochenwefel; Ghislain Fourny, PhD; Chris Taylor, CPA, Eric Cohen, CPA; Mike Willis, CPA; Louis Matherne, CPA; Hamed Mousavi; Daniel Enrique Montoya Guerrero; Yury Volkovich; Michelle Rojo; Todd Flores; Jason Meyers; Miguel Calejo; Andrew Noble; Tom Egan, CPA; Tom McKinney, CPA; Dudley Gould, ACA; Jaume Mora Pedros; Willi Brammertz; Gregory Sharp. There are others which I probably left off and for this I apologize. I acknowledge and appreciate the thinking others contributed to this endeavor.

http://xbrl.squarespace.com/

The ***Seattle Method***[1] is a proven, industrial strength, good practices, standards-based approach to creating provably high quality XBRL-based general purpose financial reports when a report creator is permitted to modify the report model. The *Seattle Method* provides accountants the freedom and responsibility within a strict framework of a highly developed system of XBRL-based digital financial reporting. The framework offers what can be thought of as "guardrails" or "bumpers" that help steer accountants toward high-quality machine-readable financial reports. Software applications can leverage this framework to create software which is easy for accountants to use and increases financial report quality.

Financial reports are knowledge graphs. In the past, these knowledge graphs have only been readable by humans. Now, using the XBRL global standard, the knowledge graphs can be readable by both humans and machines: financial reports are machine-readable global standard knowledge graphs of XBRL-based information. The only way this "machine-readable financial report" thing can work is if such reports are trustworthy, interpretable, explainable, and preferably the origin or provenance of provided information is known (i.e. WHY you can trust, WHAT are the facts, HOW are you interpreting, WHERE did you get the information you are using, WHO stands behind that information). The knowledge graphs must be of the professional level and provably of high quality. One would expect that an XBRL-based report model and report will be:

- Encode formally in a declarative knowledge representation language such as XBRL.
- Syntactically well-formed for the chosen language, as verified by an appropriate syntax checker or parser. In our case, this would be an XBRL processor that has passed the appropriate XBRL International conformance suite(s).
- Logically consistent, as verified by a language appropriate reasoner or theorem prover. This would include a complete set of rules which exercises that logic.
- Will meet business or appropriate application requirements as demonstrated though excessive testing. For example, information can be effectively extracted from such a report model and report.

Effectively, the *Seattle Method* provides a fixed method for representing (i.e. modeling) financial accounting, reporting, auditing, and analysis experience and information in machine readable form. It starts with terms, sets of relationships, sets of rules that govern the relationships, other rules that describe what is permitted, and how to represent facts within this scheme so that information is understandable to machine-based processes. You can think of the *Seattle Method* as a pattern language with exactly the appropriate level of flexibility in exactly the right areas such that things represented using that pattern language are always "computable" because the foundational "container" of relationships, rules, and facts never

[1] *Seattle Method*, http://xbrlsite.com/seattlemethod/

change. They are just logical patterns entirely known and understood by the method. As such, software can be used to effectively reason over the structures, associations, rules, and facts represented within different models because the sense-making machinery that is "baked-in" to the capabilities of the Seattle Method pattern language. The "fixed" way of defining the patterns provides us with this consistently useful method for defining or exploring complex information logic that always exists within the "guardrails" or "bumpers" provided by definitions of what is permitted and what is not permitted by, say, some specific financial reporting scheme represented using this approach. (To best understand this paradigm, please be sure you are familiar with the *Essence of Accounting*[2].)

This method is grounded in the Venetian Method of double entry bookkeeping. The *Seattle Method* is a rebranding of what I had originally simply referred to as my method[3].

While most medieval practices were found wanting in the industrial age, Venetian bookkeeping is coming into its own. Double entry accounting is proving to be remarkably adaptive to new demands being made upon it by the world. In fact, double entry accounting seems to have been designed for the information age.

Double entry bookkeeping is science. *Mathematics Magazine* published an article written by David Ellerman, *The Mathematics of Double Entry Bookkeeping*[4], where Ellerman points out that double entry accounting is based on well-known mathematics construction from undergraduate algebra.

In her book, *Double Entry: How the Merchants of Venice Created Modern Finance*[5], the author Jane Gleeson-White refers to accounting as our first communications technology. I agree. A general purpose financial report is a high-fidelity, high-resolution, high-quality information exchange mechanism. The report is a compendium of complicated[6] logical information required by statutory requirements and regulatory rules plus whatever management of an economic entity wants to voluntarily disclose. The report represents quantitative and qualitative information about the financial condition and financial performance of an economic entity.

---

[2] Charles Hoffman, CPA, *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf
[3] *Method Overview*, http://accounting.auditchain.finance/framework/MethodOverview.pdf
[4] David Ellerman, Mathematics Magazine, *The Mathematics of Double Entry Bookkeeping*, https://ellerman.org/wp-content/uploads/2012/12/DEB-Math-Mag.CV_.pdf
[5] Amazon.com, *Double Entry: How the Merchants of Venice Created Modern Finance*, https://www.amazon.com/gp/product/B007Q6XKA8/
[6] Cynefin Framework, http://xbrl.squarespace.com/journal/2021/3/21/cynefin-framework.html

The approach used to create this method is that of "standing on the shoulders of giants[7]" or "discovering truth by building on previous discoveries." To develop this method, I leverage many other commonly accepted theories, methods, practices, and ideas. What this method contributes is pulling the necessary theories, methods, practices, and ideas together into one combined system.

Creating something that is simple takes conscious effort and is hard work. Anyone can create something that is sophisticated and complex. It is much harder to create something that is sophisticated and simple. Simple is not the same thing as simplistic. "Simple" is not about doing simple things. Simple is the ultimate sophistication. Simple is elegant. Simplicity is "dumbing down" a problem to make the problem easier to solve. Simple is about beating down complexity in order to make something simple, graceful, and elegant; to make sophisticated things simple to use rather than complex to use. This method strives to make it possible to create software that is simple and elegant.

The *Law of Conservation of Complexity*[8] states: "Every application has an inherent amount of irreducible complexity. The only question is: Who will have to deal with it-the user, the application developer, or the platform developer?"

Irreducible Complexity[9] is explained as follows: A single system which is composed of several interacting parts that contribute to the basic function, and where the removal of any one of the parts causes the system to effectively cease functioning.

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

The notions of conservation of complexity and irreducible complexity are important to understand in order to create easy to use software that gets the right job done. This method and clever programming techniques and ideas will yield extraordinary software that is easy to use and can likely be considered elegant.

[7] Wikipedia, *Standing on the Shoulders of Giants*, https://en.wikipedia.org/wiki/Standing_on_the_shoulders_of_giants
[8] *Law of Conservation of Complexity*, http://www.nomodes.com/Larry_Tesler_Consulting/Complexity_Law.html
[9] *Irreducible Complexity*, https://www.gotquestions.org/irreducible-complexity.html

It is assumed that a reader of this document is familiar with the basics of financial reporting and basic mathematics. If you are not, I would recommend that you read *Essence of Accounting*[10] prior to reading this document.

Semantic Hygiene

As pointed out by George Doris in the paper, KORZYBSKI AND GENERAL SEMANTICS*[11], communication is hard. Communication is the fight against confusion. Whether it is the "speaker" versus "listener" or "writer" versus "reader" or "information bearer" versus "information receiver"; there is a process and the process tends to be a struggle. But effective communication based on a common, shared understanding has its reward.

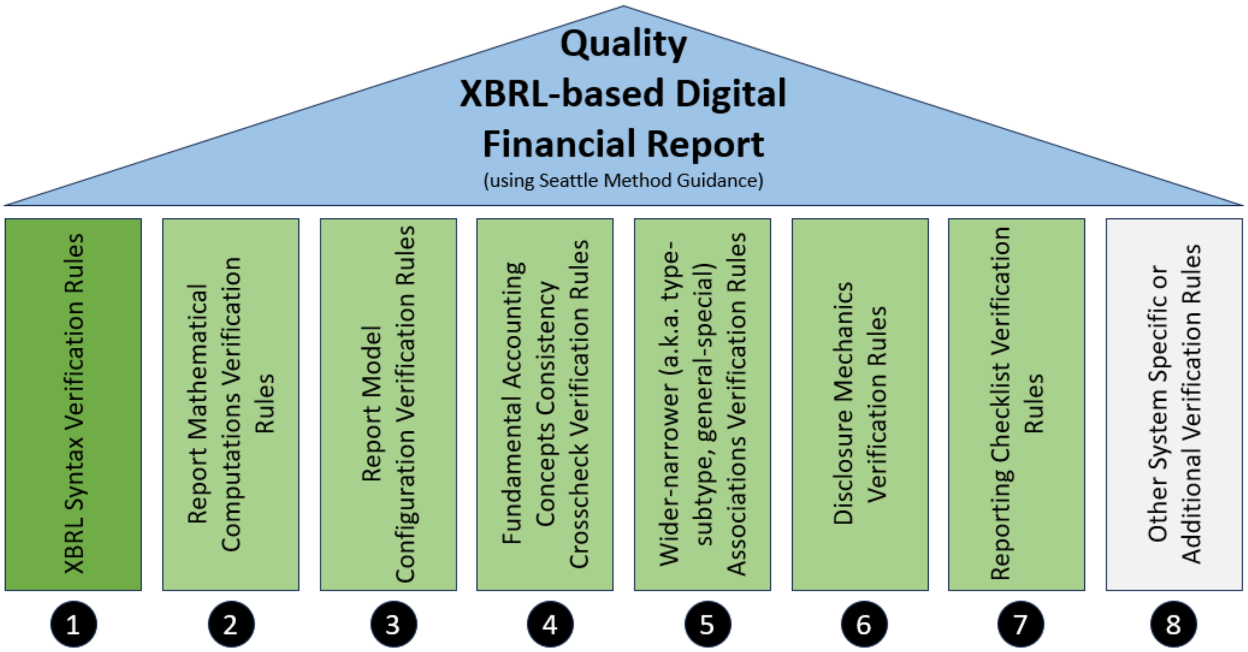
It takes deliberate conscious effort to maintain semantic hygiene.

Pillars of Trustworthiness and Quality

For a machine-readable XBRL-based financial statement to be useful, that financial statement needs to be trustworthy. The guidance provided by the Seattle Method enables accountants and others to use a proven, industrial strength, good practices based, standards-based pragmatic approach to creating provably high quality XBRL-based general purpose financial reports when report models are permitted to be modified. When a financial statement model can be modified (a.k.a. customized), the "wild behavior" of accountants creating such financial statements must be eliminated, keeping report models the accountants create within permitted boundaries. Both US GAAP and IFRS financial reporting permits the customization of economic entity report models.

A trustworthy machine-readable general purpose financial report is quite useful. To be trustworthy, you don't want any blind spots. Further, such machine-readable financial statements can only be trustworthy to the extent that rules are provided that then can be used to provide that trustworthiness. The pillars of trustworthiness provided by the Seattle Method are shown below:

[10] Charles Hoffman, CPA, *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf
[11] Institute of General Semantics, *KORZYBSKI AND GENERAL SEMANTICS**, https://generalsemantics.org/resources/documents/korzybski-and-general-semantics-by-george-doris.pdf



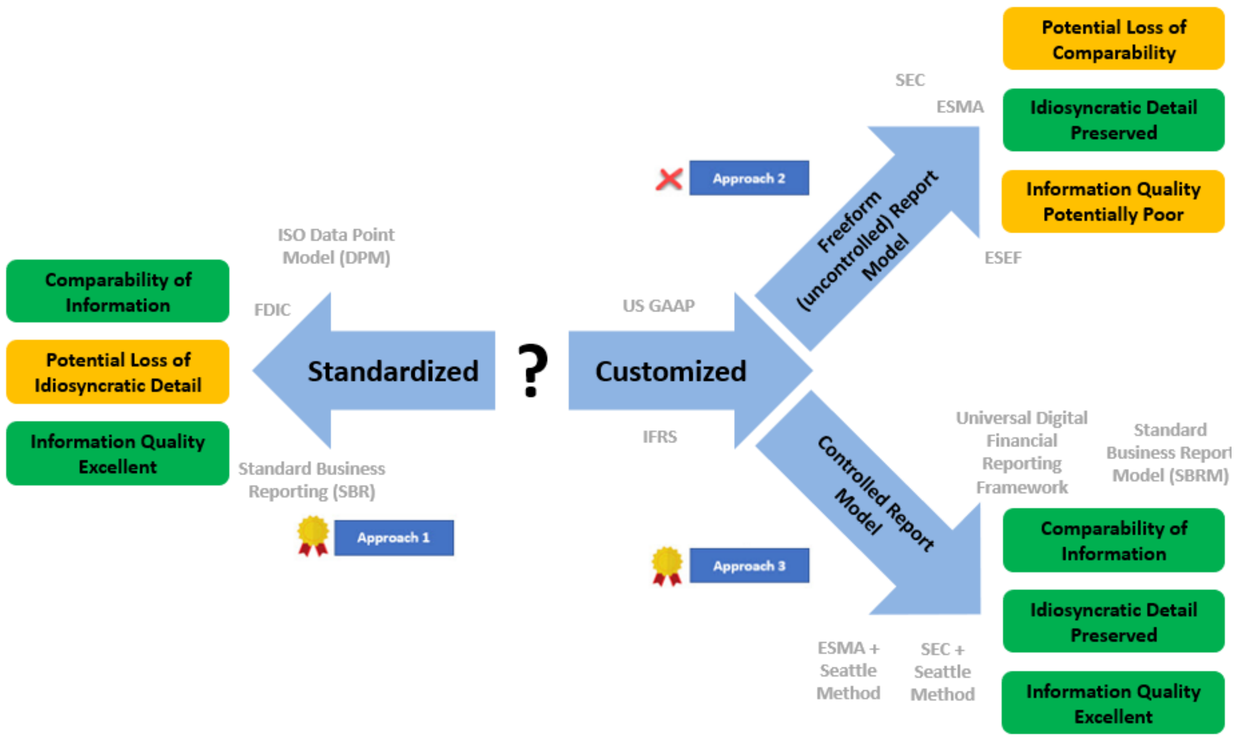
That is what the Seattle Method guidance suggests. Every one of those pillars of trustworthiness or pillars of quality is necessary. Remove a column and things can go wrong. Don't measure with rules, then you have a blind spot in your system.

Deductive logic is precise because it provides certainty; guaranteed. The machine-readable deductive rules provide a "template" for what a perfect/precise XBRL-based financial report looks like. It is to the extent that these rules are provided; it is to that extent that reports can be considered trustworthy. **Valid** reports (consistent with all the specified rules) that are also **sound** (a.k.a. precise, precisely follow real-world financial reporting rules and other logic); it is to that extent that intelligent software agents making use of such information can do so effectively. Full stop. No magic; just good engineering.

Standard vs Customized Reporting Approaches

The paper *Critical Reflection on XBRL: A "Customisable Standard" for Financial Reporting?*[12], breaks reporting into two approaches: ***standardized*** reporting and ***customized*** reporting. I modified this breakdown slightly breaking customized reporting into two distinct approaches, "freeform customization" and "controlled customization". I then reflected the three approaches in the following graphic inspired by the graphic in the referenced paper[13]:

[12] Reporting Approaches + XBRL Approaches + Implementation Approaches, http://xbrl.squarespace.com/journal/2021/12/30/reporting-approaches-xbrl-approaches-implementation-approach.html
[13] Taxonomy creation approaches, http://xbrlsite.azurewebsites.net/2022/library/TaxonomyApproachesSeattleMethod.jpg



This yields three distinct modeling approaches:

- **Standard form model**: No modifications are allowed to the report model.
- **Freeform, Uncontrolled model**: Modifications are permitted to report model, but those modifications are not controlled in any way. As such there is no differentiation between permitted and unpermitted modifications to the model.
- **Controlled model**: Modifications are permitted to report model and a mechanism is provided to control report model modifications; permitted and unpermitted report model modifications are clearly delineated and control mechanisms keep report model modification within permitted boundaries.

Effectively, uncontrolled customization of report models simply will not work. Report customization must be controlled to keep reporting economic entities within the boundaries of what is permitted. Effectively, "guardrails" need to be in place. Control provides those guardrails.

The Seattle Method provides those guardrails, that control.

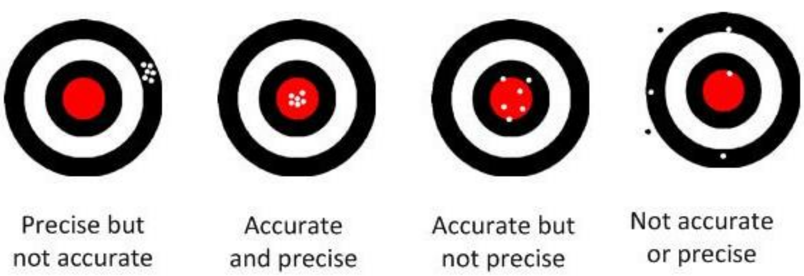
Control

When a report model can be modified, the "wild behavior" of accountants creating reports must be eliminated, keeping report models within permitted boundaries. An XBRL taxonomy which is used to represent a report model in machine readable form and serves multiple purposes:

- **Description**: It is a description of a report model (specification of what is permitted); created by standards setter or regulator or anyone else specifying a report.
- **Construction**: It is a guide to the creation of a report based on that report model description whereby a human can be assisted by software applications utilizing that machine readable description.
- **Verify**: The actual report constructed can be verified against the description assisted by software applications utilizing that machine readable description.
- **Information extraction**: Information can be effectively extracted from machine readable reports and report models assisted by software utilizing that machine readable description.

