Logical Schema of Financial Reports

This document articulates the logical schema of a financial report and supporting knowledge that is used by software developers to create software such as an expert system that is used for creating financial reports.

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

March 10, 2022 (DRAFT)

http://xbrl.squarespace.com/

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

Executive summary:

- This document describes the logical schema of a financial report.
- Pacioli helps enforce that the details of a financial report fit into the logical schema of a financial report.
- An expert system for creating financial reports does not let you violate the logical schema of a financial report.
- If the logical schema of a financial report is correct, generating any physical technical format can be performed safely and reliably.

XBRL is a technical format which has a physical format the syntax of which is enforced by a physical schema[1]. The XBRL Specifications[2] specifies the technical format, the physical schema. W3C standards such as XML Schema[3], XLink[4], Extensible Markup Language (XML)[5], XPath[6], and so forth. For the full set of technical formats that are used to specify the physical schema of XBRL see the References section of the XBRL Specification[7].

A financial report is a man-made idea, a logical conceptualization. That logical conceptualization can be represented by a logical schema[8]. That logical schema can be expressed in human readable statements such as in the *Logical Theory Describing Financial Report*[9]. Theories can be right or wrong, but the point of a theory is to express the essence of something.

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 a shared understanding of some area of knowledge. A stakeholder is anyone that has a vested interest in a logical system. Foundational to arriving at harmony between the stakeholders of a system is having a common logical conceptualization for thinking about and discussing the system.

A financial report is a specialization of the much more general category of a report, the general business report. The forthcoming *Standard Business Report Model (SBRM)* specification [10] is an industry standard that formally documents a logical conceptualization of a business report in both human readable and machine-readable models. This enables a machine-readable report represented using any technical syntax, like XBRL, JSON (JavaScript Object Notation) and RDF (Resource Description Format) or others, to be tested to see if it is consistent with that logical conceptualization using reliable automated processes.

This document articulates the logical conceptualization and logical schema of a financial report for the purposes of creating an expert system which works within the bounds of that logical schema to create high-quality financial reports. Then, that logical conceptualization can be serialized to XBRL or other physical formats.

---

[1] Wikipedia, Physical Schema, https://en.wikipedia.org/wiki/Physical_schema
[2] XBRL International, XBRL Specifications, https://specifications.xbrl.org/specifications.html
[3] W3C, XML Schema, https://www.w3.org/XML/Schema
[4] W3C, XLink, https://www.w3.org/TR/xlink11/
[5] W3C, Extensible Markup Language (XML), https://www.w3.org/TR/REC-xml/
[6] W3C, XPath, https://www.w3.org/TR/2010/REC-xpath20-20101214/
[7] XBRL International, XBRL Specification, References, https://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_6
[8] Wikipedia, Logical Schema, https://en.wikipedia.org/wiki/Logical_schema
[9] Charles Hoffman, CPA, *Logical Theory Describing Financial Report*, http://xbrlsite.com/seattlemethod/LogicalTheoryDescribingFinancialReport_Terse.pdf
[10] OMG, Standard Business Report Model (SBRM) Specification, https://www.omg.org/intro/SBRM.pdf

Knowledge Based Systems

The document *Expert System for Creating Financial Reports*[11] outlines the vision of a rules-based expert system for creating financial reports that leverages explainable artificial intelligence (XAI)[12]. This vision is based on working software applications, working proof of concepts, reverse engineering of XBRL-based financial reports submitted to the SEC.

Artificial Intelligence

There are two approaches to artificial intelligence and, as I have said before, the right approach should be used for the given job[13]. The two approaches are:

- **Rules-based systems** (expert systems, three basic types)
  - **Classification or diagnosis type**: helps users of the system select from a set of given alternatives.
  - **Construction type**: helps users of the system assemble something from given primitive components.
  - **Simulation type**: helps users of the system understand how some model reacts to certain inputs.
- **Patterns-based systems** (machine learning which can be supervised or unsupervised, five basic types[14])
  - **Clustering algorithms**: categorize or group things
  - **Explanatory algorithms**: explain the relationships between variables
  - **Ensemble learning algorithms**: use multiple models
  - **Similarity algorithms**: compute the similarity of pairs of things
  - **Dimensionality reduction algorithms**: reduces variables in a dataset

Another helpful breakdown of pattern-based systems is provided by the article *Top Machine Learning Algorithms for Prediction: A Short Summary*[15].

---

[11] Charles Hoffman, CPA, *Expert System for Creating Financial Reports*, http://xbrlsite.azurewebsites.net/2022/Library/ExpertSystemForCreatingFinancialReports.pdf

[12] *Effective Automated Information Exchange and Explainable AI (XAI)*, http://xbrl.squarespace.com/journal/2022/3/21/effective-automated-information-exchange-and-explainable-ai.html

[13] *Use the Right Artificial Intelligence Approach for the Job*, http://xbrl.squarespace.com/journal/2019/7/12/use-the-right-artificial-intelligence-approach-for-the-job.html

[14] EDUCBA, Machine Learning Models, https://www.educba.com/machine-learning-models/

[15] Medium, *Top Machine Learning Algorithms for Prediction: A Short Summary*, https://medium.com/@webadmin_46735/top-machine-learning-algorithms-for-predictions-a-short-overview-5ed1ff6942ff

Name	Type	Description	Advantages	Disadvantages
Linear Regression		-The best fit line through all data points	-Easy to understand -you can clearly see what the biggest drivers odf the model are.	-sometimes to simple to capture cpmöex relationships between variables, -Tendency für the model to overfit.
Logistic Regression		-The adoption for linear regression to problembs of classification	-Easy to understand	-sometimes to simple to capture cpmöex relationships between variables, -Tendency für the model to overfit.
Decision Tree		-A graph that uses branching method to match all possible outcomes of a decision	-Easy to understand and implement.	-Not often use of ist own for prediction because it's also often too simple and not powerful enough for complex data.
Random Forest		- Takes the average of many decision trees. Each tree is weaker than the full decision tree, but combining them we get better overall performance.	-A sort of „wisdom of the crowd", Tend to result in very high quality results. -Fast to train	-Can be slow to output predictions relative to other algorithms. -Not easy to understand predictions.
Gradient Boosting		-Uses even weaker decision trees that increasingly focused on „hard examples"	-High-performing	-A small change in the future set or training set can create radical changes in the model. -Not easy to understand predictions.
Neural Networks		-Mimics the behaviour of the brain. NNs are interconnected Neurons that pass messages to each other. Deep Learning uses severak layers of NNs to put one after the other.	-Can handle extremely complex tasks. No other alsgorithm comes close in image recognition.	-very very slow to train. Because they have so many layers. Require a lot of power. -Almost impossible to understand predictions.

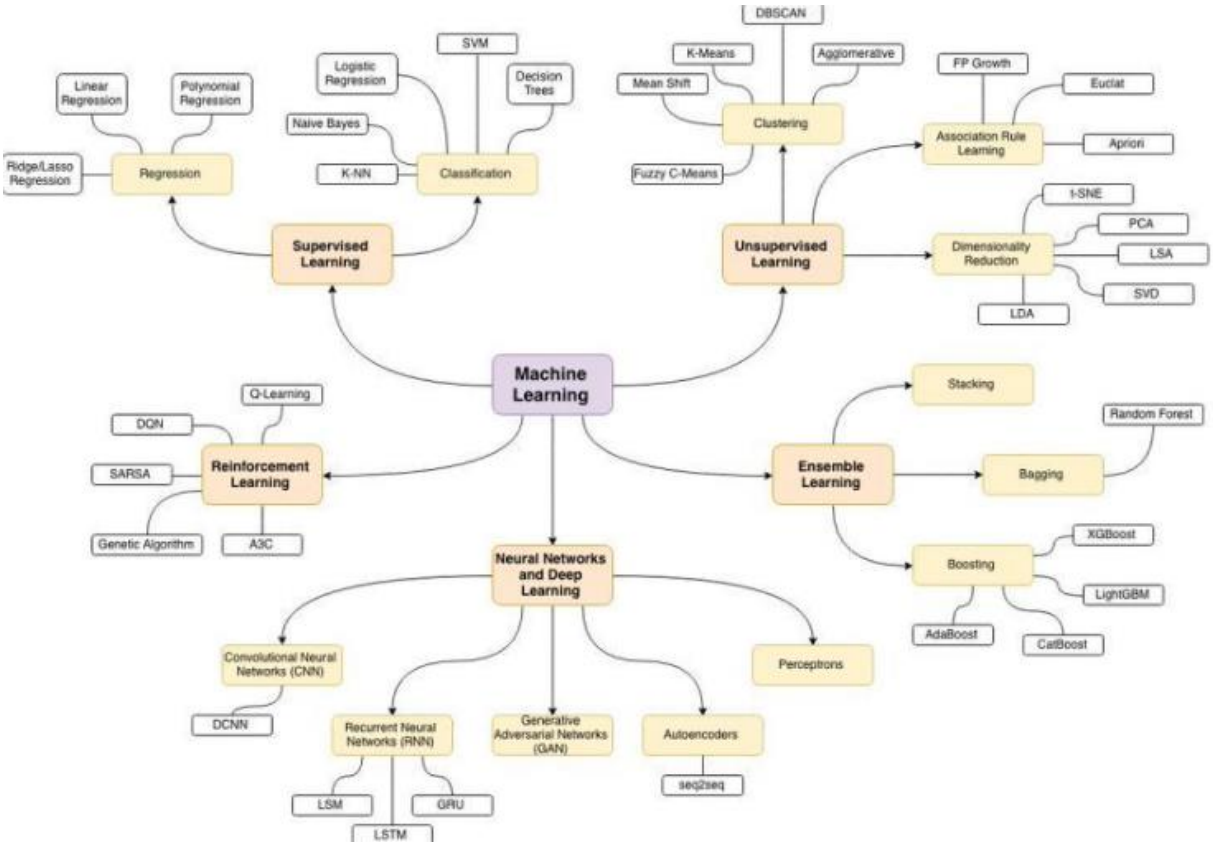
Currently, the expert systems working proof of concepts and commercial software that I have created and experimented with only leverage rules-based systems. But it is pretty clear that machine learning offers additional capabilities that will be inevitably leveraged.

Pacioli[16] is a logic/rules/reasoning/knowledge/insights engine. Frankly, I don't know exactly what to call Pacioli. At first, I called Pacioli a logic/rules/reasoning engine. Then I referred to it as a knowledge engine[17]. Now I am contemplating that Pacioli is an insights engine. For now, I consider Pacioli a logic/reasoning/rules/knowledge/insights engine in order to be complete, I guess.