Note that the machine readable version of the report model description and report can be automatically converted from the machine readable format to a human readable formation using automated processes.

A machine readable representation of a financial reporting scheme in an XBRL taxonomy must be *clear*, *complete*, and reflect accounting and reporting rules *precisely* and *accurately*[14].



Understanding the Problem

The following problem description was inspired by a similar sort of description by Harry S. Delugach, Associate Professor of Computer Science, in a presentation, *Common Logic Standards*

[14] What is Accuracy?, https://www.adamequipment.com/aeblog/what-is-accuracy

Development, (page 7). Fundamentally, a general purpose financial statement serves this purpose:

> Two economic entities, A and B, each have information about their financial position and financial performance. They must communicate their information to an investor who is making investment decisions which will make use of the combined information so as to draw some conclusions. All three parties (economic entity A, economic entity B, investor) are using a **common set of basic logical principles** (facts, statements, deductive reasoning, etc.), **common financial reporting standard terms and associations between terms** (terms, associations, structures, rules for a reporting scheme US GAAP, IFRS, IPSAS, etc.), and a **common world view** so they should be able to communicate this information fully, so that any inferences which, say, the investor draws from economic entity A's information should also be derivable by economic entity A itself using common basic logical principles, common financial reporting standards (terms, associations, structures, assertions), and common world view; and vice versa; and similarly for the investor and economic entity B.

This problem has been effectively solved for hundreds of years via the use of paper-based and human readable general-purpose financial statements. Today there is a new opportunity. That new opportunity is to automate this process using machine-readable financial information[15].

To be crystal clear, financial statements I am describing are not, should not, and need not be forms. Rather, financial reporting schemes used to create the financial statements I am describing intentionally allow variability in how economic entities provide the quantitative and qualitative information about the economic entity. Report creators are permitted to "reshape" or "alter" or make other such modifications to a report model within a specific set of well-established and understood boundaries.

This specific use case is clearly articulated in the conceptual frameworks of both US GAAP[16] and IFRS[17] and really cannot be disputed. Those less familiar with financial reporting may find my exploration of FASB's SFAC 6 *Elements of Financial Statements*[18] helpful.

[15] Charles Hoffman, CPA, *Computational Professional Services*, http://xbrlsite.azurewebsites.net/2020/library/ComputationalProfessionalServices.pdf
[16] Financial Accounting Standards Board (FASB), *Statement of Financial Reporting Concepts No. 6, Elements of a Financial Statement*, https://www.fasb.org/pdf/con6.pdf
[17] International Accounting Standards Board (IASB), *Conceptual Framework for Financial Reporting*, March 2018, https://www.ifrs.org/issued-standards/list-of-standards/conceptual-framework/
[18] Charles Hoffman, CPA, *Impediments to Creating Properly Functioning XBRL-based Reports (SFAC 6)*, http://xbrlsite.azurewebsites.net/2020/core/master-sfac6/Documentation.pdf

Finally, it is worth pointing out that financial reporting schemes have five things in common that can be leveraged in the communication of financial statement information and are unique to financial reporting schemes:

- *First*, at the foundation of every financial reporting scheme complies with the double-entry bookkeeping model[19]. Simply stated, that model is: **DEBITS = CREDITS**.
- *Second*, building on the double-entry bookkeeping model is the accounting equation[20] which is: **Assets = Liabilities + Equity**. There are other forms of the accounting equation.
- *Third*, every financial reporting scheme defines a core set of interrelated elements[21] of a financial statement that are fundamentally grounded in some form of the accounting equation. For example, the Financial Accounting Standards Board (FASB) defines these ten elements of a financial statement in SFAC 6[22]; Assets, Liabilities, Equity, Comprehensive Income, Investments by Owners, Distributions to Owners, Revenues, Expenses, Gains, Losses. Then, additional report elements are defined based on that core set.
- *Fourth*, every financial reporting scheme has what is called "articulation". Articulation[23] is the notion that the elements of a financial statement are intentionally interrelated and therefore depend on one another. And so, the four core statements; balance sheet, income statement, changes in equity, and cash flow statement; are all intentionally mathematically interrelated. Articulation is explained very methodically by the FASB in SFAC 6[24].
- *Fifth*, every financial report model has inherent variability that is the result of explicitly allowing intermediate components of a financial report (i.e. subtotals) to be combined in appropriate but perhaps different ways depending on the needs of the reporting economic entity. Again, this is explained in detail within SFAC 6[25].

These five special characteristics of a financial reporting scheme and therefore of a financial statement created using such a financial reporting scheme offers benefits above and beyond

[19] David P. Ellerman, *The Mathematics of Double Entry Bookkeeping*, Mathematics Magazine, http://www.ellerman.org/wp-content/uploads/2012/12/DEB-Math-Mag.CV_.pdf
[20] Wikipedia, *Accounting Equation*, https://en.wikipedia.org/wiki/Accounting_equation
[21] *Comparison of Elements of Financial Statements*, http://xbrlsite.azurewebsites.net/2020/master/ElementsOfFinancialStatements.pdf
[22] Financial Accounting Standards Board (FASB), Statement of Financial Reporting Concepts No. 6, Elements of a Financial Statement, page 23, https://www.fasb.org/pdf/con6.pdf
[23] Articulation, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-implementation/PROOF_Articulation.jpg
[24] ibid, page 21 – 22, "Interrelation of Elements-Articulation"
[25] Ibid, page 47, paragraph 77.

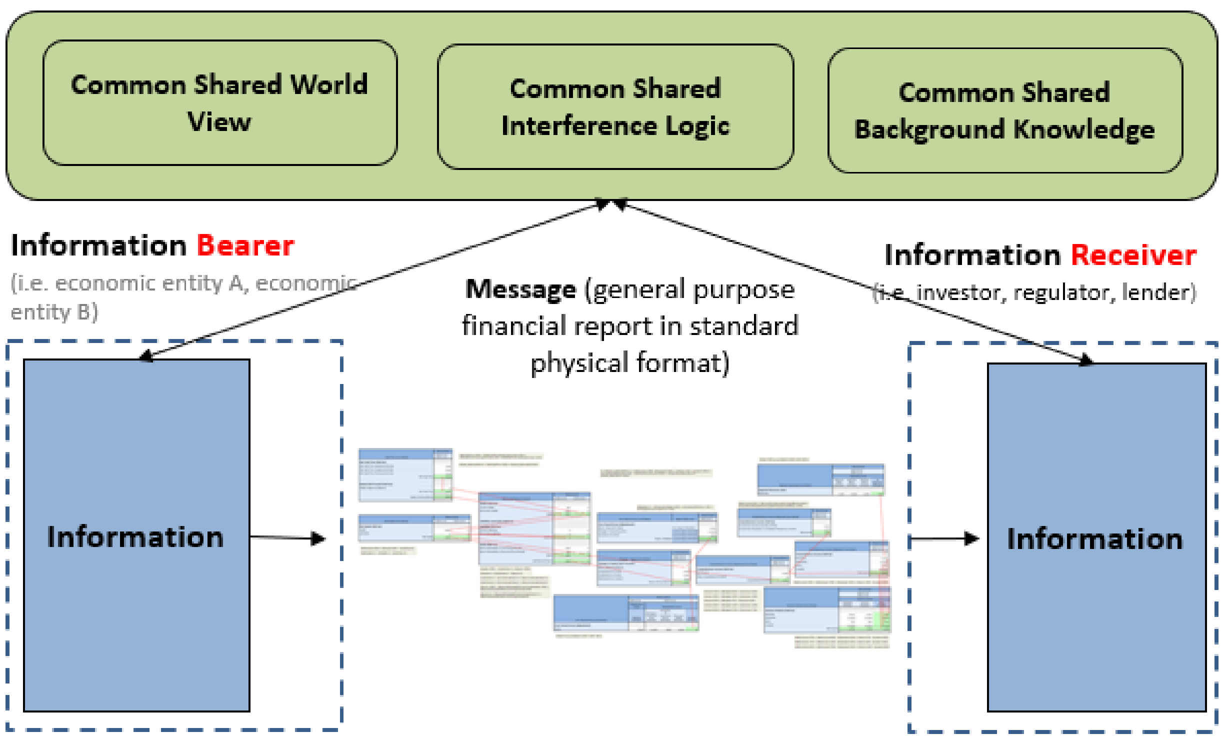
the general communication of words and numbers. This method leverages these special characteristics of financial reporting schemes.

As such, this method focuses on the special case of communication of financial statement information.

Graphic of Problem Statement

In their paper, *Towards a Theory of Semantic Communication*[26] , Jie Bao et. al. provides a visual description of the communications of information. In the diagram, Bao. Et.Al. assign variables and work through the mathematics of the problem of exchanging information from a sender to a receiver successfully.

Inspired by Jie Boa et. al.; I created a modified visual description of the communication of financial information which I provide below:



The general idea of my visual image is the same as Jie Boa et. al., however there are some specific differences that are intentional and make the communication of financial information easier.

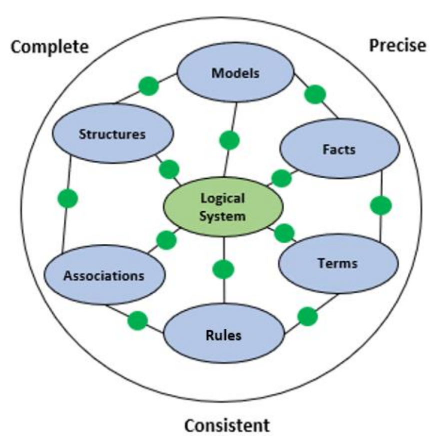
[26] Jie Bao et.al., *Towards a Theory of Semantic Communication*, page 5, Fig. 2. Semantic Information Source and Destination, https://pdfs.semanticscholar.org/fa34/3407847eea1f7e8bb8d3d7489b6945e2b0b2.pdf

First, Jie Boa et. al. state that the world view of the information sender (W_s) and receiver (W_r) are perhaps different and then reconciled. This is similar for the inference procedure (I_s, I_r) and background knowledge (K_s, K_r).

What I am trying to communicate is the notion that as many differences as possible would be eliminated from the communications problem. As such, the "World View", the "Inference Logic" and as much of the "Background Knowledge" as possible would be agreed to in advance of any financial statement information exchange. Both the information bearer and information receiver agree on the common shared world view, common shared inference logic, and common shared background knowledge in advance as part of the information exchange process. However, common information can be extended but the extension information is carefully associated with the existing common shared background knowledge.

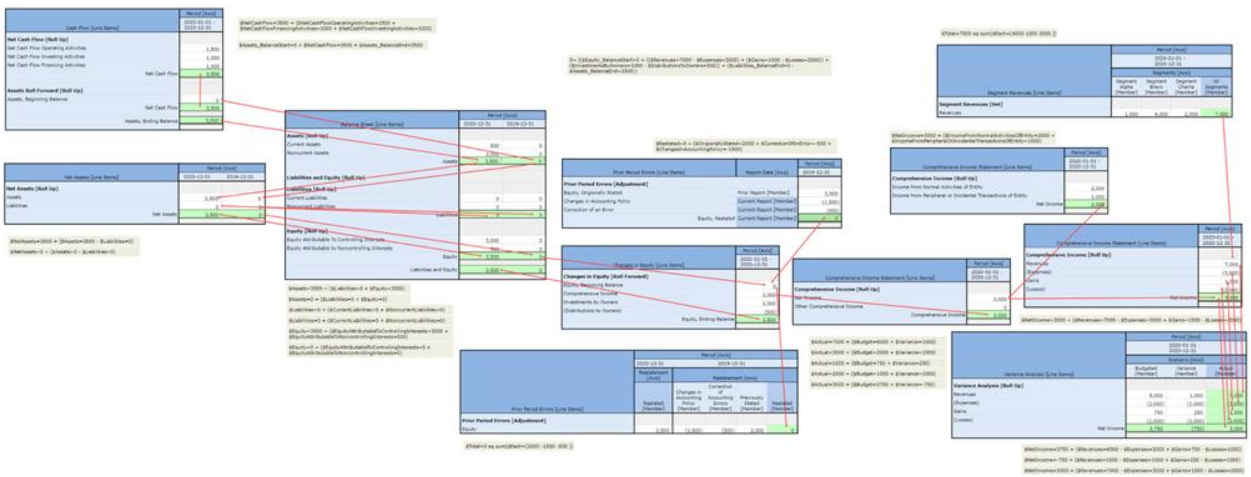
The "message" of this overall system is the general purpose financial report which is likewise a man-made logical system that has the five special characteristics we described earlier. There is nothing natural about a general purpose financial report, the idea was created by humans to serve a purpose. That purpose is to effectively exchange information about the financial status and financial performance of an economic entity. Initially, that was done using clay tablets. Then using papyrus. Then using paper. Then using e-paper. Today, the global standard XBRL-based digital format is increasingly being used. That digital format, the logical system, is consciously and deliberately configured to make it machine-readable and understandable by software applications.

Graphically depicted, the "message", the general purpose financial report, is a provably properly functioning logical system that is explainable using a logical theory[27] which should be consistent, complete, and precise:



[27] Charles Hoffman, CPA, *Logical Theory Describing Financial Report (Terse)*, http://xbrlsite.com/seattlemethod/LogicalTheoryDescribingFinancialReport_Terse.pdf

To make this more tangible to a business professional, consider the notion of articulation and how the facts reported within a financial report are interrelated to other facts if you consider only the mathematical computations of a rather basic general purpose financial report such as the following[28]:



I have demonstrated this by representing the accounting equation[29], SFAC 6[30], and common elements of financial statements[31] in the XBRL technical syntax, walking through all the things that can impede the communication process, and mitigating each impediment. The PROOF representation[32] contains an inventory of the complexity of a financial report. Mastering XBRL-based financial reporting[33] examples and prototypes represent reports that increase in volume but the complexity of any report is the same as the proof representation for all practical purposes. A comprehensive analysis of the Microsoft 10-K financial report shows this to be the case[34].

Fundamentally, it is the conscious intension of this logical system to safely, reliably, and otherwise successfully communicate financial information. The stakeholders fundamentally agree to eliminate all possible features that introduce potential failure and to leverage all possible features that lead to provable success. The stakeholders commit to the logical theory

---

[28] PROOF example, Articulation, http://xbrlsite.azurewebsites.net/2021/reporting-scheme/proof/reference-implementation/PROOF_Articulation.jpg
[29] Charles Hoffman, CPA, *Accounting Equation*, *http://accounting.auditchain.finance/examples/ae-basic/index.html*
[30] Charles Hoffman, CPA, SFAC 6, http://accounting.auditchain.finance/examples/sfac6-basic/index.html
[31] Charles Hoffman, CPA, Common Elements of Financial Statement, http://accounting.auditchain.finance/examples/common/index.html
[32] Charles Hoffman, CPA, Proof, http://accounting.auditchain.finance/reporting-scheme/proof/documentation/Index.html
[33] Mastering XBRL-based Digital Financial Reporting, http://xbrlsite.azurewebsites.net/2020/master/
[34] Knowledge Graph of Microsoft 10-K Financial Report, http://xbrl.squarespace.com/journal/2021/7/12/knowledge-graph-of-microsoft-10-k-financial-report.html

that describes the logical patterns of a financial report. Then, all of this is used to construct software that understands these ideas that the stakeholders have agreed to.

Fundamentally, the goal is to succeed and effectively exchange financial information using automated machine-based processes. This is done by agreeing to agree.

Principles

Principles help you think about something thoroughly and consistently. Overcoming disagreements between stakeholders and even within groups of stakeholders is important and principles can help in that communications process. The following principles make clear important considerations when communicating financial information in machine-readable form:

- A general-purpose financial report is a high-fidelity, high-resolution, high-quality information exchange mechanism. Its intension is, as best as practical, to faithfully represent a set of claims made by an economic entity about the financial position (a.k.a. financial status) and financial performance (a.k.a. change in financial status) of an economic entity. (i.e. a financial report is not arbitrary, is not random, is not illogical)
- Prudence dictates that using information from a financial report should not be a guessing game.
- All physical formats conveying the same set of financial information should convey the exact same meaning regardless of the information physical format be that format paper, e-paper, or some machine-readable format.
- Explicitly stated information or reliably derived information from information bearers is preferable to requiring information receivers to make assumptions.
- The double-entry bookkeeping model (Venetian Method) enables automation of processes that allow for the detection of information errors and to distinguish errors (unintentional) from fraud (intentional).
- The accounting equation, “Assets = Liabilities + Equity” is the foundation of every financial reporting scheme. There are various other forms of this equation which are semantically equivalent including, “Net Assets = Assets - Liabilities”.
- Each standards setter builds upon the double entry accounting model and some version of the accounting equation when they define their financial report elements.
- Catastrophic logical failures are to be avoided at all cost as they cause systems to completely fail.
- Nothing about processing information within this financial report logical system can be a “black box”. The innerworkings must be explainable and justifiable, providable in a human-readable manner. Information provenance must be knowable and traceable.

- The double-entry bookkeeping model as practices using the good practices Venetian Method follows a clear and commonly understood mathematical model and is based on science and is, therefore, objective. The art of accounting is deciding which information to provide, given known permissible alternatives, and is an exercise of subjective professional judgement[35]. Humans abusing the system by intentionally deceiving system users with fraudulent information is not a problem of the system, but a problem with the character of the users of the system; the system and the users of the system are distinct.

It would be, in my personal view, highly unlikely that anyone that fundamentally desires to effectively communicate machine-readable information and understands financial accounting to disagree with any of the very basic core principles.

Simple Explanation of Logical Systems and Logical Theory

A system can be explained by a logical theory. A logical theory is an abstract conceptualization[36] of specific important details of some area of knowledge. The logical theory provides a way of thinking about an area of knowledge by means of deductive reasoning to derive logical consequences of the logical theory by explaining the logical patterns of that system.

A **logical theory** enables a community of stakeholders trying to achieve a specific goal or objective or a range of goals/objectives to agree on important logical statements used for capturing meaning or representing a shared understanding of and knowledge in some area of knowledge.

A logical theory forms a logical conceptualization and is made up of a set of logical *models*, *structures*, *terms*, *associations*, *rules*, and *facts*. In very simple terms,

- **Logical conceptualization**: A *logical conceptualization* is a set of models that are consistent with and permissible per that logical conceptualization.
- **Model**: A *model*[37] is a set of structures that are consistent with and permissible interpretations of that model.
- **Structure**: A *structure* is a set of logical statements which describe the structure.

[35] *Puzzle Pieces of Digital Financial Reporting*, https://digitalfinancialreporting.blogspot.com/2023/11/puzzle-pieces-of-digital-financial.html
[36] Wikipedia, *Conceptual Model*, https://en.wikipedia.org/wiki/Conceptual_model
[37] Wikipedia, *Model Theory*, https://en.wikipedia.org/wiki/Model_theory

- **Logical statement**: A *logical statement* is a proposition, claim, assertion, belief, idea, or fact about or related to the area of knowledge to which the logical conceptualization relates. There are five broad categories of logical statements:
  - **Terms**: *Terms* are logical statements that define ideas used by the logical conceptualization such as "assets", "liabilities", "equity", and "balance sheet".
  - **Associations**: *Associations* are logical statements that describe permissible interrelationships between the terms such as "assets is part-of the balance sheet" or "operating expenses is a type-of expense" or "assets = liabilities + equity" or "an asset is a 'debit' and is 'as of' a specific point in time and is always a monetary numeric value".
  - **Rules**: *Rules* are logical statements that describe and make assertions what tend to be convertible into IF…THEN…ELSE types of relationships such as "IF the economic entity is a not-for-profit THEN net assets = assets - liabilities; ELSE assets = liabilities + equity".
  - **Facts**: *Facts* are logical statements about the numbers and words that are provided by an economic entity within a financial report. For example, the financial report might state "assets for the consolidated legal entity Microsoft as of June 20, 2017 was $241,086,000,000 expressed in US dollars and rounded to the nearest millions of dollars.
  - **Properties** are logical statements about the important qualities and traits of a model, structure, term, association, rule, and fact.

Fundamentally, a logical conceptualization is a set of logical statements that describe logical patterns. Those logical statements can be represented in human-readable form or they could be expressed in machine-readable form. Once in machine-readable form, those logical statements can be interrogated using software applications. To the extent that this can be performed effectively; software tools can assist professional accountants, financial analysts, and others working with those logical statements.

Properly Functioning Logical System

In her book *An Introduction to Ontology Engineering*[38], C. Maria Keet, PhD, provides discussion about what constitutes a good and perhaps a not-so-good ontology. There are three categories of errors she discusses:

[38] C. Maria Keet, PhD*, An Introduction to Ontology Engineering*, PDF page 23, https://people.cs.uct.ac.za/~mkeet/files/OEbook.pdf#page=23

- **Syntax errors**: She discusses the notion that a syntax error in an ontology is similar to computer code not being able to compile. For example, when an XBRL processor tells you that your XBRL taxonomy is not valid per the XBRL technical specification.
- **Logic errors**: She discusses the notion of logical errors within an information which cause the information to not work as expected. For example, if you represented something in your XBRL taxonomy as a credit when it should have been a debit.
- **Precision and coverage errors**: Finally, Keet discusses the notions of precision and coverage when it comes to judging whether information is good or bad.

These ideas related to ontologies are also appropriate for knowledge graphs and more specifically financial report knowledge graphs.

Keet uses those terms, others use different terms including sound, complete, and effective to describe a well-functioning logical system. There are other descriptions as well. The following is the description that I will use.

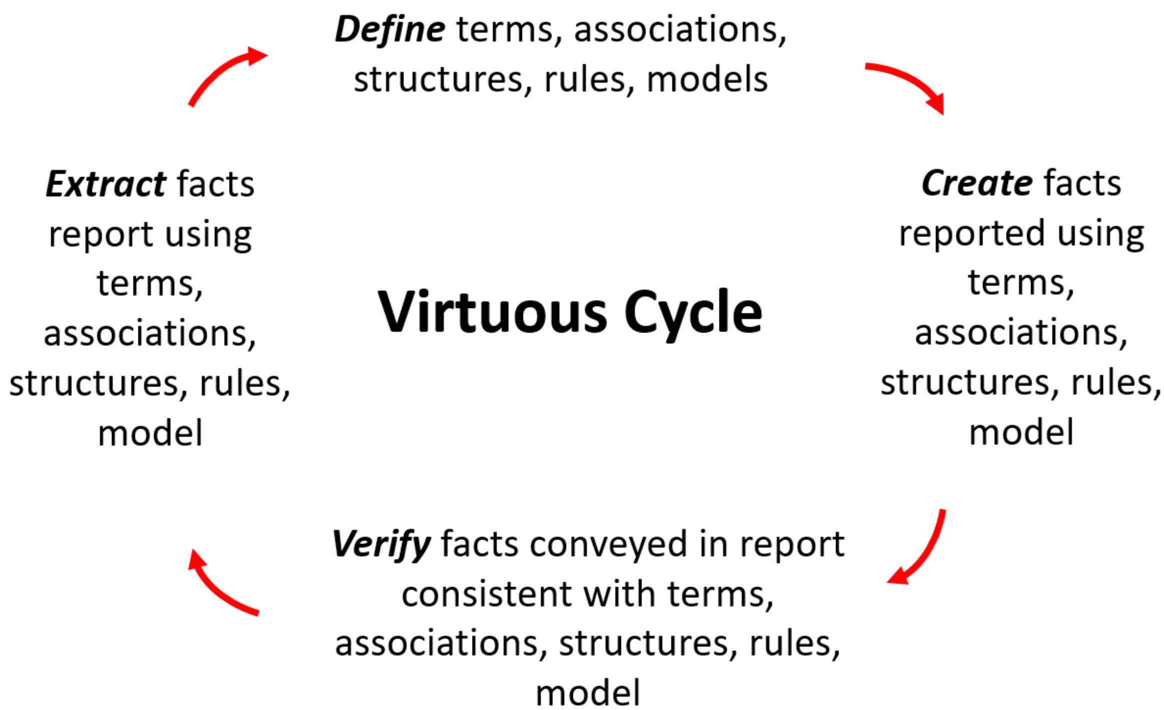
A logical theory can be used to describe a logical system. A logical system is said to be **consistent** with a logical theory if there are no contradictions with respect to the logical statements made by the logical theory that describes the logical system.

A logical theory can have high to low **precision** and high to low **coverage** with respect to describing a logical system.

Precision is a measure of how precisely the information within a logical theory has been represented as contrast to reality of the logical system for the area of knowledge. *Coverage* is a measure of how completely information in a logical theory has been represented relative to the reality of the logical system for the area of knowledge.

When a logical system is consistent and it has high precision and high coverage the logical system can be considered a **properly functioning logical system**. When a logical system is properly functioning, it creates a virtuous cycle[39].

[39] Charles Hoffman, CPA, *Virtuous Cycle*, http://xbrl.squarespace.com/journal/2020/4/29/virtuous-cycle.html



A logical theory conveys knowledge and that knowledge can be represented within a knowledge graph. For more detailed information related to logical theories and logical systems, please see *Logical Systems*[40].

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[41].

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 general purpose financial report. Below we will break down a financial report into helpful levels[42] that will enable a precise and

[40] Charles Hoffman, CPA, *Logical Systems*, http://www.xbrlsite.com/mastering/Part02_Chapter05.A_LogicalSystems.pdf
[41] Truecar, The 5 Levels of Autonomous Vehicles, https://www.truecar.com/blog/5-levels-autonomous-vehicles/
[42] Financial Report Levels, http://xbrl.squarespace.com/journal/2021/4/5/financial-report-levels.html

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 within a financial report.

- **Level 0 (Provide information physically)**: Not machine readable. *An example of Level 0 is a clay tablet, papyrus, or paper as the report medium*.
- **Level 1 (Provide information digitally)**: Machine readable, nonstandard, structured for presentation. *PDF, HTML, XHTML, and other forms of e-paper are examples of Level 1*.
- **Level 2 (Provide information digitally, structured for meaning)**: Machine readable, nonstandard, structured for meaning, no taxonomy (a.k.a. dictionary or associations), 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 (Standard syntax for structure)**: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary or associations), 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 (Common dictionary of terms)**: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary and associations), 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 (Complete set of logical statements)**: Machine readable, global standard syntax, structured for meaning, with taxonomy (a.k.a. dictionary and associations), 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 and logical 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 (Trust report logic not manipulated)**: 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 (Trust transaction provenance)**: All of Level 6 PLUS blockchain-anchored accounting 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*.

The *Seattle Method* defines the core logical kernel of a report and report model logic and the foundational rules of validity for report information. Report models and reported information must be consistent with the meta model defined by the *Seattle Method*, must be completely defined. The *Seattle Method* is about reporting information, it is not about computing information that will then be reported. This logical layer is distinct from the presentational layer of such information. The *Seattle Method* is about logical representation for machine understanding.

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. It is possible to create a Level 4 XBRL-based report that is properly functioning. Level 5 provides a guarantee that the Level 4 financial report is properly functioning within a provided specification articulated with a complete set of rules. Level 5 measures quality whereas Level 4 quality is essentially based on what amounts to luck or hope which are not effective engineering techniques.

Distilling Problem Down to Logic and Math

Rather than look at all the different moving pieces of this puzzle as being from different silos; I choose to leverage the good practices, best practices, safest practices, and create a solid, powerful, practical, and reliable system that business professionals can effectively understand and leverage by using other proven systems. Business professionals need not understand each individual theory, only that the theory has been proven.

Equilibrium is achieved by weaving the appropriately selected other systems based on the goals and objectives agreed to by the stakeholders of the information exchange mechanism. Testing and a conformance suite[43] which is agreed to by system stakeholders explains how the system works to business professionals. Business professionals decide if the system is working as expected.

[43] Conformance suite, http://xbrlsite.azurewebsites.net/2019/Prototype/conformance-suite/Production/index.xml

A logical system[44] is a type of formal system[45]. To be crystal clear what I am trying to create is a **finite model-based deductive first-order logic system**[46]. "Finite" as opposed to "infinite" because finite systems can be explained by math and logic, infinite systems cannot. "Model-based" is the means to address the necessary variability and therefore flexibility inherent in the required system. "Deductive", or rule-based and provides certainty, as contrast to inductive which is probability based which is not appropriate for this task because it is uncertain. "First-order logic" because first-order logic can be safely implemented within software applications and higher order logics are unsafe. "System" because this is a system. "Proof theory" because all of this can be proven mathematically which helps tune the system.

The point is to create a logical system that has high expressive capabilities but is also a provably safe and reliable system that is free from catastrophic failures and logical paradoxes which cause the system to completely fail to function. To avoid failure, computer science and knowledge engineering best practices seems to have concluded that the following alternatives are preferable:

- **Systems theory**: A system[47] is a cohesive conglomeration of interrelated and interdependent parts that is either natural or man-made. Systems theory explains logical systems. Systems have patterns.
- **Logical theory**: There are many approaches to representing logical systems in machine-readable form, a logical theory being the most powerful (ontology + rules). Theories describe logical patterns. (see the ontology spectrum[48])
- **Proof theory**: The ideas of proof theory[49] can be used to verify the correctness of logical systems and computer programs working with those machine-readable logical systems using mathematics[50]. Proofs verify theories.
- **Model theory**: Model theory is a way to think about flexibility. Safer finite model theory[51] is preferable to general model theory. Models provide flexibility.
- **Set theory**: Set theory is foundational to logic and mathematics. Axiomatic (Zermelo–Fraenkel) set theory[52] is preferred to naïve set theory.

---

[44] Wikipedia, *Logical Systems*, https://en.wikipedia.org/wiki/Logic#Logical_systems
[45] Wikipedia, *Formal System*, https://en.wikipedia.org/wiki/Formal_system
[46] Wikipedia, *First-order Logic, Deductive System*, https://en.wikipedia.org/wiki/First-order_logic#Deductive_systems
[47] Wikipedia, *Systems Theory*, https://en.wikipedia.org/wiki/Systems_theory
[48] *Difference between Taxonomy, Conceptual Model, Logical Theory*, http://xbrl.squarespace.com/journal/2018/12/11/difference-between-taxonomy-conceptual-model-logical-theory.html
[49] Stanford University, *The Development of Proof Theory*, *The Aims of Proof Theory*, https://plato.stanford.edu/entries/proof-theory-development/#AimProThe
[50] Samuel R. Buss, *An Introduction to Proof Theory*, https://math.ucsd.edu/~sbuss/ResearchWeb/handbookI/ChapterI.pdf
[51] Wikipedia, *Finite Model Theory*, https://en.wikipedia.org/wiki/Finite_model_theory
[52] Wikipedia, *Set Theory, Axiomatic Set Theory*, https://en.wikipedia.org/wiki/Set_theory#Axiomatic_set_theory

- **Graph theory**: Directed acyclic labeled typed property graphs[53] are preferred to less powerful "trees" and graphs which contain cycles that can lead to catastrophic problems caused by those cycles.
- **Logic**: Logic is a formal communications tool. Horn logic[54] is a subset of first-order logic and is the basis for Prolog[55]. Datalog[56] is a subset of Horn logic (function free PROLOG) which is immune from logical paradoxes should be used as contrast to more powerful but also more potentially problematic first order logic features. Note that deductive reasoning is leveraged for the process of creating a financial report and not inductive reasoning (i.e. machine learning).
- **World view**: The following are common issues which appear when implementing logical systems which exchange information in machine-readable form, the safest and most reliable alternatives are:
  - **closed world assumption**[57] (used by relational databases) is preferred to the open world assumption which can have decidability issues;
  - **negation as failure**[58] (used by relational databases) should be explicitly stated;
  - **unique name assumption**[59] (used by relational databases) should be explicitly stated;
- **Dimensional fact model**: The dimensional fact model[60] provides a clear and exhaustive representation of multidimensional concepts. XBRL Dimensions specifies a dimensional model.
- **Logical Theory Describing Financial Report**: The *Logical Theory Describing Financial Report*[61] is a logical conceptualization of the mechanical, mathematical, structural, and logical aspects of general purpose and special purpose financial reports for the purpose of representing such reports digitally using XBRL and other technical syntaxes.
- **Standard Business Report Model (SBRM)**: The Standard Business Report Model (SBRM)[62] formally documents a logical conceptualization of a business report in both human-readable and machine-readable models.
- **XBRL technical syntax physical format**: The Extensible Business Reporting Language (XBRL)[63] is the international standard for the electronic representation of business reports.

That is a lot to consider. How should a business professional think about all this "stuff"?

---

[53] Wikipedia, *Directed Acyclic Graph*, https://en.wikipedia.org/wiki/Directed_acyclic_graph
[54] Wikipedia, *Horn Logic*, https://en.wikipedia.org/wiki/Horn_clause
[55] Wikipedia, *Prolog*, https://en.wikipedia.org/wiki/Prolog
[56] Wikipedia, *Datalog*, https://en.wikipedia.org/wiki/Datalog
[57] Wikipedia, *Closed World Assumption*, https://en.wikipedia.org/wiki/Closed-world_assumption
[58] Wikipedia, *Negation as Failure*, https://en.wikipedia.org/wiki/Negation_as_failure
[59] Wikipedia, *Unique Name Assumption*, https://en.wikipedia.org/wiki/Unique_name_assumption
[60] Wikipedia, Dimensional Fact Model, https://en.wikipedia.org/wiki/Dimensional_fact_model
[61] Charles Hoffman, CPA, *Logical Theory Describing Financial Report (Terse)*, http://xbrlsite.com/seattlemethod/LogicalTheoryDescribingFinancialReport_Terse.pdf
[62] OMG, *Standard Business Report Model (SBRM)*, https://www.omg.org/intro/SBRM.pdf
[63] XBRL International, XBRL Essentials, https://specifications.xbrl.org/xbrl-essentials.html

Business professionals are (a) not trained for having precise discussions of these sorts of issues with software engineers, (b) don't care to have such technical discussions about these sorts of issues with software engineers, (c) are not interested in the theoretical or philosophical or religious debates that commonly exist related to these alternatives, (d) if the alternatives were appropriately articulated to a business professional, who tend to be very practical, they would most often error on the side of safety and reliability.