Pacioli has three roles. The first role of Pacioli is to enforce the financial report logical schema and return information about where a financial report might be violating that logical schema. This is helpful in the process of creating XBRL-based financial reports.

The second role of Pacioli is to leverage the machine-readable logical schema to work with the information that makes up the financial report knowledge graph[18].

The third role of Pacioli, which similarly leverages the machine-readable logical schema and financial report knowledge graph is effectively extracting information for financial analysis.



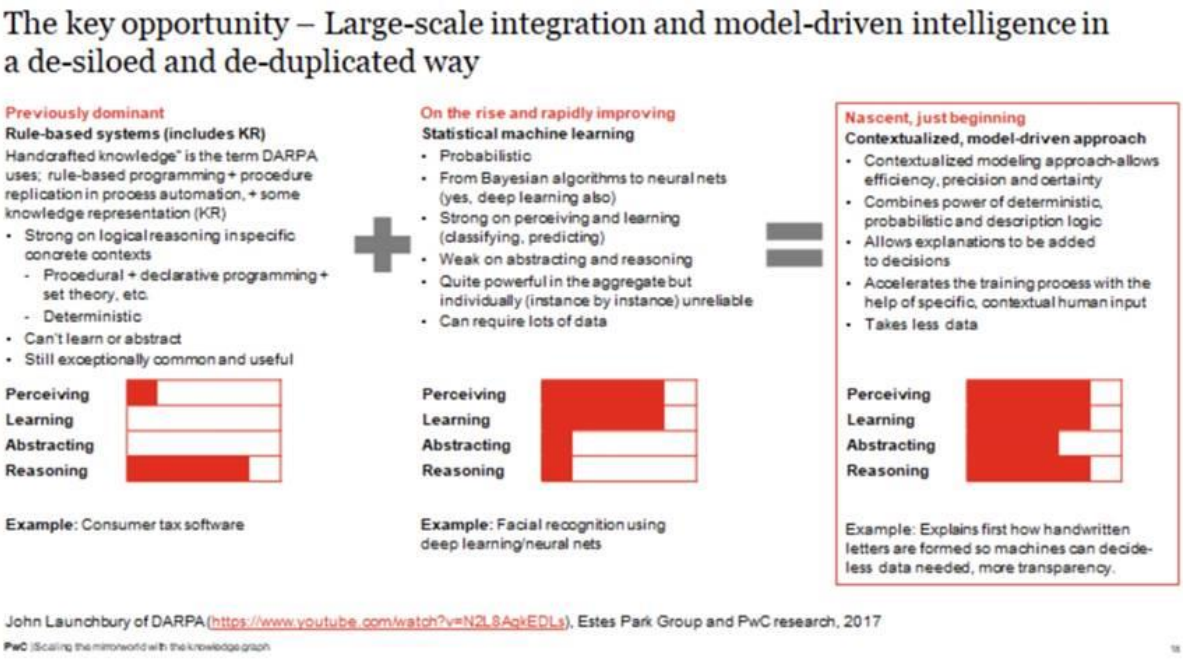
[16] Auditchain, *Pacioli Logic and Rules Engine*, https://docs.auditchain.finance/auditchain-protocol/pacioli-logic-and-rules-engine

[17] *Pacioli: an XBRL Knowledge Engine*, http://xbrl.squarespace.com/journal/2022/2/19/pacioli-an-xbrl-knowledge-engine.html

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

Programmed Ability to Process Information

DARPA and PWC do an excellent job of explaining the capabilities of artificial intelligence. This presentation[19], video[20] and this article[21] provide a summary worth reading.



The graphic above shows three approaches to implementing artificial intelligence or the programmed ability to process information. Auditchain's Pacioli is a "rule-based system" that uses "handcrafted knowledge" that I created (the LEFT SIDE). Statistical machine learning will be added (the MIDDLE) to Pacioli over time most likely. Ultimately, where Auditchain and others will end up is on the RIGHT side which is the combined capabilities of rules-based systems and statistical machine learning. I suspect that this will occur within three to five years.

Here is another explanation that was inspired from a DARPA presentation[22]. Artificial intelligence is defined as a "**programmed ability to process information**". Sure, software like, Excel and Word can process "stuff"; but they don't really understand anything about accounting, reporting, auditing, or analysis. By "understand", this is what is meant, paraphrasing form the DARPA explanations of these terms:

- **Perceive**: ability to perceive rich, complex, subtle information.

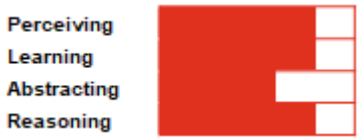
[19] DARPA, *A DARPA Perspective on Artificial Intelligence*, https://www.darpa.mil/attachments/AIFull.pdf
[20] YouTube.com, *A DARPA Perspective on Artificial Intelligence*, https://youtu.be/-O01G3tSYpU
[21] Quora, Alan Morrison, *What is the relation between Semantic Web and AI?*, https://www.quora.com/What-is-the-relation-between-Semantic-Web-and-AI/answer/Alan-Morrison?ch=2&oid=180785119&srid=Mru&target_type=answer
[22] DARPA, *A DARPA Perspective on Artificial Intelligence*, https://www.darpa.mil/attachments/AIFull.pdf

- **Learn**: ability to learn about that information within a specific environment such as within a financial report.
- **Abstract**: ability to create new meaning from existing information.
- **Reason**: ability to plan, ability to decide, ability to verify.

The following graph shows where most software currently is today in terms of helping accountants get their jobs done in terms of accounting knowledge (i.e. basically, you can think of this is the manual tasks and processes that is being done now)

But this is where something like Auditchain's Pacioli is in terms of perceiving, learning, abstracting, reasoning when it comes to the information within a financial report: (Pacioli can do this NOW using my "handcrafted knowledge" a.k.a. standard machine-readable rules represented using XBRL)

The next graphic shows where Pacioli might ultimately go when statistical machine learning is combined with the rules-based system. The rules-based system's handcrafted rules act as a starter set of the rules needed to make the statistical machine learning work. Basically, the rules-based system focuses on what it can do best and the statistical machine learning focuses on what it can do best and the result is a combination of the two approaches: (Auditchain will be here in three to five years in my view)

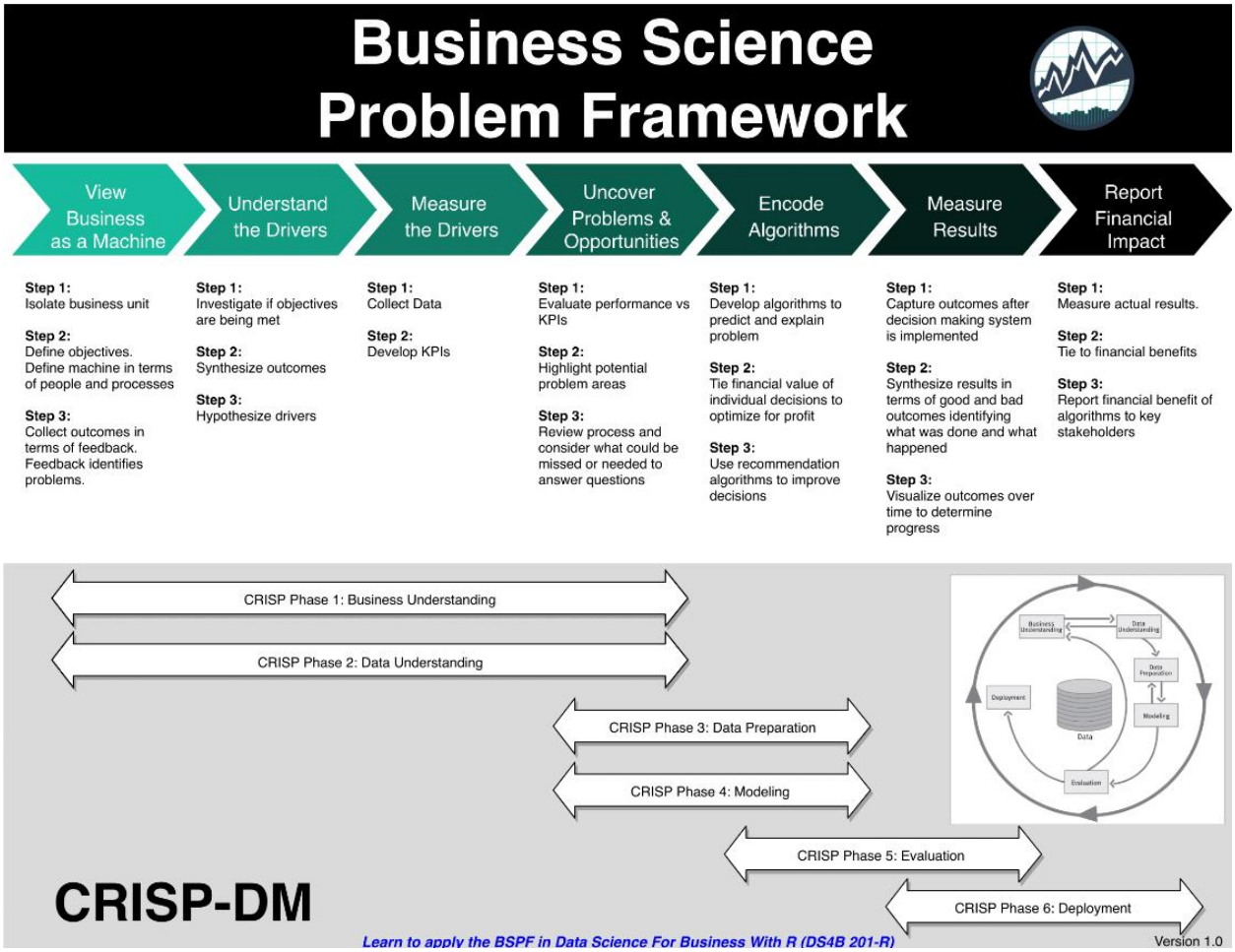


The final graphic is what full automation looks like. To understand what "full automation" is one would need to understand the specific details of what is being automated. But also keep in mind that while automation might be possible for some things, for other things it really is not realistic or even necessary to automate to derive value from automation. Making a process 50% more efficient or even 25% more efficient or only 10% more efficient is still quite valuable.