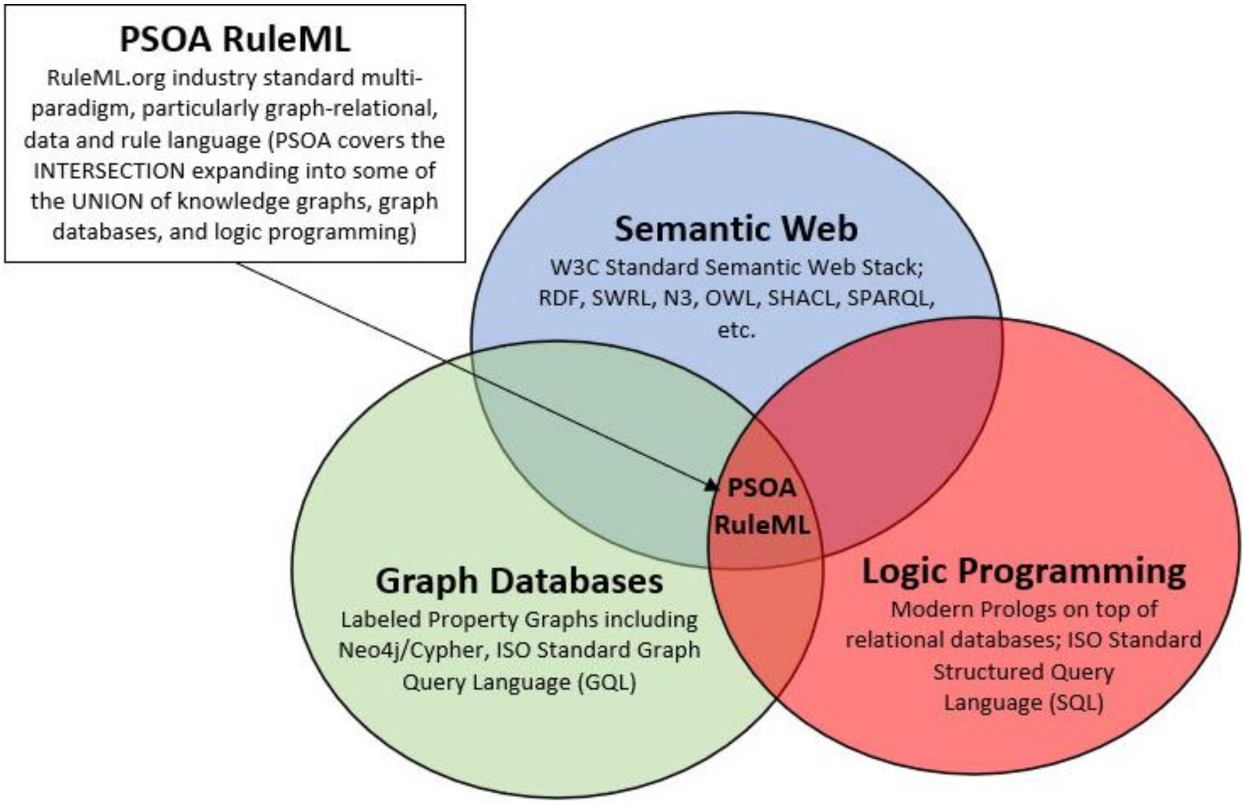
As such, we have made all of the above decisions which are consistent with modern logic programming paradigms such as Structured Query Language (SQL)[64], Graph Query Language (GQL)[65], the Semantic Web Stack[66], Prolog[67], LPS[68], DataLog[69], Efficiently Computable Datalog[70], Why3[71], Alt-Ergo[72], HETS[73], and Answer Set Programming[74].

All of these approaches can be distilled into three primary problem solving logic paradigms. Given that it is doubtful that you will convince every enterprise to have the same IT architecture; living in a world with multiple IT architectures is pretty much a given.

Primary Problem Solving Logic Paradigms

Per Harod Boley of RuleML[75], all these information processing approaches above can be distilled into one of the three fundamental problem solving logic paradigms[76]. With Mr. Boley's help I have made some modifications to his original graphic and explanations of those three problem solving logic paradigms. I have summarized this in this graphic[77]:

[64] Wikipedia, *SQL*, https://en.wikipedia.org/wiki/SQL
[65] Wikipedia, *Graph Query Language*, https://en.wikipedia.org/wiki/Graph_Query_Language
[66] Wikipedia, *Semantic Web Stack*, https://en.wikipedia.org/wiki/Semantic_Web_Stack
[67] Wikipedia, *Prolog*, https://en.wikipedia.org/wiki/Prolog
[68] Imperial College, Department of Computing, LPS, http://lps.doc.ic.ac.uk/
[69] Wikipedia, Datalog, https://en.wikipedia.org/wiki/Datalog
[70] Nichola Leona et.al., *Efficiently Computable Datalog Programs*, https://www.mat.unical.it/kr2012/shy.pdf
[71] Charles Hoffman, CPA, *Why3*, http://xbrl.squarespace.com/journal/2020/4/13/why3.html
[72] OCamlPro, *Alt-Ergo*, https://alt-ergo.ocamlpro.com/
[73] Charles Hoffman, CPA, *HETS*, http://xbrl.squarespace.com/journal/2020/4/10/hets.html
[74] Charles Hoffman, CPA, *Understanding Answer Set Programming*, http://xbrl.squarespace.com/journal/2019/5/10/understanding-answer-set-programming.html
[75] RuleML, Harold Boley, *Graph-Relational Data, Ontologies, and Rules*, http://wiki.ruleml.org/index.php/Graph-Relational_Data,_Ontologies,_and_Rules
[76] Problem Solving Logic Paradigms, http://xbrl.squarespace.com/journal/2020/9/15/primary-problem-solving-logic-paradigms.html
[77] Implementing Knowledge Graphs, http://xbrl.squarespace.com/journal/2021/9/20/implementing-knowledge-graphs.html



Further, all of the logic represented by one of these problem solving logic paradigms should be reconcilable to the logic expressed by each of the other two problem solving logic paradigms. Said another way, information should be 100% bidirectionally transferable between each of these three primary problem solving logic paradigms. PSOA RuleML[78] is that "sweet spot" in terms of functionality is where this is possible.

The bottom line here is that an integrated "graph-relational" problem solving logic paradigm will be in the future of most organizations. In particular the labeled directed property graphs will be of special interest to those implementing problem solving logic.

Saying this in yet another way; the focus of information exchange should be the logic of the information that is being exchanged and as long as the technical format supports that logic exchanging between paradigms will not be an issue.

Business professionals can simply use this system if they desire to do so, they don't need to reinvent the wheel. It does not matter which technical implementation is used, what matters is the logic.

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

A logical system or logical theory can be made flexible precisely where they need to be flexible using model theory[79].

Model theory essentially allows for any number of permissible interpretations of the logical theory, referred to as models. There are various forms of model theory including first order model theory[80], finite model theory[81], and the consciously and intentionally very safe finite first order model theory.

It is not important to understand the specific details of model theory, although it is very helpful to have a basic understanding[82]. I am not trying to prove the mathematics or logic of model theory; as I understand it that has already been proven.

What I am trying to do is apply the most powerful but also the safest, most reliable version of system theory, graph theory, model theory, set theory, logic, etc. in order to have the most expressive system possible that is also very safe and well behaved.

I can provide empirical evidence in the form of working representations of what I would call a finite model-based deductive first-order logic system using the global standard XBRL technical syntax[83]. Several of these examples have also been represented using Prolog; the XBRL and Prolog language representations yielding the same result. A smaller subset has also been converted to Cypher which is the graph query language of Neo4j. All of this was distilled into a method that provably yields high-quality information exchange where report model creators can make adjustments to that report model[84].

All the characteristics of the logical system that I point out are "necessary" meaning that they *must exist* within the logical system. What I cannot prove is that the characteristics are "sufficient" to prove that the logical system is provably consistent, precise, and complete. Perhaps a mathematician can provide this proof. Intuitively, the empirical evidence goes a long way towards proving this logical theory. Whether it goes far enough is up to others to determine.

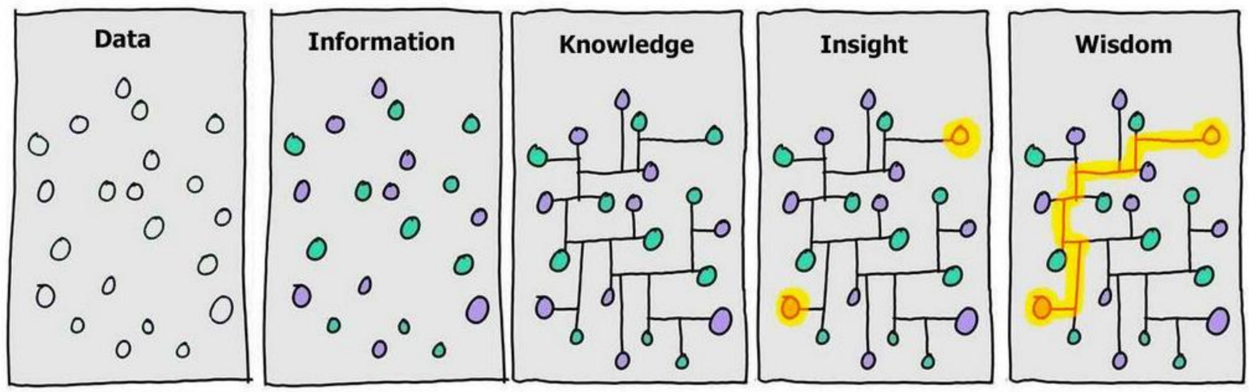
Key to understanding why we will be using relational databases and graph databases; it is important to understand the difference between data, information, and knowledge.

---

[79] Wikipedia, *Model Theory*, https://en.wikipedia.org/wiki/Model_theory
[80] Stanford University, *First Order Model Theory*, https://plato.stanford.edu/entries/modeltheory-fo/
[81] Wikipedia, Finite Model Theory, https://en.wikipedia.org/wiki/Finite_model_theory
[82] LessWrong, *Very Basic Model Theory*, https://www.lesswrong.com/posts/F6BrJFkqEhh22rFsZ/very-basic-model-theory
[83] Mastering XBRL-based Digital Financial Reporting, http://xbrlsite.azurewebsites.net/2020/master/
[84] Charles Hoffman, CPA, *Method – Terse Explanation*, http://xbrlsite.azurewebsites.net/2020/library/MethodTerse.pdf

Difference between Data, Information, Knowledge

The graphic below helps one understand the difference between data, information, knowledge, insight, and wisdom[85]:



The objective is to create a mechanism that will augment a human's capability to perform work by enabling software to take over some of the repetitive, mundane, mindless tasks that must be performed. Software applications can absorb some portion of this work if the software application can help its user understand the information the software user is working with.

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).

You can think about an area of knowledge as being characterized in a spectrum with two extremes:

- **Kind area of knowledge**: clear rules, lots of patterns, lots of rules, repetitive patterns, and unchanging tasks.
- **Wicked area of knowledge**: obscure data, few or no rules, constant change, and abstract ideas.

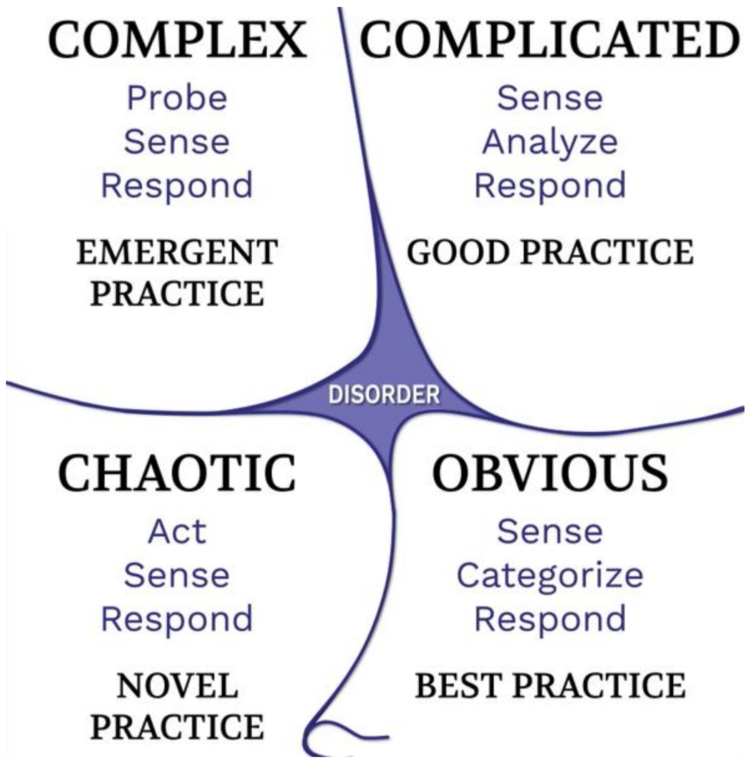
Accounting is an area of knowledge. You can explain aspects of the accounting area of knowledge, such as the nature of a financial report, using a logical theory which explains a

[85] Tumblr, *Information vs Knowledge*, https://informationversusknowledge-blog.tumblr.com/

logical model. A logical theory can be tested and proven by providing a proof. The accounting area of knowledge tends to be kind.

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.

The knowledge within an area of knowledge can be explained using the Cynefin Framework[86] which is a sensemaking process[87]. The video *Complexity, Cynefin, and Agile*[88] shows the categories of knowledge within an area of knowledge explained in terms of the Cynefin Framework:



Most accounting knowledge related to the repetitive, mechanical, mathematical, and logical aspects of accounting, reporting, auditing, and analysis are obvious and can be explained in terms of "best practices" or are complicated and can be analyzed by those with accounting expertise and explained as a set of "good practices". There are other frameworks similar to

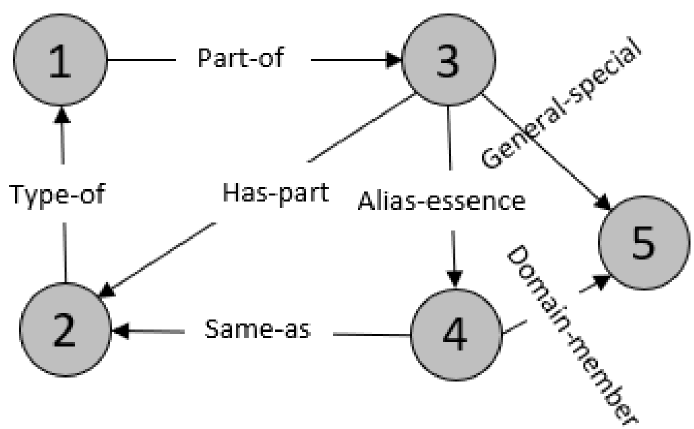
[86] YouTube.com, CognitiveEdge, *Cynefin Framework*, https://youtu.be/N7oz366X0-8
[87] Wikipedia, *Sensemaking*, https://en.wikipedia.org/wiki/Sensemaking
[88] YouTube.com, Complexity, Cynefin, and Agile, https://youtu.be/-F4enP8oBFM

Cynefin that help one make sense of things[89] such as ISO-9000 quality frameworks. Rules, algorithms, guidelines, and principles can be implemented and controlled using approaches such as Six Sigma[90].

These best practices and good practices knowledge can be represented in a machine-readable knowledge graph.

Knowledge Graph

Knowledge graphs are communications tools that are rich in terms of expressiveness but still innately understandable by humans. Further, knowledge graphs can also be read and understood by machines such as computers. Specifically, I am talking about labeled directed property graphs. Here is an example of such a knowledge graph:



A knowledge graph is one approach to storing information about some area of knowledge within a knowledge base. The specific term "knowledge graph" is more of an analogy or buzz word dreamed up in 2012 to describe the functionality you get when you use a set of web standards.   A knowledge graph is an approach to representing and storing information about entities, associations between those entities, rules related to entities and associations, and facts. Specifically, when I say knowledge graph, I mean labeled directed acyclic property graph.

Knowledge graphs is one of many different possible approaches to thinking about information. For more general information about knowledge graphs, I would recommend *The Knowledge Graph Cookbook: Recipes that Work*[91]. Different authors have different biases based on their

---

[89] Tom Graves / Tetradian, *And more 'Cynefin-like' cross-maps ('Beyond-Cynefin' series)*, http://weblog.tetradian.com/2010/02/28/and-more-crossmaps/
[90] YouTube.com, *Six Sigma In 9 Minutes*, https://youtu.be/4EDYfSl-fmc
[91] *The Knowledge Graph Cookbook: Recipes that Work*, http://xbrl.squarespace.com/journal/2021/6/27/the-knowledge-graph-cookbook-recipes-that-work.html

preferences. If one can see through these biases and look at this information in general terms and not per any specific technical implementation, one can get a very good understanding of how these systems work.

Financial Report Knowledge Graph

Knowledge graphs are general-purpose tools that can be modified and turned into special-purpose tools by adding a specific logical model that both constraints and controls the functionality of the general-purpose model.

Converting from a general-purpose tool to a special-purpose tool has two consequences. First, special-purpose tools are less functional and less flexible than general-purpose tools. Secondly, special-purpose tools are an order of magnitude easier to use that a general-purpose tool.