To summarize, there are three very important points to remember here. First, the right artificial intelligence should be used for each automation task. Second, handcrafted metadata is necessary to make automation actually work. Finally, while it is easy to discuss automation in general high level terms; when you actually automate something you need to work with the details of what is being automated.

Data Science and Business Science Connection

In the information age, data science[23] and business science[24] are connected. This graphic, the business science problem framework[25], helps one understand that connection.



[23] Data Science Cheat Sheets, https://github.com/FavioVazquez/ds-cheatsheets

[24] Business Science, https://www.business-science.io/bspf.html

[25] Business Science Problem Framework, https://raw.githubusercontent.com/FavioVazquez/ds-cheatsheets/master/Business_Science/img/Business_Science_Problem_Framework.png

Logical Statements

A system can be explained by a logical theory. A logical theory is an abstract conceptualization[26] 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.

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 statements used for capturing meaning or representing a shared understanding of and knowledge in some area of knowledge.

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

- **Logical theory**: A *logical theory* is a set of logical models that are consistent with and permissible per that logical theory.
- **Model**: A *model*[27] is a set of logical structures that are consistent with and permissible interpretations of that model.
- **Structure**: A *structure* is a set of logical statements which describe the structure.
- **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 theory relates. There are four broad categories of logical statements:
    - **Terms**: *Terms* are logical statements that define ideas used by the logical theory 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 what tend to be 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 business report. For example, the

[26] Wikipedia, *Conceptual Model*, https://en.wikipedia.org/wiki/Conceptual_model
[27] Wikipedia, *Model Theory*, https://en.wikipedia.org/wiki/Model_theory

financial report, a type of business 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.

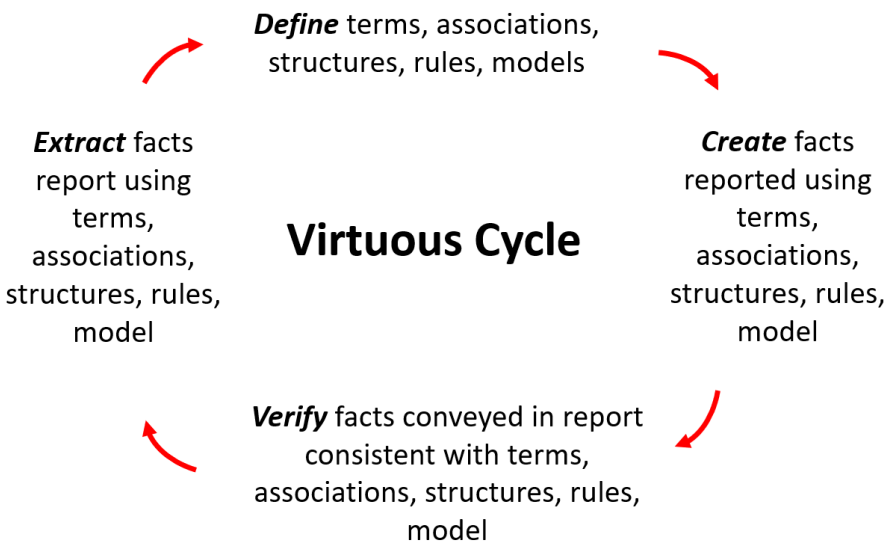
Fundamentally, a logical theory is a set of logical statements. Those logical statements can be represented in human-readable form or they could be expressed in machine-readable form. Once in machine-readable form, those logical statements can be interrogated using software applications. To the extent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

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 system is working right, it creates a virtuous cycle[28].

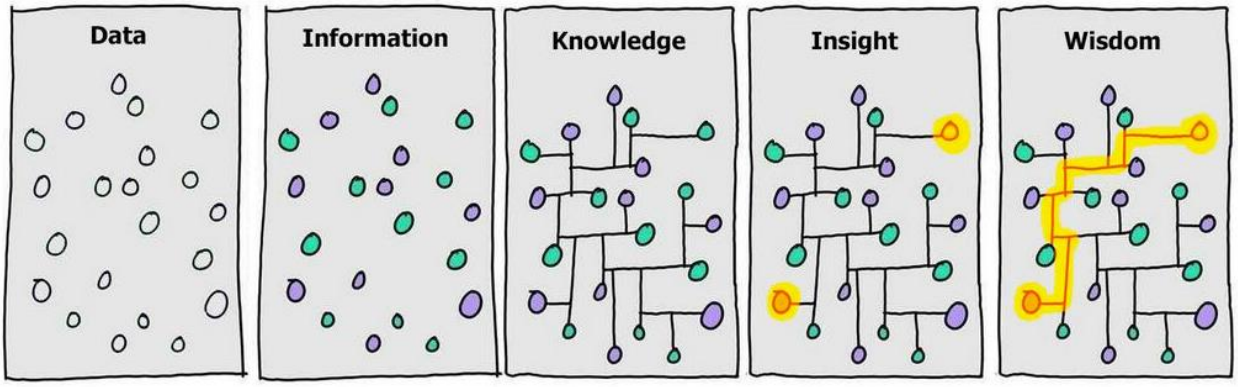


[28] Charles Hoffman, CPA, *Virtuous Cycle*, http://xbrl.squarespace.com/journal/2020/4/29/virtuous-cycle.html

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*[29].

Knowledge and Insight

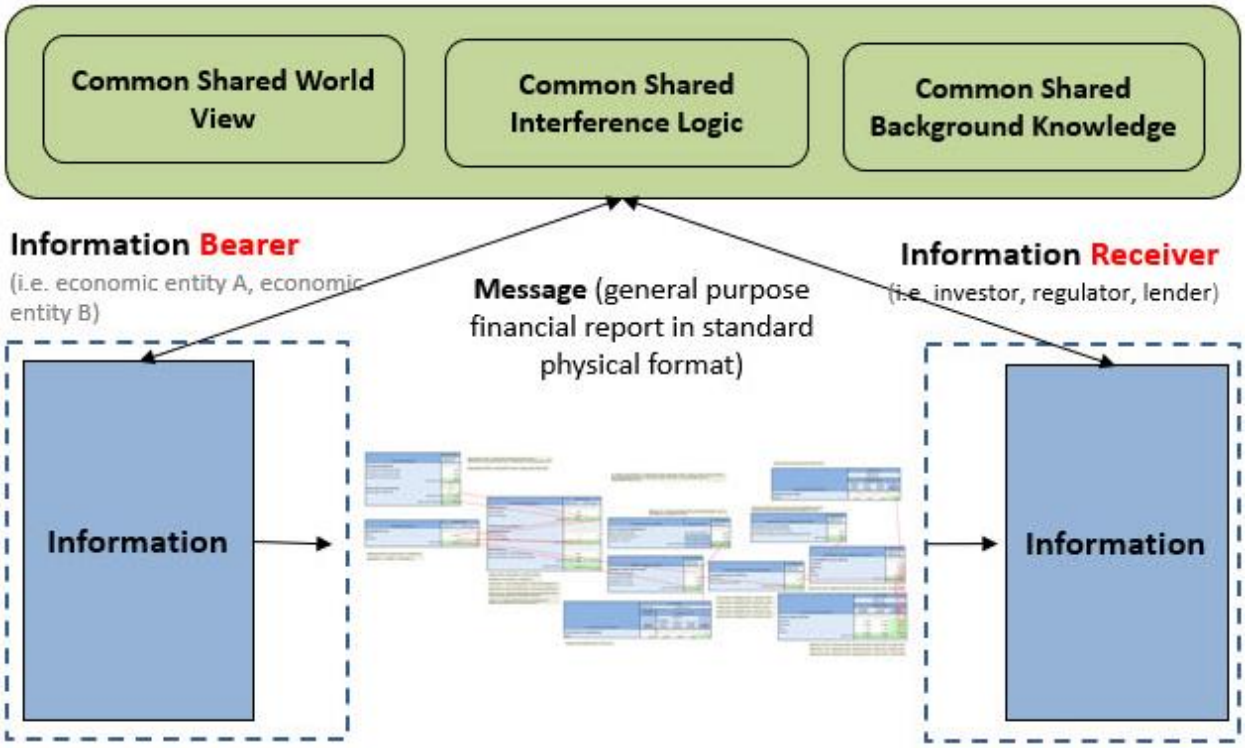
Fundamentally, all this distills down to the reality shown in the following two graphics. The first graphic helps one understand the objective which is insight and wisdom[30]:



Effective Information Exchange

The data and information have to be high-quality for a system to work. And to exchange data and information with other systems the information bearer (or creator) and the information receiver (or user) need to share a common world view, common inference logic, and common background knowledge as depicted in this graphic:

[29] Charles Hoffman, CPA, *Logical Systems*, http://www.xbrlsite.com/mastering/Part02_Chapter05.A_LogicalSystems.pdf
[30] Illustration by David Somerville, https://random-blather.com/2014/04/28/information-isnt-power/



If this can be achieved; then accounting, reporting, auditing, and analysis tasks and processes can be automated by using rules-based and patterns-based artificial intelligence. The *Seattle Method*[31] shows how this can be achieved, how to control the logical schema of a financial report.

Logical Schema of Financial Report Model and Report

The following documents the logical statements that forms the logical schema of a financial report. You can see this logical schema implemented in the cloud-based version of Luca[32]. You can see the model in the Excel import files used to import information into Luca provided by the examples.

The full set of Excel files that were used to create the screen shots in this document can be downloaded and examined to get a more complete understanding of a financial report's logical schema[33]. While I do not have an understanding of UML or OWL or any other modeling language for that matter; the Excel documentation and other information provided should help a reader understand this model and create that documentation should they have the need.

This logical schema can be implemented in many different ways. What I am showing is a straight forward version and I will be showing it in the form of Excel file screen shots. You

[31] Charles Hoffman, CPA, *Seattle Method*, http://xbrlsite.com/seattlemethod/
[32] *Cloud-based Luca*, http://xbrl.squarespace.com/journal/2021/8/31/cloud-based-luca.html
[33] Logical Schema, http://xbrlsite.com/seattlemethod/LogicalSchemaInExcel.zip

can experience the logical schema by manually entering information into the cloud-base version of Luca. See any of the provided tutorials to do that.

There are two Excel files that are provided within ZIP archives that where used to create the screen shots that are used in the following sections: Core Luca logical model[34]; supplement[35].

Base Information

Base information shows other schemas that are referenced by the schema that you are creating in Luca. (I am not happy with the name "base information", I speculate that there is a better name.) This tends to be the most technical looking information.