If you give up flexibility that you don't need then you lose nothing but you gain ease of use. That is the benefit of creating special-purpose tools for working with knowledge graphs.

A financial report is a special purpose knowledge graph[92]. In the past these financial report knowledge graphs were readable only by humans. Today with XBRL-based financial reports these financial report knowledge graphs are also readable by machines.

If you think about it, you will quickly recognize that the knowledge in those financial report knowledge graphs is the same knowledge for different parties that use that knowledge but the knowledge is used in different ways and for different things:

- The set of logical statements that is used to specify/describe how a report should be created (say by a regulator or standards setter),
- The set of logical statements used to actually create a report (say by an accountant),
- The set of logical statements used to verify that the report was created consistently to the specification/description (say an accountant or software application used by an accountant),
- The set of logical statements used to independently confirm that the report was created consistently with the specification/description (say by an independent auditor),
- The set of logical statements used to extract information from the created report (say by a financial analyst or regulator).

To understand better that financial reports are knowledge graphs, in the next section we will look at five very simple examples of these financial report knowledge graphs to help you get your head around this idea.

[92] Charles Hoffman, CPA, *Financial Report Knowledge Graphs*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf

Example Financial Report Knowledge Graphs in XBRL

Below I provide a handful of implementations that will be used to make some specific points about what is necessary to make the exchange of complex information work effectively when the creator of a report model can adjust that model within permitted boundaries.

Each example progressively increases in complexity. Simple examples help the reader get their heads around the fundamentals. To have a comprehensive example, all the different information patterns must be fundamentally provided for.

For the best understanding of these examples, I would encourage you to work through the examples of creating each of these using the cloud-based version of Luca[93]. Documentation is provided that will help you understand every detail of the example.

Accounting Equation (very basic report and report model)

The following is a very basic model of the accounting equation that I represented using XBRL and Prolog[94]:

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

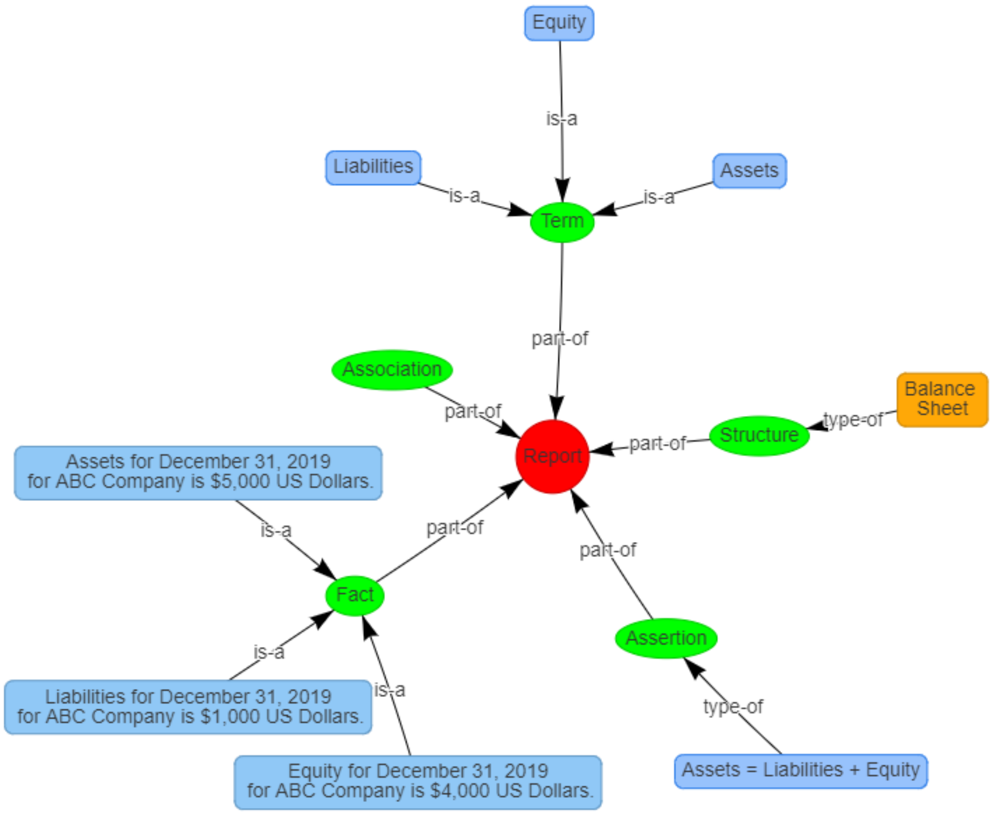
Result	Rule
Pass	$Assets = $Liabilities + $Equity

To understand this very basic model in detail, please read the documentation[95]. The essence of what you see is one structure defined using the functional term "Balance Sheet [Abstract]" that has three simple terms "Assets", "Liabilities", and "Equity", and one assertion "Assets = Liabilities + Equity".

This very basic model example is not enough to create an actual financial statement but it does represent a demonstrably complete, precise, and consistent logical system. Here is an example of a knowledge graph for that logical system:

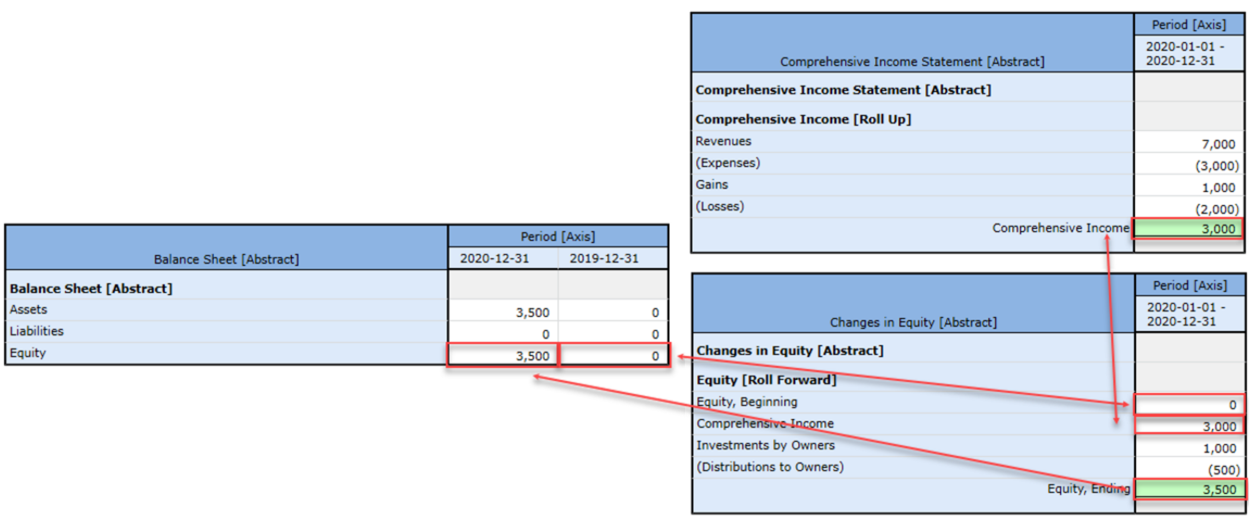
---

[93] Cloud-based Luca, http://xbrl.squarespace.com/journal/2021/8/31/cloud-based-luca.html
[94] Charles Hoffman, CPA, Accounting Equation, http://xbrlsite.azurewebsites.net/2020/Core/master-ae/
[95] Charles Hoffman, CPA, Accounting Equation Documentation, http://xbrlsite.azurewebsites.net/2020/Core/master-ae/Documentation.pdf



SFAC 6 (slightly more complex)

The following is a slightly more complex, but still pretty basic model that represents what is articulated by the FASB in SFAC 6 related to the elements of a financial statement[96]:



Balance Sheet [Abstract]	Period [Axis] 2020-12-31	2019-12-31
Balance Sheet [Abstract]		
Assets	3,500	0
Liabilities	0	0
Equity	3,500	0

Comprehensive Income Statement [Abstract]	Period [Axis] 2020-01-01 - 2020-12-31
Comprehensive Income Statement [Abstract]	
Comprehensive Income [Roll Up]	
Revenues	7,000
(Expenses)	(3,000)
Gains	1,000
(Losses)	(2,000)
Comprehensive Income	3,000

Changes in Equity [Abstract]	Period [Axis] 2020-01-01 - 2020-12-31
Changes in Equity [Abstract]	
Equity [Roll Forward]	
Equity, Beginning	0
Comprehensive Income	3,000
Investments by Owners	1,000
(Distributions to Owners)	(500)
Equity, Ending	3,500

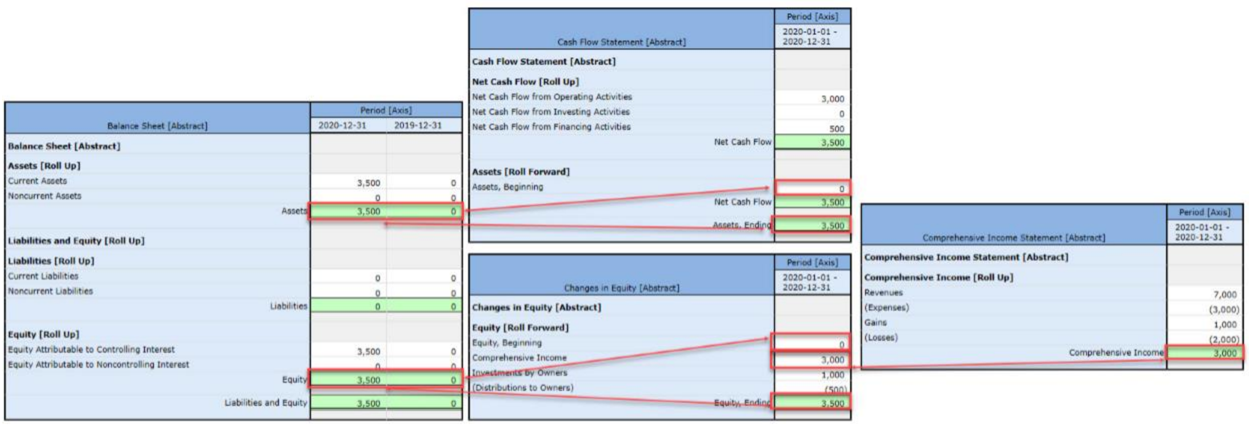
[96] Charles Hoffman, CPA, SFAC 6, http://xbrlsite.azurewebsites.net/2020/master/sfac6/

Again, the best way to understand all the details are to read the documentation[97]. The essence of the representation, again both in XBRL and Prolog and also using Cypher; are three interconnected structures, ten terms, and three rules defined by SFAC 6.

Again, this slightly more complex, but still pretty basic model is a demonstrably complete, precise, and consistent logical system.

Common Elements of Financial Statement (four statement model)

The following is again another slightly more complex model[98], still pretty basic model that expands on the FASB's SFAC 6 adding additional elements that no professional accountant could really dispute:



Again, the documentation provided helps one understand the representation in detail[99]. What you see are four interconnected structures, 20 terms, four assertions, 29 facts, and a plethora of associations.

MINI Financial Reporting Scheme

The accounting equation example, the SFAC 6 example, and the common elements of financial report example were created because they are grounded in well understood accounting ideas but were small enough to understand all the moving pieces of the puzzle without the need of automated processing to prove that everything works as would be expected. Humans can simply look and see that everything works as expected.

[97] Charles Hoffman, CPA, SFAC 6 Documentation, http://xbrlsite.azurewebsites.net/2020/Core/master-sfac6/Documentation.pdf
[98] Charles Hoffman, CPA, Common Elements of Financial Statement (Four Statement Model), http://xbrlsite.azurewebsites.net/2019/Core/master-elements/
[99] Charles Hoffman, CPA, Common Elements of Financial Statement , http://xbrlsite.azurewebsites.net/2019/Core/master-elements/CommonElementsOfFinancialStatement.pdf

The MINI Financial Reporting Scheme example[100] takes a significantly larger step toward what an actual financial report might look like. While the MINI Financial Reporting Scheme might look relatively small, don't be fooled by its simplicity. The MINI example contains 100% of the use cases that one will ever find in an XBRL-based digital financial report. The example was intentionally engineered to be a comprehensive test of XBRL-based financial reports. This example is explained in the document, *Proving Financial Reports are Properly Functioning Logical Systems*[101]. It is also compared and contrasted to the smaller examples and then to a complete 10-K financial report of Microsoft. I believe that this helps the reader bridge the gap between the smaller examples and larger, actual financial reports.

Looking at these examples, patterns emerge.

PROOF (complete model)

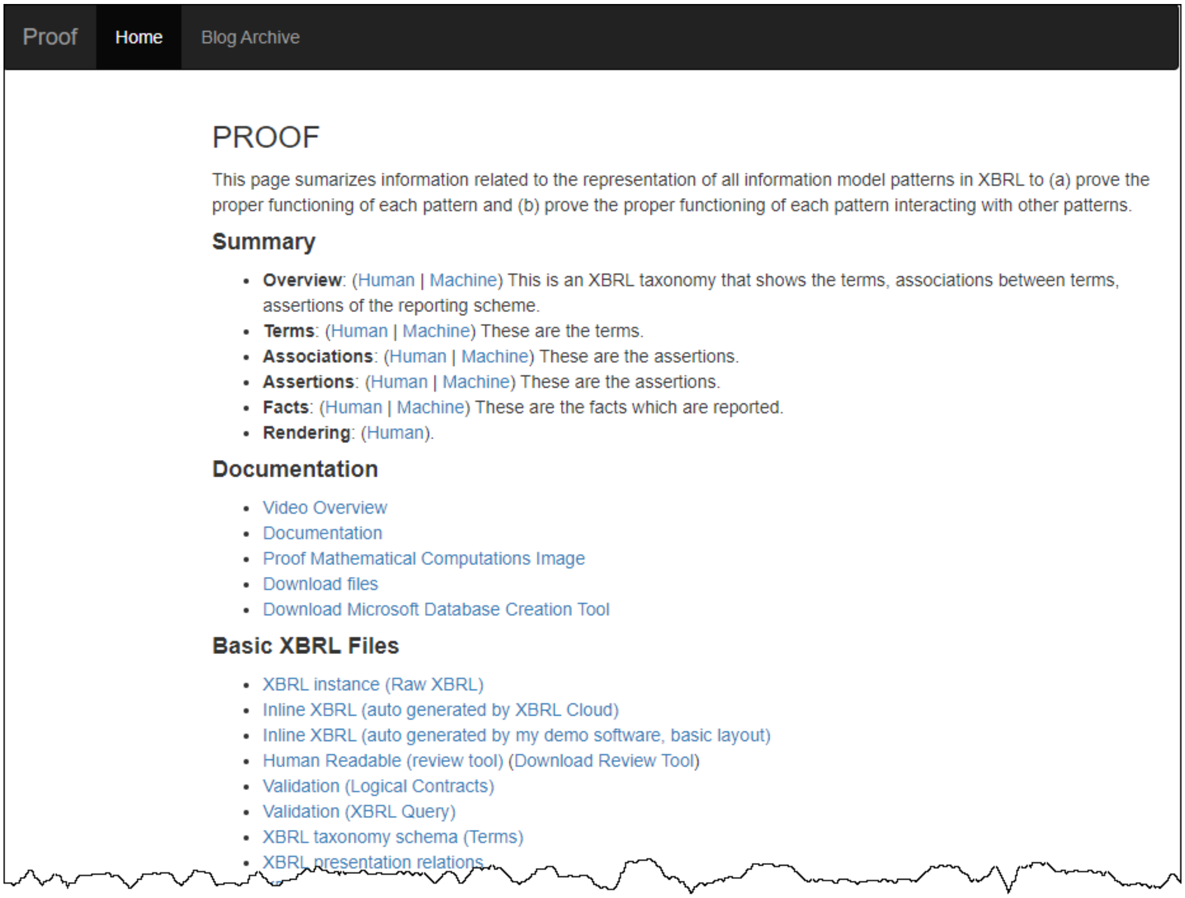
The Proof representation[102] contains all the technical and logical complexity (i.e. patterns) that have been discovered from analyzing about 6,000 US GAAP financial reports and 400 IFRS financial reports that have been submitted to the U.S. Securities and Exchange Commission. The Proof representation takes all of those patterns, represents them within an XBRL report model and report, tests that representation to make sure everything works logically as expected. Further, additional logic was added and represented for known logical patterns of financial reporting that were not found in XBRL-based financial reports submitted to the SEC.

ESMA reports XBRL-based financial reports are expected to be very consistent to SEC XBRL-based financial reports in terms of the logic of the report model itself. Yes, what goes into the report can be different because the SEC and ESMA have different specific requirements. But the report models themselves are logically consistent. Further, both SEC and ESMA XBRL-based financial reports are consistent with the XBRL International *Open Information Model (OIM)* 1.0[103].