Logical Schema:

Code	NamespacePrefix	NamespaceIdentifier	SchemaLocation	DefaultLanguage	TaxonomyDescription
Local	report	http://www.xbrlsite.com/report	report.xsd	en	Report Model Created from Proof
Import	proof	http://accounting.auditchain.finance/proof	http://accounting.auditchain.finance/reporting-scheme/proof/base-taxonomy/proof.xsd	en	PROOF (2021-04-19)

Code: The Code has exactly two values: Local and Import. "Local" indicates that the row is for the report that is currently being created. "Import" indicates that the row is for a referenced XBRL taxonomy (base taxonomy). Every report MUST have exactly one entry for Local. Zero to many entries for Import may exist.

NamespacePrefix: The namespace prefix is any valid namespace prefix per XML.

NamespaceIdentifier: The namespace identifier is any valid namespace identifier per XML and must be consistent with the schema provided in the SchemaLocation.

SchemaLocation: The schema location is the name of the XBRL taxonomy schema file to be used for the Local code that is provided. For entries with Import it can be any XBRL taxonomy that exists on the internet.

DefaultLanguage: The default language must (a) exist within the ListLanguages lookup list, see http://luca.yaxbrl.com/administration/LanguagesList; the ISOCode field.

TaxonomyDescription: The taxonomy description is an optional description that may be added to a row in the base information.

A list of reporting schemes or base taxonomies that can be referenced by the system can be used exclusively or as an option for populating an Import row. For example, this is a list of reporting schemes provided in Luca:

http://luca.yaxbrl.com/administration/ReportingSchemesList

[34] Excel for core logical model, http://accounting.auditchain.finance/library/proof-import-excel-2022-02-23.zip

[35] Excel for supplement, http://accounting.auditchain.finance/library/proof-import-excel-2022-02-23_Supplement.zip

Terms

Terms is information about each report element that is (a) to be written into a Local report model or (b) exists within a base taxonomy schema which is imported into the local report model.

Logical schema:

Category	StandardLabel	Prefix	ReportElementName	DataType	BalanceType	PeriodType
Hypercube	Balance Sheet [Hypercube]	proof	BalanceSheetHypercube			
LineItems	Balance Sheet [Line Items]	proof	BalanceSheetLineItems			
Abstract	Balance Sheet [Arithmetic]	proof	BalanceSheetSet			
Concept	Assets	proof	Assets	Monetary	Debit	Instant
Concept	Liabilities	proof	Liabilities	Monetary	Credit	Instant
Concept	Equity	proof	Equity	Monetary	Credit	Instant
Hypercube	Comprehensive Income Statement [Hypercube]	proof	ComprehensiveIncomeStatementHypercube			
LineItems	Comprehensive Income Statement [Line Items]	proof	ComprehensiveIncomeStatementLineItems			
Abstract	Comprehensive Income [Roll Up]	proof	ComprehensiveIncomeRollUp			
Concept	Revenues	proof	Revenues	Monetary	Credit	Duration
Concept	Expenses	proof	Expenses	Monetary	Debit	Duration

Category: All valid values for category include: Hypercube, Dimension, Member, LineItems, Concept, Abstract.

StandardLabel: All terms MUST provide a standard label. The standard label can be used to autogenerate the report element name by removing spaces, special characters that are not allowed or desired within report element names.

Prefix: All terms MUST provide a prefix and the prefix MUST exist in the BaseInformation table. The prefix also MUST be a valid namespace prefix per XML. (Note that the field name is different. Should BaseInformation be changed to Prefix, or this table changed to NamespacePrefix)

ReportElementName: All terms MUST provide a report element name and that name MUST be consistent with XML Schema. (Note that this field might better be called the TermName.)

DataType: If the term is a Concept, then a data type is required and the data type MUST be from the look up list of allowed data types. See lookup lists, http://luca.yaxbrl.com/administration/DataTypesList

BalanceType: If the term is a Concept, then the balance type of Debit or Credit MAY be provided (this field is optional).

PeriodType: If the term is a Concept, then the period type is required and MUST be either Instant or Duration. (NOTE that the Forever period type is not supported.)

IMPORTANT NOTE. XBRL requires all elements defined to have data types and period types. For all Categories of report elements the data type of "String" is used and the period type of "Duration" is used to serialize XBRL. Other details related to serializing this logical information into the XBRL technical syntax is beyond the scope of this document.

Labels

Labels is a property of terms (a.k.a. report elements). Each term has a collection of labels ONE (the standard label) to MANY labels. The standard label, which is always required, is provided in the logical schema of a term.

Term	Language	LabelRole	Label
proof:Expenses	en	Negated	(Expenses)
proof:Losses	en	Negated	(Losses)
proof:Equity	en	PeriodStart	Equity, Beginning Balance
proof:Equity	en	PeriodEnd	Equity, Ending Balance
proof:DistributionsToOwners	en	Negated	(Distributions to Owners)
proof:Assets	en	Documentation	Assets are probable future economic beneits obtained or controlled by a particular entity as a result of past transactions or events.
proof:Equity	en	Documentation	Equity or net assets is the residual interest in the assets of an entity that remains after deducting its liabilities. In a business enterprise, the equity is the ownership interest.
proof:Liabilities	en	Documentation	Liabilities are probable future sacrifices of economic benefits arising from present obligations of a particular entity to transfer assets or provide services to other entities in the future as a result of past transactions or events.

Term: All labels MUST have a qualified report element name that identifies the term (a.k.a. report element) for the label with which the label is associated.

Language: All labels MUST provide a language code from the look up list of language codes.

LabelRole: All labels MUST provide a LabelRole that exists as a friendly label role in the look up list of roles that has a Category value of LabelRole.

Label: All labels MUST provide a human readable label that SHOULD be less than 255 characters (preferably shorter). Note that documentation is the exception and MAY be in excess of the 255 character limit. (NOTE, this is poorly defined, but I can explain what I am getting at and why.)

References

Labels is a property of terms (a.k.a. report elements). Each term has a collection of ZERO to MANY references.

ReportElementName	ReferenceRole	Publisher	Name	Number	Paragraph	URI	URIDate	Sequence
proof:ActualMember	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	1
proof:AllSegmentsMember	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	2
proof:Assets	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	3
proof:AssetsRollForward	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	4
proof:AssetsRollUp	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	5
proof:BalanceSheetHypercube	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	6
proof:BalanceSheetLineItems	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	7
proof:BalanceSheetSet	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	8
proof:BasisOfReportingTextBlock	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	9
proof:BudgetedMember	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	10
proof:CashFlowHypercube	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	11
proof:CashFlowLineItems	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	12
proof:ChangesInAccountingPolicy	Standard	TEST	Test	1	1	http://xbrlsite.azurewebsites.net/2021/library/reference.html	2021-02-14	13

ReportElementName: All references MUST have a qualified report element name that identifies the term (a.k.a. report element) for the reference with which the reference is associated. (NOTE, this field name is inconsistent with the Labels table and should be made consistent.)

ReferenceRole: All references MUST provide a ReferenceRole that exists as a friendly label role in the look up list of roles that has a Category of ReferenceRole.

Publisher: This is a simple standard reference part defined for the Seattle Method and MAY be provided.

Name: This is a simple standard reference part defined for the Seattle Method and MAY be provided.

Number: This is a simple standard reference part defined for the Seattle Method and MAY be provided.

Paragraph: This is a simple standard reference part defined for the Seattle Method and MAY be provided.

URI: This is a simple standard reference part defined for the Seattle Method and MUST be provided.

URIDate: This is a simple standard reference part defined for the Seattle Method and MUST be provided.

Note that this is a simplified version of using references. Generally, base taxonomies provide references by reports so not (per SEC and ESMA rules). However, we are providing this simplified version to make working and implementing reference easier. What is made easier is the fact that any reference per the XBRL specification MAY have any number of reference parts that might be standard or an extension provided by the user. This is an order of magnitude more complex to implement and use. That approach may be implemented in the future, but for not this simplified approach is adequate. (NOTE, this needs to be discussed.)

Structures

Structures is a property of a report. A report MUST have at least a minimum of ONE structure (or what is the point of the report?) and MAY have MANY structures.

NetworkIdentifier	NetworkTitle	Sequence
BalanceSheet	01-Balance Sheet	1
ComprehensiveIncome	02-Comprehensive Income	2
ChangesInEquity	03-Changes in Equity	3
PriorPeriodErrors	04-Prior Period Errors	4
VarianceAnalysis	05-Variance Analysis	5
StockPlanActivity	06-Stock Plan Activity	6
FinancialHighlights	07-Financial Highlights	7
Policies	08-Policies	8
SegmentRevenues	09-Segment Revenues	9

NetworkIdentifier: The network identifier is the unique identifier of the structure and MUST be provided. (NOTE, this should probably be renamed StructureIdentifier.)

NetworkTitle: The network title is a human readable label or title of a structure and MUST be provided. This single title MAY be separated into parts similar to how the SEC uses Number, Sortcode, and Title. This can be implemented by using user preferences or reporting scheme preferences. (NOTE, this needs to be discussed and it needs to be determined if the SEC and ESMA use the same schemes.)

Sequence: The sequence is used to provide the ability to order the structures only within a software application.

IMPOTANT NOTE, structures need some extensive discussion to implement them correctly in software. The problem with structures is that the US GAAP XBRL Taxonomy and SEC reports violates XBRL best practices in the way they implement XBRL networks and XBRL hypercubes. A critically important discussion needs to take place. It is important to be familiar with information in the document *Essentials of XBRL-based Digital Financial Reporting*[36] to have this discussion. **A mistake here can cause software development expense to go up by 200%, ease of use to go down significantly, or software to be unsellable in certain use cases**.

[36] Essentials of XBRL-based Digital Financial, http://xbrlsite.azurewebsites.net/2021/essentials/EssentialsOfXBRLBasedDigitalFinancialReporting.pdf

Associations

Associations (a.k.a. relations) is a property of a structure. A report has a structure and a structure has associations. In this implementation associations parallel XBRL presentation, calculation, and definition relations; HOWEVER:

1. It is the case that 100% of XBRL definition associations can be autogenerated from XBRL presentation associations. This is particularly true if best practices are used, but also true if they are not. (This is proven by XBRL Cloud who provides this feature.)
2. It is very likely the case that XBRL calculations can be autogenerated by XBRL presentation relations when best practices are used. But, this has NOT BEEN PROVEN.
3. IF it is the case that 100% of definition associations and 100% of calculation associations can be autogenerated from best practice presentation associations; then WHY force the user to enter this information (i.e. software should automatically provide this information).
4. Information that ends up as XBRL calculation relations could also have been provided using XBRL formulas which is more powerful and safer to use because you get more control. I am not sure, but I speculate, that software can bi-directionally convert XBRL calculation associations to XBRL formulas and the other way around.