While the XBRL International explanation of the logical model of a business report tends to be quite technical, the logical conceptualization of a business report is explained in terms more approachable and understandable to business professionals by the Standard Business Report

[100] Charles Hoffman, CPA, *MINI Financial Reporting Scheme*, http://xbrlsite.azurewebsites.net/2019/Prototype/mini/documentation/Index.html
[101] Charles Hoffman, CPA, *Proving Financial Reports are Properly Functioning Logical Systems*, http://xbrlsite.azurewebsites.net/2019/Library/ProvingFinancialReportAreProperlyFuncioning.pdf
[102] Proof representation, http://xbrlsite.azurewebsites.net/2020/master/proof/index.html
[103] XBRL International, Open Information Model (OIM) 1.0, https://specifications.xbrl.org/work-product-index-open-information-model-open-information-model.html

Model (SBRM)[104]. SBRM documents the logical patterns that exist in financial reports. The PROOF representation provides an inventory of the complete set of those logical patterns:


Proof | Home | Blog Archive

PROOF

This page sumarizes information related to the representation of all information model patterns in XBRL to (a) prove the proper functioning of each pattern and (b) prove the proper functioning of each pattern interacting with other patterns.

Summary

- **Overview**: (Human | Machine) This is an XBRL taxonomy that shows the terms, associations between terms, assertions of the reporting scheme.
- **Terms**: (Human | Machine) These are the terms.
- **Associations**: (Human | Machine) These are the assertions.
- **Assertions**: (Human | Machine) These are the assertions.
- **Facts**: (Human | Machine) These are the facts which are reported.
- **Rendering**: (Human).

Documentation

- Video Overview
- Documentation
- Proof Mathematical Computations Image
- Download files
- Download Microsoft Database Creation Tool

Basic XBRL Files

- XBRL instance (Raw XBRL)
- Inline XBRL (auto generated by XBRL Cloud)
- Inline XBRL (auto generated by my demo software, basic layout)
- Human Readable (review tool) (Download Review Tool)
- Validation (Logical Contracts)
- Validation (XBRL Query)
- XBRL taxonomy schema (Terms)
- XBRL presentation relations

AASB 1060 (20% of a real financial reporting scheme)

The AASB 1060 working prototype[105] is a representation of about 20% of a real financial reporting scheme that is similar to IFRS for SMEs.

Key Patterns Documented by Standard Business Report Model (SBRM)

Examining the patterns[106] of the first four examples, an additional small financial reporting scheme representation[107], and reconciling all examples to a full 10-K financial statement of a

[104] Auditchain, *Standard Business Report Model (SBRM)*, http://accounting.auditchain.finance/sbrm/index.html
[105] *Two AASB Reports to Fiddle With*, https://digitalfinancialreporting.blogspot.com/2024/01/two-aasb-1060-reports-to-fiddle-with.html
[106] YouTube, *The Science of Patterns*, https://www.youtube.com/watch?v=kh6KMW8J3RQ
[107] Charles Hoffman, CPA, MINI Financial Reporting Scheme, http://xbrlsite.azurewebsites.net/2019/Prototype/mini/documentation/Home.html

public company in the document *Proving Financial Reports are Properly Functioning Logical Systems*[108], shows that all of these financial report related representations (a) follow the documented logical system of a financial report and (b) point out an even more detailed model of a business report and financial report that is documented in the forthcoming OMG standard, *Standard Business Report Model (SBRM)*[109].

While the more detailed patterns are quite helpful at arriving at the fundamental description of a logical theory of a financial report; it is the *Logical Theory Describing Financial Report*[110] itself which explains the logic of a financial report. That high-level theory explains what logical statements must be communicated and that those logical statements must be consistent, complete, and precise.

Finally, the impediments to creating a properly functioning logical system document the properties that must exist within a logical system for that logical system to be considered properly functioning. Here is a quick summary the impediments or things that can go wrong[111]:

- Improper XBRL technical syntax used to represent logic
- Improper mathematical associations between reported facts
- Improper XBRL presentation relations associations (i.e. improper report model)
- Improper use of a type of line item as if were some different type of line item
- Inconsistent or contradictory high-level reported information
- Improper mechanical structure of disclosures
- Improper or incomplete set of disclosures provided within report

When all of these impediments are overcome, then logical information can be effectively communicated by the report. Note that (a) improper technical format syntax, in this case XBRL, is a given and (b) does not tend to be a problem because of the rigorous conformance suite provided by XBRL International[112] which is used which effectively guarantees interoperability because 100% of the conformance suite is automated.

And so, to effectively communicate logical information (a.k.a. semantics, meaning) the impediments described above simply need to be mitigated. Empirical evidence exists that

---

[108] Charles Hoffman, CPA, *Proving Financial Reports are Properly Functioning Logical Systems*, http://xbrlsite.azurewebsites.net/2019/Library/ProvingFinancialReportAreProperlyFuncioning.pdf
[109] *OMG Standard Business Report Model (SBRM) Initial Submission Information*, http://xbrl.squarespace.com/journal/2019/11/15/omg-standard-business-report-model-sbrm-initial-submission-i.html
[110] *Logical Theory Describing Financial Report*, http://xbrl.squarespace.com/logical-theory-financial-rep/
[111] Learning XBRL-based Reporting: PROOF Verification using Seattle Method, http://xbrlsite.azurewebsites.net/2022/Prototypes/proof/Dashboard.html
[112] XBRL International, *XBRL 2.1 Conformance Suite*, https://specifications.xbrl.org/work-product-index-group-base-spec-base-spec.html

shows the reliable detection of these impediments, the correction of the impediment, and the resulting properly functioning logical system, the XBRL-based digital financial report[113].

But none of this necessarily guarantees that *every model* that needs to be created by reporting entities can be effectively created and how to *control* what could be an arbitrarily large set of finite models.

Large Set of Specific Finite Models

No one would really dispute that it is possible to effectively exchange information from some sender to some receiver if the machine-readable message is an unchangeable form and both the sender and receiver of the information have exactly the same world view, use the same inference logic (basically no inference logic would really be necessary), and have the same common shared background knowledge.

For example, take this very simple form[114]:

Property, Plant and Equipment Subclassifications [Line Items]	Period [Axis]	
	2018-12-31	2017-12-31
Property, Plant and Equipment [Roll Up]		
Property, Plant and Equipment, Gross [Roll Up]		
Land	1,000	1,000
Buildings	1,000	0
Equipment	4,000	0
Property, Plant and Equipment, Gross	6,000	1,000
Accumulated Depreciation	0	0
Property, Plant and Equipment	6,000	1,000

If every economic entity were required to report the roll up of property, plant, and equipment subclassifications in exactly the same manner using exactly the same concepts and still used the same world view and inference assumptions I think it would be easy to understand that the communication of such information in machine-readable form would be rather trivial.

[113] YouTube.com, *Understanding the Financial Report Logical System*, https://www.youtube.com/playlist?list=PLqMZRUzQ64B7EWamzDP-WaYbS_W0RL9nt

[114] Company 1, http://xbrlsite.azurewebsites.net/DigitalFinancialReporting/mini/repository/company1/evidence-package/contents/index.html#Rendering-PropertyPlantAndEquipmentDetail-mini_PropertyPlantAndEquipmentSubclassificationsHypercube.html

However, that is not the way financial reporting schemes work. For example, the following is a possible allowed interpretation of what amounts to the breakdown of the subclassifications of property, plant and equipment:

	Period [Axis]	
Property, Plant and Equipment Subclassifications [Line Items]	2018-12-31	2017-12-31
Property, Plant and Equipment [Roll Up]		
Property, Plant and Equipment, Gross [Roll Up]		
Land and Buildings	2,000	1,000
Computer Equipment	2,000	0
Manufacturing Equipment	2,000	0
Property, Plant and Equipment, Gross	6,000	1,000
Accumulated Depreciation	0	0
Property, Plant and Equipment	6,000	1,000

What is different between the first example and the second example is the subclassifications of the line items that are actually disclosed. Note that in the above representation the subclassifications "Land" and "Buildings" have been combined and that "Equipment" has been disaggregated and "Computer Equipment" and "Manufacturing Equipment" have been reported.

This sort of variability is common in financial reports and can make it more challenging for those who desire to make use of the information reported to do so effectively. Even though one could effectively argue that the two examples of property, plant, and equipment disclosures would be quite easy to compare; it is easy to grasp that if, say, the subtotal and the grand total concepts were also changed that could make using the information more challenging.

So, the fact that for the past 10 years thousands of U.S. public companies have created literally tens of thousands of reports using XBRL and have submitted the reports to the U.S. Securities and Exchange Commission is evidence that it is possible to represent both models of the subclassifications of things such as property, plant, and equipment effectively.

However, can the information be used effectively by financial analysts?

Complains about information quality[115], the excessive use of extension concepts, and other such complaints that tend to be rather general in nature (as compared to very precise and specific complains). Also, the goal is not to complain; rather, the objective is to effectively communicate financial information between the sender/creator of the information and the receiver/analyst that would like to actually make use of the reported financial information.

The next section shows that it is possible to reliably extract information from a digital financial report if the appropriate machine-readable statements are provided within the financial report logical system.

Extending Models and Providing Important Properties

Essentially, the primary financial statements and the related policies and disclosures provided in the disclosure notes can be represented using any permitted alternative model. This does not mean that disclosures can be "random" or "illogical" or completely "arbitrary". Rational thinking does play a role here. What is permitted can be a bit subjective because the existing financial reporting standards can be ambiguous in some areas. But, given some interpretation of the financial reporting standards whether a disclosure is permitted or not permitted can be quantified into some finite set of possible disclosures. That finite set of possible disclosures can be represented using the XBRL technical syntax.

So intuitively, one could imagine that it is possible to represent the finite set of possible information representations into some number of what would amount to forms for each possible representation alternative permitted for each possible disclosure. Potentially a lot of work, but certainly possible.

But how do those that wish to use the information reported within a specific disclosure actually locate that specific permitted alternative disclosure within the set of all disclosures which make up a financial statement? It is possible to actually physically name each of those permitted possible disclosures[116].

And so how does XBRL-based financial reporting satisfy both the needs of economic entities reporting information and the needs of analysts to consume that information? The short answer is consciously, skillfully, and consistently.

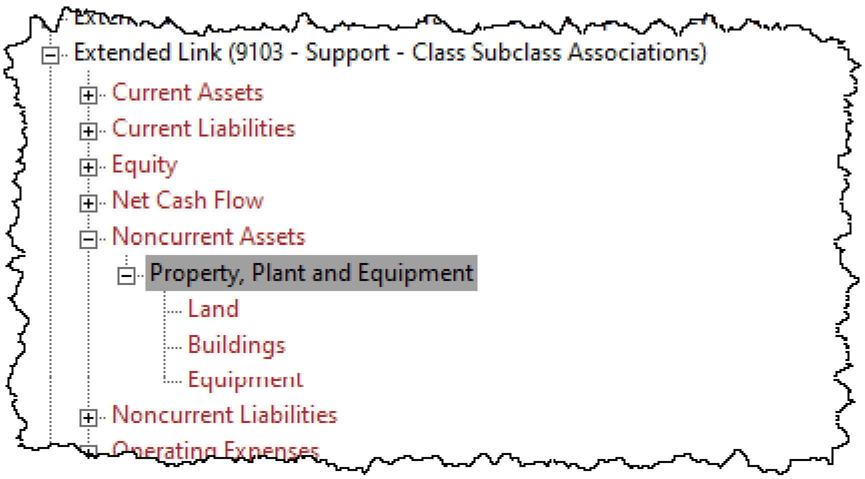
---

[115] 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

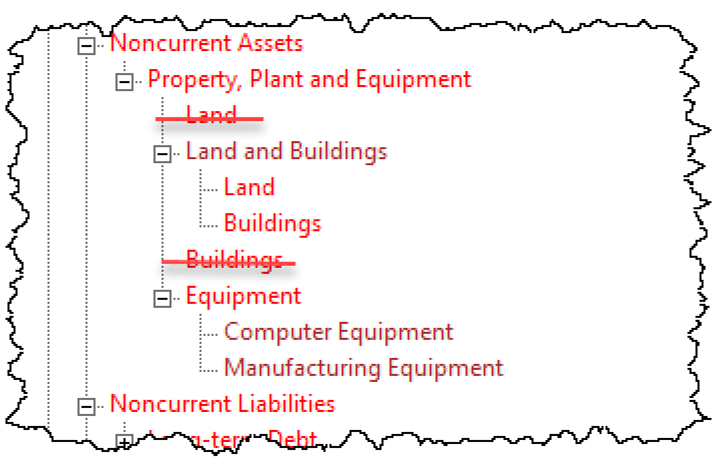
[116] *US GAAP Disclosures*, http://xbrlsite.azurewebsites.net/2020/reporting-scheme/us-gaap/documentation/Disclosures.html

The ESMA's use of "wider-narrower" association and "anchoring" is one possible approach[117]. Although, this approach has always existed in XBRL via the "general-special" association. So, for example, two things are necessary to satisfy the property, plant, and equipment example shown previously.

First, some explicit structure is necessary to anchor to. For example, here are a set of "general-special" relations represented in a prototype XBRL taxonomy:



Then second, once the context is clear (i.e. which structure you are working within), then new associations can be established per the model of the reporting economic entity relative to the base model of the financial reporting scheme:



[117] *ESMA Explains Anchoring and 2020 ESEF Implementation Requirement*, http://xbrl.squarespace.com/journal/2019/3/1/esma-explains-anchoring-and-2020-esef-implementation-require.html

In this manner, any extended concept that is defined relative to some existing base model concept can be understood correctly per the "wider-narrower" or "general-special" association and anchoring to that existing concept.

That works when there is some base taxonomy report element that can be anchored to. But what about a completely new structure?

This is a completely new structure which has an existing report element from the base taxonomy as part of that new structure:

Finished Goods Subclassifications [Line Items]	Period [Axis]	
	2018-12-31	2017-12-31
Finished Goods [Roll Up]		
Product Foxtrot	300	550
Product Golf	100	50
Product Hotel	100	50
Product India	100	50
Finished Goods	600	700

So, in the representation above, the total of the roll up, "Finished Goods" is from a base taxonomy but the items that make up that total are all extension concepts created by the reporting entity. While the structure is new, it is identifiable by the pattern of the information and the total.

Finally, below you see a completely new structure that is in no way associated with any existing report element that is defined within the base financial reporting scheme model:

Etiam [Roll Up]	Period [Axis]	
	2018-12-31	2017-12-31
Etiam [Roll Up]		
Eu eleifend augue	600	700
Est aliquet ante	200	100
Nunc mattis aliquam	200	200
Etiam	1,000	1,000

But just because some new completely new structure with completely new report elements does not mean that nothing is known about the new structure. The fact that the structure is a roll up, for example, is known.

When a new extension is created, there are exactly four logical possibilities of how that new idea can be associated to some potentially existing idea:

- More general idea
- More specific idea
- Similar idea
- Completely new Idea

Even if the idea is completely new, because of the fundamental primitive building blocks of XBRL-based reports, every completely new thing must be (per XBRL syntax rules as restricted by SEC EDGAR Filing manual rules) represented using one of the primitive building blocks provided by XBRL.

Below you see those primitive building blocks:

- **Term (primitive or atomic term)**
  - Dimension (a.k.a. Axis)
  - Member
  - Primary Items (a.k.a. Line Items)
  - Abstract
  - Concept
    - Level 1 Note Text Block
    - Level 2 Policy Text Block
    - Level 3 Disclosure Text Block
    - Level 4 Detail
- **Structure (functional term)**
  - Network
    - Document
    - Statement
    - Disclosure
    - Schedule
  - Hypercube (a.k.a. Table)
- **Associations**
  - Parent-child
  - Summation-item
  - Essence-alias
  - General-special
  - Other associations
  - Property associations
    - Concept-label
    - Label-role
    - Concept-reference
    - Reference-role
    - Reference-part
- **Assertion**
  - XBRL Formula or XBRL Calculation
- **Fact**

For brevity, some possibilities are not shown. But this makes the point that there is a finite set of primitive structures that can be used to create anything that is possible to add to a financial reporting scheme. No XBRL-based model can add any new ideas at the first two layers. It is only below those first to layers that creators of an extension can work with.

I have provided mappings of the XBRL-based report objects to the hierarchy above for both the accounting equation[118] and SFAC 6[119] examples. See the last page of the documentation.

Modifying Existing Associations

In addition to creating a new disclosure by extending the information of a base taxonomy with new information, it is possible to modify existing associations, correctly or incorrectly, and represent disclosures using alternative approaches.

For example, consider the following long-term debt maturities disclosure:

	Period [Axis]	
Long-term Debt Maturities [Line Items]	2018-12-31	2017-12-31
Long-term Debt Maturities [Roll Up]		
Matures in One Year	1,000	
Matures in Two Years	1,000	
Matures in Three Years	1,000	
Matures in Four Years	1,000	
Matures in Five Years	1,000	
Matures Thereafter	1,000	
Long-term Debt	6,000	1,000

Above the disclosure is represented as a roll up of a set of items to a total.

Below you see an alternative representation based on the fact that numerous public companies represent this same disclosure by modifying the set of associations, dropping the total, and simply providing information about the maturities without the total:

[118] Accounting Equation example, Documentation, page 13, http://xbrlsite.azurewebsites.net/2019/Core/master-ae/Documentation.pdf#page=13
[119] SFAC 6 example, Documentation, page 21, http://xbrlsite.azurewebsites.net/2019/Core/core-sfac6/Documentation.pdf#page=21

Long-term Debt Maturities [Line Items]	Period [Axis]
	2018-12-31
Long-term Debt Maturities	
Matures in One Year	1,000
Matures in Two Years	1,000
Matures in Three Years	1,000
Matures in Four Years	1,000
Matures in Five Years	1,000
Matures Thereafter	5,000

The point is not about whether either the version of the disclosure with the roll up total or the version without the total are both allowed or not. The point is that per model theory, it is possible to represent both representations or any other alternative that a public company creating this disclosure might come up with.