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

StructureType: The structure type MUST be provided and would be one of: Presentation, Calculation, Definition.

NetworkIdentifier: The network identifier is the unique identifier of the structure in which the association exists and MUST be provided. (NOTE, this should probably be renamed StructureIdentifier.)

AssocationFromName: The namespace prefix qualified ReportElementName of the parent (from) of the association which MUST be provided.

AssociationRole: The association role is the HumanReadableName from the lookup list of association roles and MUST be provided.

AssocationToName: The namespace prefix qualified ReportElementName of the child (to) of the association which MUST be provided.

CalculationPolarity: The calculation polarity is used exclusively for associations which have a StructureType of Calculation; it MUST occur those associations and the value MUST be either Add or Subtract.

PreferredLabelRole: The preferred label role is used exclusively for associations which have a structure type of Presentation; it MAY occur and specify a label role from the lookup list of label roles the value of which is the HumanReadableName. A label with that label role for the AssocationToName MUST exist within the set of labels for that report element.

Sequence: The sequence is the order that the association appears in the list of associations for a specific parent and MUST be provided.

Rules

Rules (a.k.a. assertions, restrictions, constraints) is a property of a structure but also can be a property of a report. There are eight specific rule categories:

- Arithmetic
- Roll forward
- Member aggregation
- Adjustment
- Variance
- Consistency (same as Arithmetic)
- Derivation
- Nonstandard rules (everything else supported by XBRL Formula not specified above)

Effectively, the primary types of rule patterns are specifically specified and user interfaces are provided specifically for those patterns to make rule creation easier. The Nonstandard rules pattern is a "catch all" that gives this logical schema (a) everything else in terms of functionality but also (b) an easy way to handle the most common rule patterns. If new rule patterns can be determined, additional rule patterns can be specified.

Arithmetic

RuleType	RuleCode	Rule	Structure	Concept	Sequence	Commentary
ConsistencyRule	BS01	$Assets = ($Liabilities + $Equity)	BalanceSheet	proof:Assets	1	
ConsistencyRule	IS01_CI	$ComprehensiveIncome = ($Revenues - $Expenses + $Gains - $Losses)	ComprehensiveIncome	proof:ComprehensiveIncome	2	

Roll forward

RuleType	RuleCode	Rule	Structure	Concept	Sequence	Commentary
RollForwardRule	SHE01	$Equity_BalanceStart + $ComprehensiveIncome + $InvestmentsByOwners - $DistributionsToOwners = $Equity_BalanceEnd	ChangesInEquity	proof:Equity	1	

Member aggregation

RuleType	RuleCode	Rule	Structure	Concept	Dimension	Sequence	Commentary
MemberAggregationRule	MA1	$Total eq sum($Each)	SegmentRevenues	proof:Revenues	proof:SegmentsAxis	1	

Adjustment

RuleType	RuleCode	Rule	Structure	Concept	Dimension	MemberPrior	MemberCurrent	Sequence	Commentary
AdjustmentRule	ADJ01	$Restated = ($OriginallyStated + $CorrectionOfAnError + $ChangesInAccountingPolicy)	PriorPeriodErrors	proof:Equity	proof:ReportDateAxis	proof:PriorReportMember	proof:CurrentReportMember	1	

Variance

RuleType	RuleCode	Rule	Structure	Concept	Dimension	MemberActual	MemberBudget	MemberVariance	Sequence	Commentary
VarianceRule	VAR01_Revenues	$Actual = ($Budget + $Variance)	VarianceAnalysis	proof:Revenues	proof:ScenarioAxis	proof:ActualMember	proof:BudgetedMember	proof:VarianceMember	1	
VarianceRule	VAR02_Expenses	$Actual = ($Budget + $Variance)	VarianceAnalysis	proof:Expenses	proof:ScenarioAxis	proof:ActualMember	proof:BudgetedMember	proof:VarianceMember	2	
VarianceRule	VAR03_Gains	$Actual = ($Budget + $Variance)	VarianceAnalysis	proof:Gains	proof:ScenarioAxis	proof:ActualMember	proof:BudgetedMember	proof:VarianceMember	3	
VarianceRule	VAR04_Losses	$Actual = ($Budget + $Variance)	VarianceAnalysis	proof:Losses	proof:ScenarioAxis	proof:ActualMember	proof:BudgetedMember	proof:VarianceMember	4	
VarianceRule	VAR05_ComprehensiveIncome	$Actual = ($Budget + $Variance)	VarianceAnalysis	proof:ComprehensiveIncome	proof:ScenarioAxis	proof:ActualMember	proof:BudgetedMember	proof:VarianceMember	5	

Consistency

RuleType	RuleCode	Rule	Structure	Concept	Sequence	Commentary
ConsistencyRule	BS01	$Assets = ($Liabilities + $Equity)	BalanceSheet	proof:Assets	1	
ConsistencyRule	IS01_CI	$ComprehensiveIncome = ($Revenues - $Expenses + $Gains - $Losses)	ComprehensiveIncome	proof:ComprehensiveIncome	2	

Derivation

RuleType	RuleCode	Rule	Precondition	DerivedFact	BasedOnSourceFact	Commentary	Sequence
DerivationRule	BS-Impute-01	$CurrentAssets	exists($LiabilitiesAndEquity) and exists($CurrentAssets) and ($Assets eq null) and ($NoncurrentAssets eq null)	fac:Assets	fac:LiabilitiesAndEquity		1
DerivationRule	BS-Impute-02	$CurrentAssets	exists($LiabilitiesAndEquity) and exists($CurrentAssets) and ($Assets eq null)	fac:Assets	fac:LiabilitiesAndEquity		2
DerivationRule	BS-Impute-03	$CurrentAssets	($Assets eq null) and ($NoncurrentAssets eq null) and exists($Liabilities) and exists($Equity) and exists($LiabilitiesAndEquity) and ($LiabilitiesAndEquity = $Liabilities + $Equity)	fac:Assets	fac:LiabilitiesAndEquity	This is incorrect. The concept fac:CurrentAssets needs to exist within the precondition. Needs to be added.	3
DerivationRule	BS-Impute-04	$Assets - $CurrentAssets	($NoncurrentAssets eq null) and exists($CurrentAssets) and exists($Assets)	fac:NoncurrentAssets	fac:Assets		4
DerivationRule	BS-Impute-05	$Assets	($LiabilitiesAndEquity eq null) and exists($Assets)	fac:LiabilitiesAndEquity	fac:Assets		5
DerivationRule	BS-Impute-06	$EquityAttributableToParent + $EquityAttributableToNoncontrollingInterest	($Equity eq null) and exists($EquityAttributableToNoncontrollingInterest) and exists($EquityAttributableToParent)	fac:Equity	fac:EquityAttributableToParent		6

Nonstandard

RuleType	RuleCode	Rule	Sequence	Commentary
NonstandardRule	NA01	<generic:link xlink:type='extended' xlink:role='http://www.xbrl.org/2003/role/link'> <va:valueAssertion xlink:type='resource' xlink:label='ASSERTION' id='NA01' aspectModel='dimensional' implicitFiltering='true' test='$NetAssets = ($Assets - $Liabilities)' /> <generic:arc xlink:type='arc' xlink:arcrole='http://xbrl.org/arcrole/2008/element-label' xlink:from='ASSERTION' xlink:to='ASSERTION_Label' order='1.0'/> <label:label xlink:type='resource' xlink:label='ASSERTION_Label' xlink:role='http://www.xbrl.org/2008/role/label' xml:lang='en'> $NetAssets = ($Assets - $Liabilities)	1	
NonstandardRule	[illegible]	[illegible]	2	

(NOTE, a question exists in my mind as to how to store the rules in some sort of database but I really don't have the background to have an informed opinion as to the best approach to use. I do understand the functionality that is necessary.)

Facts

Facts are a property of a report. A fact has a property (a collection) which represents the aspects (a.k.a. dimensions) of that fact. A fact has a property (a collection) which represents the parenthetical exploitations related to that fact.

ReportingEntityAspect	CalendarPeriodAspect	ConceptAspect	FactValue	Units	Rounding	FactID	Sequence
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-12-31	proof:Assets	3500	iso4217:USD	INF	FACT01	1
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-12-31	proof:Liabilities	0	iso4217:USD	INF		2
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-12-31	proof:Equity	3500	iso4217:USD	INF		3
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Revenues	7000	iso4217:USD	INF		4
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Gains	1000	iso4217:USD	INF		5
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Expenses	3000	iso4217:USD	INF		6
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Losses	2000	iso4217:USD	INF		7
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:ComprehensiveIncome	3000	iso4217:USD	INF		8
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2019-12-31	proof:Assets	0	iso4217:USD	INF		9
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2019-12-31	proof:Liabilities	0	iso4217:USD	INF		10
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2019-12-31	proof:Equity	0	iso4217:USD	INF		11
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:InvestmentsByOwners	1000	iso4217:USD	INF		12
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:DistributionsToOwners	500	iso4217:USD	INF		13
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Revenues	6000	iso4217:USD	INF	FACT14	14
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Gains	750	iso4217:USD	INF	FACT15	15
GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442	2020-01-01 | 2020-12-31	proof:Expenses	2000	iso4217:USD	INF	FACT16	16

ReportingEntityAspect: The reporting entity aspect MUST exist and is an entity identifier (a token) plus a scheme of entity identifiers used (a token) separated by " | " which is a space, a pipe character, and another space.

CalendarPeriodAspect: The calendar period aspect MUST exist and is an instant (as of date) or a duration (for period). An instant is a token, a duration is two tokens (start data and end date) separated by " | " (space, pipe character, space).

ConceptAspect: The concept is a qualified report element name from the category of Concept that exists within the list of Terms.

FactValue: The fact value is a numeric or nonnumeric value that is described by the other information related to the fact.

Units: The units is a human readable friendly identifier used on numeric facts that comes from the FriendlyUnits field from the units lookup list.

Rounding: The rounding is the value that will be used for the decimals attribute of an XBRL fact when an XBRL instance is serialized.

FactID: The fact id is a unique identifier for a fact that MAY be provided. A fact id MUST be provided when entries in the fact dimensions table or the fact parenthetical explanations table. The fact ID MUST follow the syntax rules for an XML ID. (NOTE that fact IDs could be autogenerated for every fact, this approach might be preferable.)

Fact Dimensions

Fact dimensions (a.k.a. aspects) is a property of a fact. It is a collection of the set of noncore dimensions (aspects) that are associated with a fact.

FactID	Dimension	Member	Sequence
FACT14	proof:ScenarioAxis	proof:BudgetedMember	1
FACT15	proof:ScenarioAxis	proof:BudgetedMember	2
FACT16	proof:ScenarioAxis	proof:BudgetedMember	3
FACT17	proof:ScenarioAxis	proof:BudgetedMember	4
FACT18	proof:ScenarioAxis	proof:BudgetedMember	5
FACT19	proof:ScenarioAxis	proof:VarianceMember	6
FACT20	proof:ScenarioAxis	proof:VarianceMember	7
FACT21	proof:ScenarioAxis	proof:VarianceMember	8
FACT22	proof:ScenarioAxis	proof:VarianceMember	9
FACT23	proof:ScenarioAxis	proof:VarianceMember	10
FACT29	proof:SegmentsAxis	proof:SegmentAlphaMember	11
FACT30	proof:SegmentsAxis	proof:SegmentBravoMember	12
FACT31	proof:SegmentsAxis	proof:SegmentCharlieMember	13
FACT32	proof:ReportDateAxis	proof:PriorReportMember	14

FactID: The fact id is a key that associations each dimension/aspect to a specific fact and MUST exist.

Dimension: The dimension is a qualified report element name of the Category Dimension that exists in the list of Terms for the report model which is used to associate a noncore dimension to a fact and MUST be provided for every row.

Member: The member is a qualified report element name of the Category Member that exists in a list of Terms for the report model which is used to association a member to a dimension and MUST be provided. The member MUST exist within a set of members for the dimension specified for a row of the dimensions table.

Sequence: The sequence is an optional field that can be use to sort dimensions within a software application.

Fact Parenthetical Explanations

The fact parenthetical explanations (a.k.a. XBRL footnotes) is a property of a fact. It is a collection of the parenthetical explanations that relate to a specific fact.

FactID	FootnoteID	Parenthetical	Sequence
FACT01	FOOTNOTE01	This is footnote #1 that goes with fact #1	1
FACT01	FOOTNOTE02	This is footnote #2 that goes with fact #1	2
FACT01	FOOTNOTE03	This is footnote #3 that goes with fact #1	3

FactID: The fact id is a key that associations each parenthetical explanation to a specific fact and MUST exist.

FootnodeID: The footnote ID is a key that associates the fact id with a specific footnote id and MUST be provided.

Parenthetical: The parenthetical is the actual value of the parenthetical explanation and MUST be provided.

Sequence: The sequence is an optional field that can be used to sort parenthetical explanations within a software application.

Logical Schema of Supporting Accounting and Other Knowledge

The following is the logical schema information for supporting information related to other knowledge used to work with financial reports. This knowledge is generally associated with a financial reporting scheme[37].

Note that each and every category of knowledge below fits into one or more of the database tables for a report model and report: Terms, Labels, References, Structures, Associations, or Rules.

Topics

Topics[38] is a mechanism that is used to sort and filter disclosures. Topics use the patterns of Terms, Labels, References, Associations, and Structures.

Disclosures

Disclosures[39] is a mechanism that associates a Block of information with a specific financial or nonfinancial disclosure within a financial report. Disclosures use the patterns of Terms, Labels, References, Associations, and Structures.

Model Structure Rules

Model structure[40] is a mechanism to be sure XBRL presentation relations are represented logically. The following table specifies the relations allowed within an Associations table:

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

Model structure rules follow the pattern of associations but would never use CalculationPolarity or PreferredLabelRole.

[37] PROOF financial reporting scheme prototype, http://www.xbrlsite.com/seattlemethod/proof/documentation/index.html

[38] PROOF topics, http://www.xbrlsite.com/seattlemethod/proof/documentation/Topics.html

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

[40] PROOF model structure, http://www.xbrlsite.com/seattlemethod/proof/model-structure/ModelStructure.html

StructureType	NetworkIdentifier	AssociationFromName	AssociationRole	AssociationToName	CalculationPolarity	PreferredLabelRole	Sequence
Definition	http://www.xbrl.org/2003/role/link	cm:Network	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/parentCategory-allowedChildCategory	cm:Abstract			1
Definition	http://www.xbrl.org/2003/role/link	cm:Thing	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	cm:Network			2
Definition	http://www.xbrl.org/2003/role/link	cm:Network	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/parentCategory-allowedChildCategory	cm:Table			3
Definition	http://www.xbrl.org/2003/role/link	cm:Thing	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	cm:Table			4
Definition	http://www.xbrl.org/2003/role/link	cm:Network	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/parentCategory-disallowedChildCategory	cm:Network			5
Definition	http://www.xbrl.org/2003/role/link	cm:Thing	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	cm:Axis			6
Definition	http://www.xbrl.org/2003/role/link	cm:Network	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/parentCategory-disallowedChildCategory	cm:Axis			7
Definition	http://www.xbrl.org/2003/role/link	cm:Thing	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	cm:Member			8
Definition	http://www.xbrl.org/2003/role/link	cm:Thing	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/class-subClass	cm:LineItems			9
Definition	http://www.xbrl.org/2003/role/link	cm:Network	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/parentCategory-disallowedChildCategory	cm:LineItems			10

See Framework documentation[41] for a complete list of model structure association logic.

Fundamental Accounting Concepts and Relations

Fundamental accounting concepts and relations are consistency and continuity cross checks to be sure that reported facts used within a report are consistent and do not contradict other reported facts.

Reporting Styles

Reporting styles[42] is a scheme that is used to organize financial reports into specific patterns for the purpose of assigning a set of consistency rules, derivation rules, and mapping rules to that report. (NOTE, a physical reporting style is not necessarily required; all that is needed is to know which parts to use for a specific report.)

Consistency Rules

Consistency rules[43] are arithmetic rules that specify the relations between fundamental accounting concepts. Follows the logic pattern of consistency rules.

Derivation Rules

Derivation rules[44] are rules that are used to derive high-level financial information that is not explicitly reported within a financial report using consistency rules and other information. Follows the logic pattern of derivation rules.

Mapping Rules

Mapping rules[45] are rules that are used to associate a report fact concept to a high-level fundamental accounting concept. Mapping rules are necessary because reporting entities could use many different report elements to report the same high-level financial concepts. Mapping rules follow the logic pattern of associations.

[41] Framework, Model Structure, http://xbrlsite.com/seattlemethod/Framework.pdf
[42] PROOF reporting styles, http://www.xbrlsite.com/seattlemethod/proof/documentation/ReportingStyles.html
[43] PROOF consistency rules, http://www.xbrlsite.com/seattlemethod/proof/documentation/Consistency.html
[44] PROOF derivation rules, http://www.xbrlsite.com/seattlemethod/proof/documentation/Derivation.html
[45] PROOF mapping rules, http://www.xbrlsite.com/seattlemethod/proof/fac/Rules_Mapping/ConceptMap_General-mapping.html

Fundamental Accounting Concepts

Fundamental accounting concepts[46] are high-level financial concepts that have consistent patterns across multiple financial reports per some reporting style pattern. Fundamental accounting concepts follow the logic patterns of Terms, Labels, References, Associations, Rules, Structures.

Disclosure Mechanics

Disclosure mechanics rules[47] is a mechanism for helping software users find report elements in an XBRL taxonomy, construct report fragments to report specific disclosure information, verify that the created report fragments or information Block have been correctly constructed, and for extracting information from reports for analysis.

Disclosure mechanics rules follow the logical pattern of Associations.

StructureType	NetworkIdentifier	AssociationFromName	AssociationRole	AssociationToName	CalculationPolarity	PreferredLabelRole	Sequence
Definition	BalanceSheet	cm:Thing	cm-arcroles:class-equivalentClass	disclosures:BalanceSheet			0
Definition	BalanceSheet	disclosures:BalanceSheet	drules-arcroles:disclosure-requiresHypercube	proof:BalanceSheetHypercube			1
Definition	BalanceSheet	disclosures:BalanceSheet	drules-arcroles:disclosure-hasConceptArrangementPattern	cm:Component			2
Definition	BalanceSheet	disclosures:BalanceSheet	drules-arcroles:reportedDisclosure-requiresDisclosure	disclosures:AssetsRollUp			3
Definition	BalanceSheet	disclosures:BalanceSheet	drules-arcroles:reportedDisclosure-requiresDisclosure	disclosures:LiabilitiesAndEquityRollUp			4
Definition	AssetsRollUp	cm:Thing	cm-arcroles:class-equivalentClass	disclosures:AssetsRollUp			0
Definition	AssetsRollUp	disclosures:AssetsRollUp	drules-arcroles:disclosure-requiresHypercube	proof:BalanceSheetHypercube			1
Definition	AssetsRollUp	disclosures:AssetsRollUp	drules-arcroles:disclosure-hasConceptArrangementPattern	cm:RollUp			2

See the Framework documentation[48] for a complete set of association types for disclosure mechanics rules.

Reporting Checklist

Reporting checklist rules[49] is a mechanism for managing a financial report creation project, specifying what disclosures are required for different reporting styles, helping a report creation tool user find disclosures and related report element information in XBRL taxonomies, and such. Reporting checklist follows logic pattern of associations.

StructureType	NetworkIdentifier	AssociationFromName	AssociationRole	AssociationToName	CalculationPolarity	PreferredLabelRole	Sequence
Definition	ReportingChecklist	cm:Thing	cm-arcroles:class-equivalentClass	cm:FinancialReport			1
Definition	ReportingChecklist	cm:FinancialReport	drules-arcroles:financialReport-requiresDisclosure	disclosures:BalanceSheet			2
Definition	ReportingChecklist	cm:FinancialReport	drules-arcroles:financialReport-requiresDisclosure	disclosures:AssetsRollUp			3
Definition	ReportingChecklist	cm:FinancialReport	drules-arcroles:financialReport-requiresDisclosure	disclosures:LiabilitiesAndEquityRollUp			4
Definition	ReportingChecklist	cm:FinancialReport	drules-arcroles:financialReport-requiresDisclosure	disclosures:AssetsRollForward			5
Definition	ReportingChecklist	cm:FinancialReport	drules-arcroles:financialReport-requiresDisclosure	disclosures:BasisOfReporting			6
Definition	ReportingChecklist	cm:FinancialReport	drules-arcroles:financialReport-requiresDisclosure	disclosures:CashFlowStatement			7

See the Framework documentation[50] for a complete set of association types for reporting checklist rules.