Representing the disclosure effectively and whether a represented disclosure is or is not permissible per financial reporting rules and practices are two different questions.

Proper Use of Subtypes

An XBRL taxonomy is not, or should not, be simply a list of terms. An XBRL taxonomy, at a very minimum, should provide a set of terms and a comprehensive set or sets of associations between terms that document the proper use of the term. Consider this example of a cash flow statement:

Cash Flow Statement [Line Items]	Period [Axis] 2018-01-01 - 2018-12-31
Cash Flow Statement [Roll Forward]	
Net Cash Flow [Roll Up]	
Net Cash Flow Operating Activities [Roll Up]	
Collection of Receivables	3,000
Payment of Accounts Payable	(2,000)
Net Cash Flow Operating Activities	1,000
Net Cash Flow Financing Activities [Roll Up]	
Additional Long-term Borrowings 2	6,000
Repayment of Long-term Borrowings 2	(1,000)
Net Cash Flow Financing Activities	5,000
Net Cash Flow Investing Activities [Roll Up]	
Capital Additions of Property, Plant and Equipment 2	(5,000)
Net Cash Flow Investing Activities	(5,000)
Net Cash Flow	1,000
Cash and Cash Equivalents, Beginning Balance	3,000
Cash and Cash Equivalents, Ending Balance	4,000

Note that in the example above, the line items “Additional Long-term Borrowings” and “Repayment of Long-term Borrowings” are part of “Net Cash Flow Financing Activities”. Contrast that to the example below which uses those two line items as part of “Net Cash Flow from Investing Activities”. (Essentially, the financing activities and investing activities line items have been switched.)

Cash Flow Statement [Line Items]	Period [Axis] 2018-01-01 - 2018-12-31
Cash Flow Statement [Roll Forward]	
Net Cash Flow [Roll Up]	
Net Cash Flow Operating Activities [Roll Up]	
Collection of Receivables	3,000
Payment of Accounts Payable	(2,000)
Net Cash Flow Operating Activities	1,000
Net Cash Flow Financing Activities [Roll Up]	
Capital Additions of Property, Plant and Equipment 2	(5,000)
Net Cash Flow Financing Activities	(5,000)
Net Cash Flow Investing Activities [Roll Up]	
Additional Long-term Borrowings 2	6,000
Repayment of Long-term Borrowings 2	(1,000)
Net Cash Flow Investing Activities	5,000
Net Cash Flow	1,000
Cash and Cash Equivalents, Beginning Balance	3,000
Cash and Cash Equivalents, Ending Balance	4,000

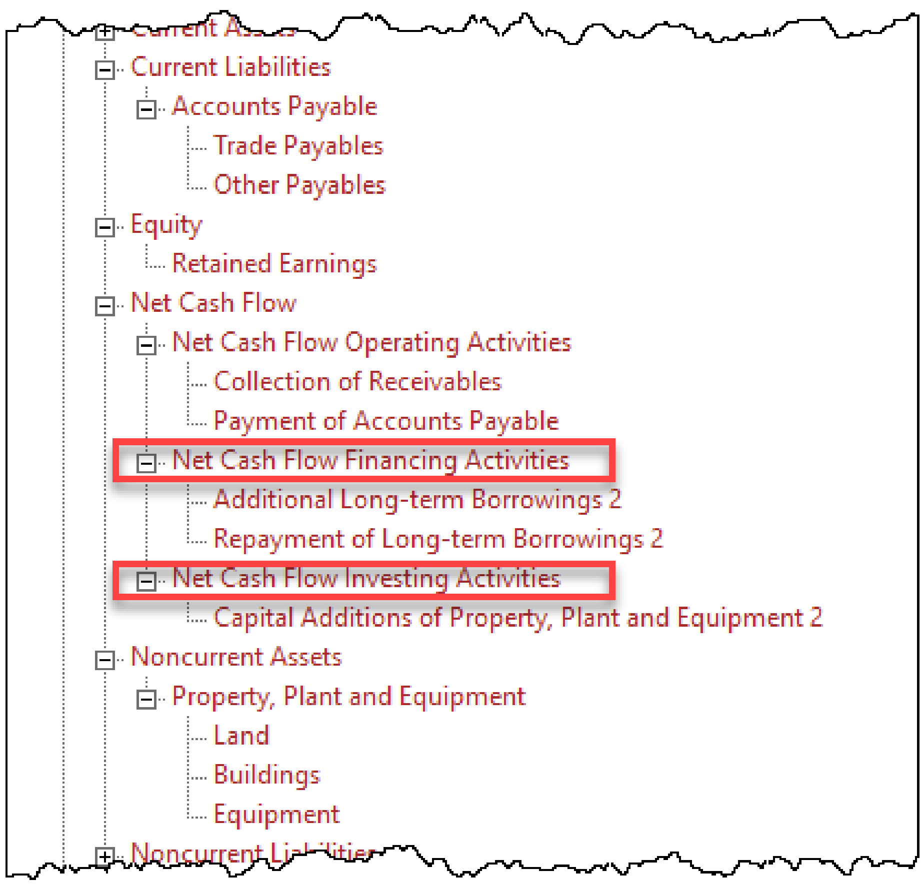
While for this specific example it is probably the case that every professional accountant would recognize that additional borrowings and repayments should be part of financing activities and not investing activities. But the obvious mistake was used to make a specific point.

How exactly do you communicate within an XBRL taxonomy where line items can, and cannot, be used? How do you know that something is a current asset and not a noncurrent asset?

Taxonomies have long been tools for representing this sort of information in the form of a hierarchy of "general" and "special" relations or perhaps "wider" or "narrower" concepts in the form of a thesaurus.

The same information can, should, and in fact must be articulated within an XBRL taxonomy or any other logical system that hopes to be effective and have the remotest chance of working effectively to communicate information represented in machine-readable form.

For example, consider the following XBRL definition relations that represent "general-special" relations between concepts in order to assist users creating extension taxonomies and software engineers to assist in the process of using the right line items within the right associations within a financial report.



And so, the proper use of "type-subtype" or "general-special" relations or "wider-narrower" relations are necessary to create quality financial report scheme relations and likewise financial reports that are correctly represented per that financial reporting scheme.

Controlling Logical System and Keeping it Properly Functioning

If a process cannot be controlled then the process simply cannot repeatedly and reliably output high-quality. If process output is not high-quality, automation cannot possibly be effective.

So, control of a process is necessary in order for the process to be effective. How do you control a process? You control a process using rules. Manual processes are controlled by rules that are read by humans. Automated processes are controlled by rules that are readable by both machines (i.e., to execute the process) and humans (i.e., to make sure the rules are right).

Who creates these machine-readable rules that are used to control processes that yield effective automation? Accountants must create these rules because the rules tend to be accounting oriented. Technical rules tend to relate to syntax and such technical rules can be completely hidden from business professionals which.

All the examples work the same and distill down to what can be described by the statements of a financial report logical system. All such logical systems work the same regardless of the number of terms, associations, structures, rules, and facts. The best example to describe the functioning of the system is the "Slightly More Complex, but still Basic Model Example" (FASB's SFAC 6 Elements of Financial Statements) because it is small enough to still get your head around but big enough to see what causes the logical system to be properly functioning, what causes the logical system to function improperly, and how to distinguish the difference[120].

Control of a system is described by classical control theory[121]. Systems can be open or systems can be closed. Advantages of closed systems is better control, stable performance, and guaranteed performance. Control of a financial report system and being able to define proper functioning system and keep such systems in control is desirable.

Based on all the things that can go wrong with the system, the following is the set of specific characteristics that can be employed to control the logical system and keep it properly functioning:

---

[120] YouTube.com, Digital Financial Reporting, *Distinguishing Between Properly and Improperly Functioning Logical System*, https://youtu.be/MFxStNn1Tjw
[121] Wikipedia, *Classical Control Theory*, https://en.wikipedia.org/wiki/Classical_control_theory

- Using the notion of "report element categories"
- Used the report element categories and organized them consistent with a set of strict "model structure rules"
- Used “derivation rules” (I used to call these impute rules) to overcome unreported financial report line items
- Used "consistency rules" to overcome contradictions or inconsistencies in reported facts
- Used "reporting styles" to facilitate model variability. (i.e. set of permissible models)
- Explicitly named “disclosures” so that they can be referred to.
- Using the notion of “information model” and "concept arrangement patterns".
- Using the notion of “disclosure mechanics rules” to specify the proper representation of a specific disclosure.
- Using the notion of “type-subtype” or “wider-narrower” or “general-special” relations to explicitly represent these relations.
- Using the notion of “mapping rules” to explicitly represent certain specific relations.
- Using the notion of “disclosure rule” or “reporting checklist” specifies the circumstances when each specific disclosure is required to be reported.

Use of these characteristics to control the logical system is demonstrated by the most current SFAC 6 *Elements of Financial Statements* representation in XBRL[122] and explained in the document *Impediments to Creating Properly Functioning XBRL-based Reports*[123]. Details are explained in the video, *Compensating for US GAAP and IFRS XBRL Taxonomy Design Choices*[124].

As such, it was these specific features which are included in the Standard Business Report Model (SBRM)[125] in order to control a business report logical system to keep that system properly functioning.

Finally, in order to test 100% of the information model patterns that would exist within such a system and to prove that each information model pattern functioned as expected and interacted properly with other information model patterns, a proof was created as a comprehensive test[126].

[122] SFAC 6 Elements of Financial Statements Representation in XBRL, http://xbrlsite.azurewebsites.net/2020/core/master-sfac6/
[123] Charles Hoffman, *Impediments to Creating Properly Functioning XBRL-based Reports (SFAC 6)*, http://xbrlsite.azurewebsites.net/2020/core/master-sfac6/Documentation.pdf
[124] *Compensating for US GAAP and IFRS XBRL Taxonomy Design Choices*, https://youtu.be/sKs02VjFJgw
[125] SBRM Progress Report, http://xbrl.squarespace.com/journal/2020/1/30/sbrm-progress-report.html
[126] Charles Hoffman, CPA, Digital Financial Reporting Proof of Semantics, http://xbrlsite.azurewebsites.net/2020/core/master-proof/Proof.pdf

Empirical Evidence

When Rene van Egmond and I first created the *Financial Report Semantics and Dynamics Theory*[127] back in 2012 we offered a proof that provided empirical evidence for that theory. Today, we can offer an improved proof based on 10 years of empirical evidence.

There are two similar, but separate, sets of XBRL-based reports that are used to prove that the logical theory of an XBRL-based report works as is expected.

The **first set** is a set of 10-K and 10-Q XBRL-based financial reports of 5,716 public companies that have been submitted to the U.S. Securities and Exchange Commission and are all publicly available[128]. These were used to test the fundamental accounting concept relations of the financial reports.

The **second set** is the last 10-K financial report of 5,555 public companies that have been submitted to the U.S. Securities and Exchange Commission and are likewise all publicly available[129]. These were used to test the disclosure mechanics and reporting check list of each report.

The **first set** shows that of the 5,716 reports:

- Over 99.9% of all reports were valid XBRL technical syntax.
- 99.24% (124,790 associations) of all fundamental accounting relations were consistent with expectation.
- .76% (962 associations) were not consistent with expectation and each of the errors was manually examined and determined to be an error in the facts reported by the public company[130].
- 89.1% of all reports were 100% consistent with each of the fundamental accounting concept relations rules.

Excel-based extraction tools were created for 4,060 reports or 68% so anyone can rerun these tests[131].

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

[128] *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

[129] Last 10-K submitted to SEC by public companies as of March 31, 2019, http://www.xbrlsite.com/site1/2018/10k/rss.xml

[130] Negative results from tests, http://xbrlsite.azurewebsites.net/2019/Library/2019-03-31_FAC-ErrorDetails.zip

[131] Excel-based extraction tool, http://xbrl.squarespace.com/journal/2018/1/11/further-updated-and-expanded-xbrl-based-financial-report-ext.html

For this first set, there are exactly six causes of errors and each error has a specifically identifiable task that would cause the error to be corrected and then be consistent with expectation:

1. **Fact** error in report. A report contained one or more errors in the facts reported within the report. To make this logical system consistent, the fact in the report simply needs to be corrected.
2. **Rule** error in knowledge base**.** While we are unaware of any rule errors in the knowledge base containing rules (i.e. because all such errors were fixed because they were under our control); if there were an error in the rule used to test facts, the rule would be in error. To make this logical system consistent, the rule in the knowledge base simply needs to be corrected.
3. **Association** error in knowledge base. A report contained one or more association errors in either the base taxonomy or the extension taxonomy. To make this logical system consistent, the association simply needs to be corrected.
4. **Structure** error in knowledge base (i.e. reporting style used is incorrect). A report could use the wrong structure (reporting style) to evaluate the report. To make this logical system consistent, the structure (reporting style) simply needs to be corrected.
5. **Rules engine** error. The rules engine used to process the report and test its facts against the knowledge base could be flawed. To make this logical system consistent, the rules engine algorithms simply need to be corrected.
6. **Structure** missing (i.e. reporting style does not exist). A report could be unique and a reporting style does not exist for the report. To make this logical system consistent, a new structure (reporting style) simply needs to be added and then used by the report.

Once the terms, associations, structures, assertions, and facts are brought into equilibrium for a report; then the report would be consistent and a properly functioning logical system. This process is repeated for each report. Then, the entire system of reports is properly functioning.

For the **second set**, there are more possibilities for inconsistencies and only approximately 68 disclosures were tested in each 10-K of the anticipated perhaps 500 to 1,500 possible disclosures. So, the testing is not as complete. And, the testing is not based on sound statistical testing so I cannot say that a sampling of disclosures was tested. However, there is no evidence to lead me to believe that I am missing something important. And so, what testing was done did show that, similar to the first set, there are specifically identifiable errors and specifically identifiable tasks that would cause the errors to be corrected and then cause the report fact to be consistent with the knowledge base. The categories of error are very similar and so they will not be repeated here. Simply identifying and correcting mistakes would make the system a properly functioning system.

Lean Six Sigma

This method leverages Lean Six Sigma principles, philosophies, and techniques. Lean Six Sigma[132] is a discipline that combines the problem-solving methodologies and quality enhancement techniques of Six Sigma[133] with the process improvement tools and efficiency concepts of Lean Manufacturing[134]. 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[135] 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[136] 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[137].

Fundamentally, the objective of the *Seattle Method* is to reduce financial report defect rates from, say, sigma level three which has a defect rate of 6.7% or about 67,000 defects per opportunity to a sigma level six level which has a defect rate of 0.00034% or about 3.4 defects per opportunity[138].

Implementations

The first implementation of this method was by myself using Microsoft Excel, Microsoft Access, and two XBRL processors.

The second implementation, the first commercial implementation, was XBRL Cloud[139] which I made use of for my quarterly quality checks of XBRL-based reports submitted to the SEC[140].

---

[132] Wikipedia, *Lean Six Sigma*, https://en.wikipedia.org/wiki/Lean_Six_Sigma
[133] Wikipedia, *Six Sigma*, https://en.wikipedia.org/wiki/Six_Sigma
[134] Wikipedia, *Lean Manufacturing*, https://en.wikipedia.org/wiki/Lean_manufacturing
[135] YouTube.com, *Lean Six Sigma in 8 Minutes*, https://youtu.be/s2HCrhNVfak
[136] YouTube.com, *Six Sigma in 9 Minutes*, https://youtu.be/4EDYfSl-fmc
[137] Charles Hoffman, CPA, *Lean Six Sigma*, http://www.xbrlsite.com/mastering/Part01_Chapter02.K_LeanSixSigma.pdf
[138] Wikipedia, Six Sigma, Sigma Levels, https://en.wikipedia.org/wiki/Six_Sigma#Sigma_levels
[139] XBRL Cloud Clean Score, https://www.xbrlcloud.com/cleanscore.html
[140] *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

The third implementation, a commercial implementation, was 28msec[141], which was a NOSQL database that used my model as the database schema and they loaded the entire SEC EDGAR system (10-Ks and 10-Qs) into that database. 28msec seems to have gone defunct.

The fourth implementation, a working proof of concept, was Pesseract[142] which is a Windows Forms application and was created by a software engineer and myself. Pesseract could be used to verify reports, but not create reports.

The fifth implementation, a working proof of concept, was a free open-source tool I created using Microsoft Access[143] that could be used to generate an XBRL report, report model, and Inline XBRL. (This has been downloaded over 100 times.)

The sixth implementation, a working proof of concept, was a Windows Forms application called Luca[144] which could be used to create XBRL-based reports and report models.

The seventh implementation, a commercial implementation by Auditchain which they call Pacioli[145], which is hands down the best implementation of this method yet. Pacioli uses a logic/reasoning/rules engine for the first time; in this case PROLOG is used. Further, Auditchain has made some enhancements to the method.

The eighth implementation, which will very likely become commercial software, was created by a software engineer and is a cloud-based version of Luca[146].