Templates

Templates[51] are example disclosures that can be leveraged to create disclosures within a financial report. Templates are documented in machine readable form using XBRL references linkbase[52].

---

[46] PROOF fundamental accounting concepts, http://www.xbrlsite.com/seattlemethod/proof/documentation/FundamentalAccountingConcepts.html

[47] PROOF disclosure mechanics rules, http://www.xbrlsite.com/seattlemethod/proof/disclosure-mechanics/DisclosureMechanicsRulesInNaturalLanguage.html

[48] Framework, Disclosure Mechanics, page 9, http://xbrlsite.com/seattlemethod/Framework.pdf

[49] PROOF reporting checklist rules, http://www.xbrlsite.com/seattlemethod/proof/reporting-checklist/reporting-checklist-rules.html

[50] Framework, Reporting Checklist, page 10, http://xbrlsite.com/seattlemethod/Framework.pdf

Exemplars

Exemplars[53] are similar to templates in that they can be leveraged to create disclosures within a financial report; but they are different in that an exemplar is an example from some other existing financial report that provides that same or similar disclosure. Exemplars are documented using an XBRL references linkbase[54].

Type-subtype Rules

Type-subtype rules[55] (a.k.a. wider-narrower rules, a.k.a. general-special rules) are used to specify allowable associations between report elements within a report model. Type-subtype rules are expressed using associations type table.

See the Framework documentation[56] for the complete set of type-subtype association logic.

Properties

Properties[57] are useful characteristics that can be used to sort, filter, categorize, and otherwise make use of other information. Properties are expressed using associations type table.

See the Framework documentation[58] for the complete set of type-subtype association logic.

Lookup Lists and Other Supporting Logical Schemas

The following is a set of what amounts to lookup lists that are used within a financial report model or financial report.

List Reporting Schemes

The following is a lost of financial reporting schemes supported by the system:

NamespacePrefix	NamespaceIdentifier	SchemaLocation	ModelStructureLocation	Description	Source	Sequence
ae	http://www.xbrlsite.com/ae	http://xbrlsite.azurewebsites.net/2020/introduction/ae-basic/ae.xsd	http://xbrlsite.azurewebsites.net/2020/introduction/ae-basic/ae_ModelStructure.xml	Accounting Equation, very basic (Prototype) (test of 02.10.2021)	XBRL Site	1
ae	http://www.xbrlsite.com/ae	http://xbrlsite.azurewebsites.net/2020/introduction/ae-na/ae.xsd	http://xbrlsite.azurewebsites.net/2020/introduction/ae-na/ae_ModelStructure.xml	Accounting Equation, with Net Assets Approach (Prototype)	XBRL Site	2
sfac6	http://www.xbrlsite.com/sfac6	http://xbrlsite.azurewebsites.net/2020/intermediate/sfac6/sfac6.xsd	http://xbrlsite.azurewebsites.net/2020/intermediate/sfac6/sfac6_ModelStructure.xml	SFAC 6 Elements of Financial Statements, Basic	XBRL Site	3
sfac6	http://www.xbrlsite.com/sfac6	http://xbrlsite.azurewebsites.net/2020/introduction/sfac6-dim/sfac6.xsd	http://xbrlsite.azurewebsites.net/2020/introduction/sfac6-dim/sfac6_ModelStructure.xml	SFAC 6 Elements of Financial Statements, with Hypercubes (Prototype)	XBRL Site	4
sfac6	http://www.xbrlsite.com/sfac6	http://xbrlsite.azurewebsites.net/2020/introduction/sfac6-dim2/sfac6.xsd	http://xbrlsite.azurewebsites.net/2020/introduction/sfac6-dim2/sfac6_ModelStructure.xml	SFAC 6 Elements of Financial Statements, alternative hypercube approach	XBRL Site	5
common	http://www.xbrlsite.com/common	http://xbrlsite.azurewebsites.net/2020/intermediate/common/common.xsd	http://xbrlsite.azurewebsites.net/2020/intermediate/common/common_ModelStructure.xml	Common Elements of Financial Report (Prototype)	XBRL Site	6
mini	http://www.xbrlsite.com/mini	http://xbrlsite.azurewebsites.net/2020/Prototype/iteration2/mini.xsd	http://xbrlsite.azurewebsites.net/2020/Prototype/iteration2/mini_ModelStructure.xml	MINI Financial Reporting Scheme, Basic (no hypercubes)	XBRL Site	7
mini	http://www.xbrlsite.com/mini	http://xbrlsite.azurewebsites.net/2019/Prototype/mini/base-taxonomy/mini.xsd	http://xbrlsite.azurewebsites.net/2019/Prototype/mini/base-taxonomy/mini_ModelStructure.xml	MINI Financial Reporting Scheme, with disclosures (Demonstration)	XBRL Site	8
proof	http://www.xbrlsite.com/proof	http://xbrlsite.azurewebsites.net/2020/intermediate/proof/proof.xsd	http://xbrlsite.azurewebsites.net/2020/intermediate/proof/proof_ModelStructure.xml	Proof Financial Reporting Scheme (Prototype BASE)	XBRL Site	9
gaap	http://www.xbrlsite.com/gaap	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/xasb/base-taxonomy/gaap.xsd	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/xasb/base-taxonomy/gaap_ModelStructure.xml	XASB Financial Reporting Scheme (Prototype)	XBRL Site	10
nfp	http://www.xbrlsite.com/nfp	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/nfp/base-taxonomy/nfp.xsd	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/nfp/base-taxonomy/nfp_ModelStructure.xml	Not for Profit (US GAAP) Financial Reporting Scheme	XBRL Site	11
ipsas	http://www.xbrlsite.com/ipsas	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ipsas/base-taxonomy/ipsas.xsd	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ipsas/base-taxonomy/ipsas_ModelStructure.xml	IPSAS Financial Reporting Scheme (Prototype)	XBRL Site	12
ifrs-full	http://xbrl.ifrs.org/taxonomy/2019-03-27/ifrs-full	http://xbrl.ifrs.org/taxonomy/2019-03-27/full_ifrs/full_ifrs-cor_2019-03-27.xsd	http://xbrlsite.azurewebsites.net/2020/reporting-scheme/ifrs/base-taxonomy/viewer/ifrs-full_ModelStructure.xml	IFRS XBRL Taxonomy 2019	XBRL Site	13
us-gaap	http://fasb.org/us-gaap/2019-01-31	http://xbrl.fasb.org/us-gaap/2019/elts/us-gaap-2019-01-31.xsd	http://www.xbrlsite.com/2019/US-GAAP-2019-AsReleased/us-gaap-stm-sfp-cls-pre-2019-01-31_ModelStructure.xml	US GAAP XBRL Taxonomy 2019 (only balance sheet network)	XBRL Site	14
proof	http://accounting.auditchain.finance/proof	http://accounting.auditchain.finance/reporting-scheme/proof/base-taxonomy/proof.xsd	http://accounting.auditchain.finance/reporting-scheme/proof/base-taxonomy/proof_ModelStructure.xml	PROOF BASELINE (Auditchain)	Auditchain	90
mini	http://accounting.auditchain.finance/mini	http://accounting.auditchain.finance/reporting-scheme/mini/base-taxonomy/mini.xsd	http://accounting.auditchain.finance/reporting-scheme/mini/base-taxonomy/mini_ModelStructure.xml	MINI (Auditchain)	Auditchain	91
xasb	http://accounting.auditchain.finance/xasb	http://accounting.auditchain.finance/reporting-scheme/xasb/base-taxonomy/xasb.xsd	http://accounting.auditchain.finance/reporting-scheme/xasb/base-taxonomy/xasb_ModelStructure.xml	XASB (Auditchain)	Auditchain	93
pfs	http://accounting.auditchain.finance/pfs	http://accounting.auditchain.finance/reporting-scheme/pfs/base-taxonomy/pfs.xsd	http://accounting.auditchain.finance/reporting-scheme/pfs/base-taxonomy/pfs_ModelStructure.xml	PFS (Auditchain)	Auditchain	94
nfp	http://accounting.auditchain.finance/nfp	http://accounting.auditchain.finance/reporting-scheme/nfp/base-taxonomy/nfp.xsd	http://accounting.auditchain.finance/reporting-scheme/nfp/base-taxonomy/nfp_ModelStructure.xml	Not-for-Profit (Auditchain)	Auditchain	95
frf-sme	http://accounting.auditchain.finance/frf-sme	http://accounting.auditchain.finance/reporting-scheme/frf-sme/base-taxonomy/frf-sme.xsd	http://accounting.auditchain.finance/reporting-scheme/frf-sme/base-taxonomy/frf-sme_ModelStructure.xml	FRF for SMEs (Auditchain)	Auditchain	96
ipsas	http://accounting.auditchain.finance/ipsas	http://accounting.auditchain.finance/reporting-scheme/ipsas/base-taxonomy/ipsas.xsd	http://accounting.auditchain.finance/reporting-scheme/ipsas/base-taxonomy/ipsas_ModelStructure.xml	IPSAS (Auditchain)	Auditchain	97
ifrs-full	http://xbrl.ifrs.org/taxonomy/2021-03-24/ifrs-full	http://xbrl.ifrs.org/taxonomy/2021-03-24/full_ifrs/full_ifrs-cor_2021-03-24.xsd	http://accounting.auditchain.finance/reporting-scheme/ifrs-full/base-taxonomy/ifrs-full_ModelStructure.xml	IFRS FULL 2021 (Auditchain)	Auditchain	98
us-gaap	http://fasb.org/us-gaap/2021-01-31	http://xbrl.fasb.org/us-gaap/2021/elts/us-gaap-2021-01-31.xsd	http://accounting.auditchain.finance/reporting-scheme/us-gaap/base-taxonomy/us-gaap_ModelStructure.xml	US GAAP 2021 (Auditchain)	Auditchain	99
fac	http://www.xbrlsite.com/fac	http://accounting.auditchain.finance/fac/fac.xsd	http://accounting.auditchain.finance/fac/fac_ModelStructure.xml	FAC Auditchain	Auditchain	99

NamespacePrefix: The namespace prefix of the financial reporting scheme.