The ninth implementation, is another version of Luca created by Auditchain which is near commercial quality[147].

All of these implementations are interoperable at both the syntax level and the semantics level.

What I hope to be able to show sometime is a commercial implementation of the entire record to report process that I have prototyped[148].

---

[141] 28msec, https://twitter.com/28msec?lang=en
[142] Pesseract, http://pesseract.azurewebsites.net/
[143] 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
[144] Luca, http://xbrl.squarespace.com/journal/2020/9/15/luca.html
[145] *Auditchain Protocol Launches on Testnet*, http://xbrl.squarespace.com/journal/2021/10/21/auditchain-protocol-launches-on-testnet.html
[146] *Cloud-based Luca*, http://luca.yaxbrl.com/
[147] *Getting Started with Auditchain Luca*, https://digitalfinancialreporting.blogspot.com/2024/01/getting-started-with-auditchain-luca.html
[148] *Effective Automation of Record to Report Process (Iteration #4)*, http://xbrl.squarespace.com/journal/2021/1/25/effective-automation-of-record-to-report-process-iteration-4.html

Learning XBRL-based Financial Reporting[149] provides a set of dashboards that contain example reports and example rules that help you understand how to use the Seattle Method and what it does. An excellent way to understand the details is to reverse engineers what you see in the examples. There are four specific dashboards:

- **Build up**: Starts with a very small reporting scheme, the accounting equation, and then incrementally builds up to the Microsoft 10-K.
- **Verification tasks**: Starts with XBRL technical syntax verification and then walks you through each of the different verification tasks.
- **Common errors**: Provides an example of each type of common error and shows how the error is detected.
- **DOW 30**: Provides XBRL technical syntax, model structure, and fundamental accounting concepts verification for each of the DOW 30 companies.

Build up dashboard[150]:

#	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

Verification tasks dashboard[151]:

[149] *Learning XBRL-based Financial Reporting*, http://xbrl.squarespace.com/journal/2022/2/9/learning-xbrl-based-digital-financial-reporting.html
[150] Build up dashboard, http://xbrlsite.azurewebsites.net/2022/Prototypes/buildup/Dashboard.html
[151] Verification dashboard, http://xbrlsite.azurewebsites.net/2022/Prototypes/proof/Dashboard.html

#	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 (No XBRL verification, no other verification)	PROOF	Info			Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	NONE
2	GH259400TOMPUOLS65II (With XBRL verification, no XBRL calculations verification, no XBRL Formulas verification, no other verification)	PROOF	Info			OK	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	NONE
3	GH259400TOMPUOLS65II (No XBRL formulas, no model structure, no fundamental accounting concepts, no type-subtype, no disclosure mechanics, no reporting checklist Verification)	PROOF	Info			OK	OK	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	NONE
4	GH259400TOMPUOLS65II (No model structure, no fundamental accounting concepts, no type-subtype, no disclosure mechanics, no reporting checklist Verification)	PROOF	Info			OK	OK	OK	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	NONE
5	GH259400TOMPUOLS65II (No fundamental accounting concepts, no type-subtype, no disclosure mechanics, no reporting checklist Verification)	PROOF	Info			OK	OK	OK	OK	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	NONE
6	GH259400TOMPUOLS65II (No type-subtype, no disclosure mechanics, no reporting checklist Verification)	PROOF	Info			OK	OK	OK	OK	OK	UNKNOWN	UNKNOWN	UNKNOWN	NONE
7	GH259400TOMPUOLS65II (No disclosure mechanics, no reporting checklist Verification)	PROOF	Info			OK	OK	OK	OK	OK	OK	UNKNOWN	UNKNOWN	NONE
8	GH259400TOMPUOLS65II (No reporting checklist Verification)	PROOF	Info			OK	OK	OK	OK	OK	OK	OK	UNKNOWN	NONE
9	GH259400TOMPUOLS65II (With Full Battery of Seattle Method Verification)	PROOF	Info			OK	OK	OK	OK	OK	OK	OK	OK	NONE

Components of a Knowledge Based System

Wikipedia defines a knowledge-based system as follows[152]:

> "A knowledge-based system is a computer program that reasons and uses a knowledge base to solve complex problems."

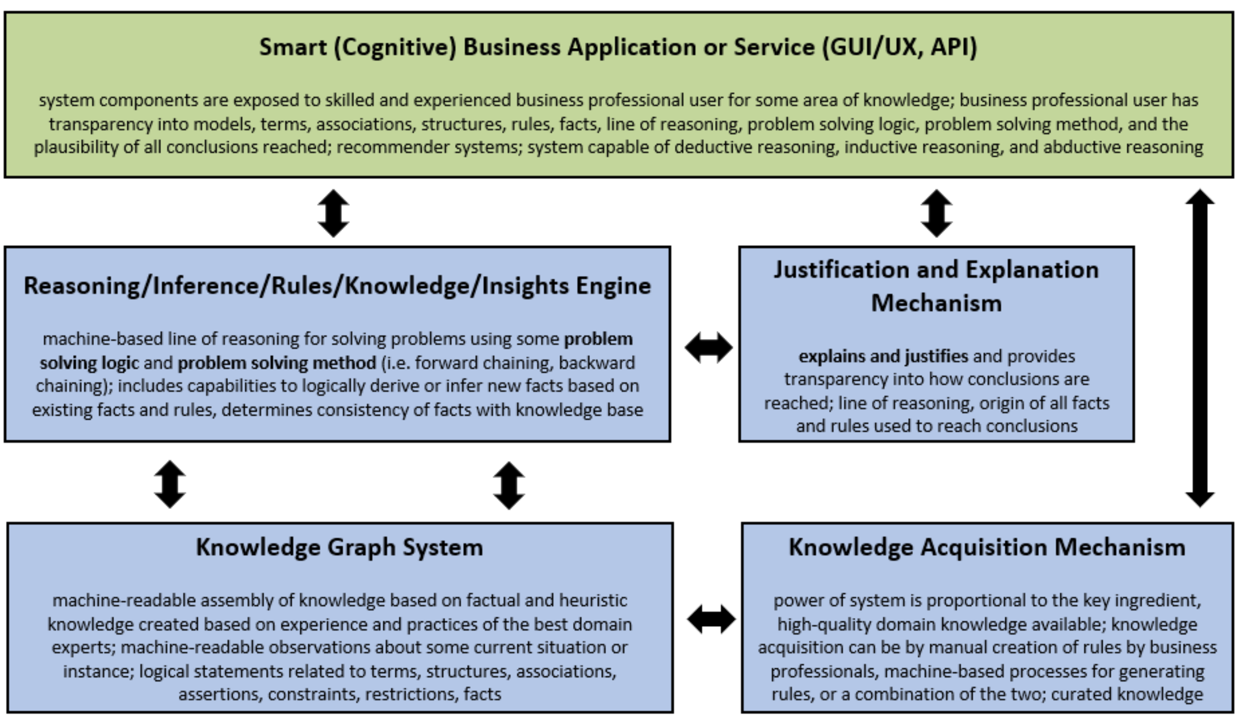
To create a knowledge based system, information is acquired from skilled, knowledgeable professionals. This information is stored in a knowledge graph (knowledge base and a fact database). The system uses a reasoning/inference/rules engine to process the knowledge. Some sort of justification and explanation mechanism helps users understand the line of reasoning used to reach conclusions. The system then presents that information to the business professional using the system using some interface. Nothing is a "black box". The origin of information used to reach conclusions is always apparent to the users of the application. The following describes each of those knowledge-based system components:

- **Knowledge acquisition mechanism**: Somehow knowledge needs to be acquired and put into the knowledge-based system.

[152] Wikipedia, Knowledge-based System, https://en.wikipedia.org/wiki/Knowledge-based_systems

- **Knowledge graph**: Somehow the knowledge acquired needs to be stored in machine-readable form such that it can be used by the system. A knowledge graph stores a database of knowledge which includes logical statements about models, structures, terms, rules, associations, and facts.
- **Reasoning/inference/rules engine**: Some rules engine is necessary to process the knowledge and facts. Deductive reasoning is essential; inductive reasoning is a nice-to-have.
- **Justification and explanation mechanism**: Nothing in the system should be a black box. Users of the system must be able to understand the origin of information (providence) and there needs to be an audit trail to understand every decision made and the reasoning behind the system.
- **Business professional user interface**: Business professionals need to interact with the system to be able to perform work on their terms. Technical complexity must be buried deep within the application, business professionals don't care about technical details. Domain complexity is what users should be working with.

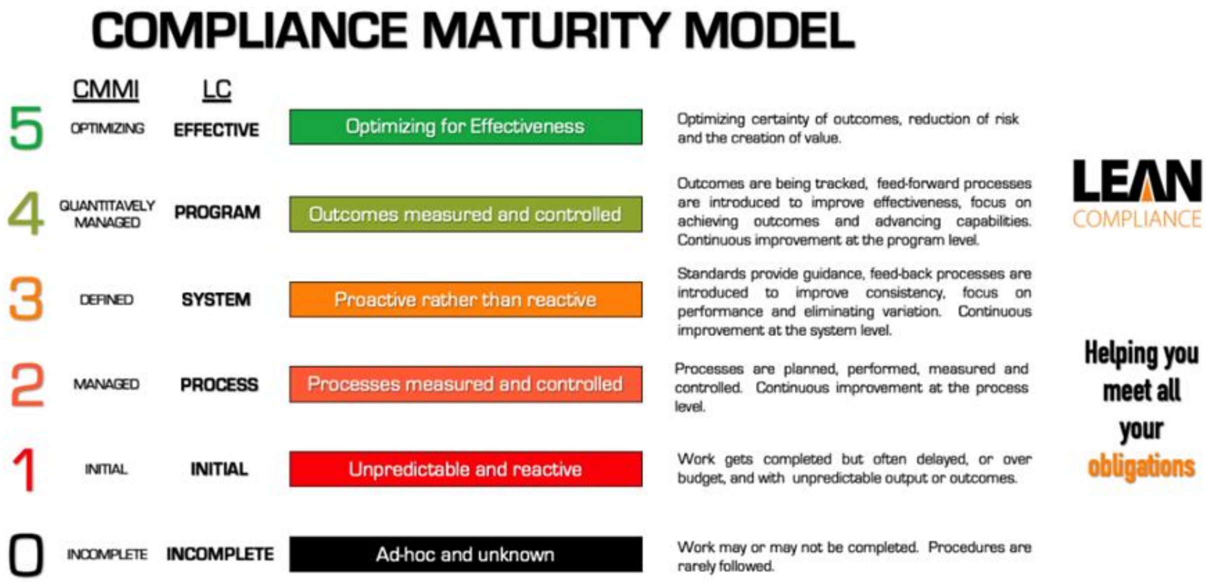
The following graphic provides a visual summary of the components of a knowledge-based system[153]:



[153] *Smart (Cognitive) Business Applications and Services (Work in Progress)*, https://digitalfinancialreporting.blogspot.com/2024/02/smart-cognitive-business-applications.html

Compliance Maturity Model

The *Seatle Method* is a system that enables the creation of a program per the *Compliance Maturity Model*[154] which is an application of the *Capabilities Maturity Model*[155].



Conclusion

If a process cannot be controlled then the process simply cannot repeatedly and reliably output high-quality information. If process output is not high-quality information, automation cannot possibly be effective and therefore effective exchange of information cannot possibly occur.

So, control of a process is necessary in order for the process to be effective and for the process to be repeatable. How do you control a process? You control a process using rules. Manual processes are controlled by rules that are read by humans and then humans figure out if the process is working properly. Automated processes are controlled by rules that are readable by both machines (i.e., to execute the process) and humans (i.e., to make sure the rules are right).

Who creates these machine-readable rules that are used to control processes that yield effective automation? For financial reporting, accountants must create these rules because the rules tend to be accounting oriented. Technical rules tend to relate to syntax and such technical rules can be hidden from business professionals. What is left is the business logic and accounting rules that are used to control information and control process workflow. As such, the creation of machine-readable rules must be "self-service". Business professionals must be

[154] LinkedIn, Raimund Laqua, PMP, Peng, *Compliance Maturity Model*, https://www.linkedin.com/pulse/capabilities-maturity-model-compliance-raimund-laqua-pmp-peng/
[155] Wikipedia, *Capabilities Maturity Model*, https://en.wikipedia.org/wiki/Capability_Maturity_Model

empowered to create, adjust, maintain, and otherwise manage the rules that are used to control and therefor effectively automate processes. Once you have the machine-readable rules, you need software that can process the rules; this is sometimes called a rules engine or reasoning engine or a semantic reasoner.

The Department of Philosophy of Texas State provides this excellent differentiation between a condition that is *necessary* and a condition that is *sufficient*[156]:

> A **necessary condition** is a condition that must be present for an event to occur. A **sufficient condition** is a condition or set of conditions that *will produce the event*. A necessary condition must be there, but it alone does not provide sufficient cause for the occurrence of the event. Only the sufficient grounds can do this. In other words, all of the necessary elements must be there.

To effectively communicate the meaning of financial statements where you have complicated information and the financial report creators are permitted to modify the report model it is **necessary** to:

- Agree on a specific common shared background knowledge.
- Agree on a specific common shared inference logic.
- Agree on a specific common shared world view.
- Agree to extend the common background knowledge terms, associations, structures, and rules in understood and permissible ways.
- Communicate the semantics of facts using the above agreed specific items.
- Physically transport those logical statements (machine-readable structures, terms, associations, rules, facts) using some syntax effectively.
- Prove that the logical statements are consistent, complete, precise and therefore that the financial statement is a properly functioning logical system.

In such financial reports, there is a specific knowable set of things that can go wrong that can be verified as to being correct or incorrect using automated machine-based processes. That set of things that can go wrong are:

1. Incorrect XBRL technical syntax or report semantics (i.e., anything verifiable per the XBRL technical specification rules).
2. Incorrect XBRL presentation associations. (not in the scope of XBRL rules)
3. Impermissible or inconsistent or contradictory fundamental accounting concept relations. (not in the scope of XBRL rules)

[156] Texas State, Department of Philosophy, *Confusion of Necessary with a Sufficient Condition*, https://www.txstate.edu/philosophy/resources/fallacy-definitions/Confusion-of-Necessary.html

4. Impermissible type-subtype associations. (not in the scope of XBRL rules)
5. Impermissible disclosure mechanical structures. (not in the scope of XBRL rules)
6. Impermissible set of information reported. (not in the scope of XBRL rules)

All six categories of verification are necessary for a financial report to be proven to be a properly functioning logical system that is complete and consistent. All six categories are necessary, but they might not be sufficient to verify everything about the financial report. To the extent other items can be verified using automated processes is the extent to which entire processes can be automated. All non-automatable tasks must be verified using human powered verification steps.

Any lack of agreement or flaws will require additional steps to be taken in order to effectively communicate the semantics of financial information and to use that communicated information effectively.

"Hope" and "wishful thinking" or "good intensions" are not sound engineering principles and will never help in achieving successful communication of semantic information. Effective engineering creates the possibility of successful communication of information.

Business professionals should not need to be concerned with the engineering details, they simply need to use the logic within their area of knowledge within the system and the system should be reliable and safe.

Empirical evidence, in my view, seems to prove what is necessary to exchange semantic information, the "words" and "numbers", contained in financial reports.

Professional accountants are responsible for creating "true and fair" financial reports whether those reports are human-readable or machine-readable XBRL. The *Venetian Method* is held out to be "good bookkeeping". The *Seattle Method* is held out to be "good XBRL-based reporting" and helps to adapt the Venetian Method for the information age[157] by helping to make that method digital.

This method carries with it its own proof; it tends to be self-evident once you are conscious of the moving pieces. It, like the *Venetian Method*, is a methodical and orderly system.

There are many signs that a de facto good practices industry standard digital general purpose financial report metamodel is emerging[158]. While there is no formally documented global standard XBRL-based financial report, there is movement towards a general theoretical

---

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

[158] *De Facto Good Practices Industry Standard Digital Financial Report Metamodel*, http://xbrl.squarespace.com/journal/2021/10/20/de-facto-good-practices-industry-standard-digital-financial.html

consensus. At a minimum, the *Seattle Method* is good information to bring to the table when discussing creating such a formal global standard.

Whether the *Seattle Method* is effectively employed for standards based general purpose financial reports is yet to be seen or realized. But, these ideas are just as applicable to general business reporting and information exchanges in other areas of knowledge.

Deductive logic is precise because it provides certainty; guaranteed. The machine-readable deductive rules provide a "template" for what a perfect/precise XBRL-based financial report looks like. It is to the extent that these rules are provided; it is to that extent that reports can be considered trustworthy. Valid reports (consistent with all the specified rules) that are also sound (a.k.a. precise, precisely follow real-world financial reporting rules and other logic); it is to that extent that intelligent software agents making use of such information can do so effectively. Full stop. No magic; just good engineering.

On this foundation of deductive logic; inductive reasoning and abductive reasoning capabilities can be built. Also, digital distributed ledger and NFT related capabilities can be built.

* * *

For more details, please refer to ***Mastering XBRL-based Digital Financial Reporting***[159].

[159] Charles Hoffman, CPA, *Mastering XBRL-based Digital Financial Reporting*, http://xbrl.squarespace.com/mastering-xbrl/