---

[51] PROOF templates, http://www.xbrlsite.com/seattlemethod/proof/documentation/Templates.html

[52] PROOF (Auditchain) templates, machine readable, http://accounting.auditchain.finance/reporting-scheme/proof/templates/disclosures-templates-ref.xml

[53] PROOF exemplars, http://www.xbrlsite.com/seattlemethod/proof/documentation/Exemplars.html

[54] PROOF (Auditchain) exemplars, machine readable, http://accounting.auditchain.finance/reporting-scheme/proof/exemplars/disclosures-exemplarsList-ref.xml

[55] PROOF type-subtype rules, http://www.xbrlsite.com/seattlemethod/proof/documentation/TypeSubtype.html

[56] Framework, Type-subtype associations, page 11, http://xbrlsite.com/seattlemethod/Framework.pdf

[57] PROOF mappings, http://www.xbrlsite.com/seattlemethod/proof/documentation/Properties.html

[58] Framework, Properties, page 11, http://xbrlsite.com/seattlemethod/Framework.pdf

NamespaceIdentifier: The namespace identifier of the financial reporting scheme.

SchemaLocation: The schema location of the XBRL taxonomy for the financial reporting scheme.

ModelStructureLocation: This is the XML infoset generated from the XBRL taxonomy entry point that provides all presentation, calculation, and definition relations. (NOTE, this should be replaced by the entry point XBRL taxonomy URL; the only reason this is used is to show that a pre-processed version can be leveraged.)

Description: This is a required human readable label that MUST be provided for the financial reporting scheme.

Source: This is the human readable source of the financial reporting scheme and MUST be provided.

Sequence: This is the sequence of the financial reporting scheme which MUST be provided and is used as one mechanism to sort the list of financial reporting schemes.

List Data Types

This is a list of data types that are used by the system. Other data types are provided by XBRL International Data Types Registry[59] and XML Schema Part 2: Datatypes[60]. For additional information please see the Seattle Method Framework[61].

[59] XBRL International, *Data Types Registry*, http://www.xbrl.org/dtr/dtr.xml
[60] W3C*, XML Schema Part 2: Datatypes*, https://www.w3.org/TR/xmlschema-2/
[61] Seattle Method, *Framework*, http://xbrlsite.com/seattlemethod/Framework.pdf

FriendlyDataType	TechnicalDataType	Category	GeneralUse	OWLDataType
Monetary	xbrli:monetaryItemType	Numeric	Common	xsd:decimal
String	xbrli:stringItemType	Nonnumeric	Common	xsd:string
TextBlock	nonnum:textBlockItemType	Nonnumeric	Common	xsd:string
Shares	xbrli:sharesItemType	Numeric	Common	xsd:decimal
Percent	num:percentItemType	Numeric	Common	xsd:decimal
Date	xbrli:dateItemType	Nonnumeric	Common	xsd:dateTime
Decimal	xbrli:decimalItemType	Numeric	Common	xsd:decimal
Float	xbrli:floatItemType	Numeric	Uncommon	xsd:float
Double	xbrli:doubleItemType	Numeric	Uncommon	xsd:double
Integer	xbrli:integerItemType	Numeric	Common	xsd:integer
Integer-Nonpositive	xbrli:nonPositiveIntegerItemType	Numeric	Uncommon	xsd:nonPositiveInteger
Integer-Negative	xbrli:negativeIntegerItemType	Numeric	Uncommon	xsd:negativeInteger
Long	xbrli:longItemType	Numeric	Uncommon	xsd:long
Int	xbrli:intItemType	Numeric	Uncommon	xsd:int
Short	xbrli:shortItemType	Numeric	Uncommon	xsd:short
Byte	xbrli:byteItemType	Numeric	Uncommon	xsd:byte
Integer-Nonnegative	xbrli:nonNegativeIntegerItemType	Numeric	Uncommon	xsd:nonNegativeInteger
Pure	xbrli:pureItemType	Numeric	Common	xsd:decimal
Boolean	xbrli:booleanItemType	Nonnumeric	Common	xsd:boolean
String-Token	xbrli:tokenItemType	Nonnumeric	Common	xsd:token
String-Normalized	xbrli:normalizedStringItemType	Nonnumeric	Common	xsd:normalizedString
Time	xbrli:timeItemType	Nonnumeric	Uncommon	xsd:dateTime
Base64	xbrli:base64BinaryItemType	Nonnumeric	Uncommon	xsd:base64Binary
URI	xbrli:anyURIItemType	Nonnumeric	Common	xsd:anyURI
Date+Time	xbrli:dateTimeItemType	Nonnumeric	Uncommon	xsd:dateTime
Integer-Positive	xbrli:positiveIntegerItemType	Numeric	Uncommon	xsd:positiveInteger
Hex	xbrli:hexBinaryItemType	Nonnumeric	Uncommon	xsd:hexBinary
Qualified Name	xbrli:QNameItemType	Nonnumeric	Uncommon	xsd:anyURI
Per-Share	num:perShareItemType	Numeric	Common	xsd:decimal

FriendlyDataType: This is a nontechnical looking human readable friendly data type name that can be presented to the user of software applications. This field MUST be provided and MUST be a token.

TechnicalDataType: This is the official XBRL technical data type name which MUST be provided and is used to serialize XBRL.

Category: This is a code that distinguishes data types as Numeric or Nonnumeric which MUST be provided and is used when serializing information to the XBRL technical syntax. This determines if a fact does or does not have a units attribute.

GeneralUse: This is a code that distinguishes commonly used data types from uncommonly used data types and can be used by software applications to filter and sort data types in ways useful to human users of software applications.

OWLDataType: This is the XML Schema data type that is used by OWL or possibly other software applications and is used to convert XBRL data types to RDF/OWL/SHACL format.

List Languages

The language list is a list of ISO 639 language code[62] names and other information that is used for identifying languages.

Key	ISOLanguageName	ISOCode	LanguageFamily	Enabled
1	Abkhazian	ab	Northwest Caucasian	FALSE
2	Afar	aa	Afro-Asiatic	FALSE
3	Afrikaans	af	Indo-European	FALSE
4	Akan	ak	Niger–Congo	FALSE
5	Albanian	sq	Indo-European	FALSE
6	Amharic	am	Afro-Asiatic	FALSE
7	Arabic	ar	Afro-Asiatic	FALSE
8	Aragonese	an	Indo-European	FALSE
9	Armenian	hy	Indo-European	FALSE
10	Assamese	as	Indo-European	FALSE
11	Avaric	av	Northeast Caucasian	FALSE
12	Avestan	ae	Indo-European	FALSE
13	Aymara	ay	Aymaran	FALSE

Key: The key is really more of a sequence which can be used to sort language codes in software applications. (NOTE that perhaps this should be renamed Sequence or perhaps even dropped.)

ISOLanguageName: This is the ISO language name which MUST be provided.

ISOCode: this is the ISO code which is used within an XBRL or other standard syntax. Either a two letter scheme such as “en” or the four letter scheme such as “en-UK” and “en-US” is MAY be used. For more information see the xml:lang attribute.

LanguageFamily: The language family provides a mechanism to sort and filter language codes and MUST be provided.

Enables: The enabled field is a Boolean field that MUST be provided and is used to sort and filter language codes within software applications.

[62] Wikipedia, List of ISO 639-1 codes, https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

List Roles

The list roles table is a list of XBRL label[63] and reference[64] roles that are mapped to human readable information. Label and reference roles are used to sort and filter labels and references. For additional roles, please see the XBRL International Link Role Registry[65].

RoleName	HumanReadableName	Category	RoleURI	SchemaLocation
standard	Standard	LabelRole	http://www.xbrl.org/2003/role/label	
periodStartLabel	PeriodStart	LabelRole	http://www.xbrl.org/2003/role/periodStartLabel	
periodEndLabel	PeriodEnd	LabelRole	http://www.xbrl.org/2003/role/periodEndLabel	
documentation	Documentation	LabelRole	http://www.xbrl.org/2003/role/documentation	
negatedLabel	Negated	LabelRole	http://www.xbrl.org/2009/role/negatedLabel	http://www.xbrl.org/lrr/role/negated-2009-12-16.xsd
originallyStatedLabel	OriginallyStated	LabelRole	http://accounting.auditchain.finance/conceptual-model/cm-roles/roles/originallyStatedLabel	http://accounting.auditchain.finance/cm/cm-roles.xsd
restatedLabel	Restated	LabelRole	http://www.xbrl.org/2006/role/restatedLabel	http://www.xbrl.org/lrr/role/restated-2006-02-21.xsd
standardReference	Standard	ReferenceRole	http://www.xbrl.org/2003/role/reference	
exampleGuidance	ExampleGuidance	LabelRole	http://www.xbrl.org/2003/role/exampleGuidance	
commentaryGuidance	CommentaryGuidance	LabelRole	http://www.xbrl.org/2003/role/commentaryGuidance	
definitionGuidance	DefinitionGuidance	LabelRole	http://www.xbrl.org/2003/role/definitionGuidance	
verboseLabel	VerboseLabel	LabelRole	http://www.xbrl.org/2003/role/verboseLabel	
terseLabel	TerseLabel	LabelRole	http://www.xbrl.org/2003/role/terseLabel	
totalLabel	TotalLabel	LabelRole	http://www.xbrl.org/2003/role/totalLabel	

RoleName: The role name is the ID of the role that is defined and MUST be provided.

HumanReadableName: The human readable name is a friendly identifier that can be used in software than helps business users work with label and reference roles.

Category: The category distinguishes label roles and reference roles and MUST be provided and should be either LabelRole or ReferenceRole.

RoleURI: The role URI is the URI of the role as defined within an XBRL schema and MUST be provided.

SchemaLocation: The schema location is the location of the schema that contains the role definition and MUST be provided for all roles defined other than those defined by the XBRL specification.

List Association Roles

The list of association roles is a list of XBRL arcroles that are mapped to human readable information. All roles EITHER relate to XBRL presentation, XBRL calculation, or XBRL definition associations. NOTE that this is not a complete list of arcroles but a complete list of supported can be provided.

AssociationName	HumanReadableName	Category	Arcrole	SchemaLocation	StructureType
parent-child	Parent-Child	XBRL	http://www.xbrl.org/2003/arcrole/parent-child		Presentation
summation-item	Total-Item	XBRL	http://www.xbrl.org/2003/arcrole/summation-item		Calculation
class-subclass	Class-Subclass	SBRM	http://xbrlsite.azurewebsites.net/2016/conceptual-model/arcrole/class-subClass	http://xbrlsite.azurewebsites.net/2016/conceptual-model/cm-arcroles.xsd	Definition
general-special	General-Special	XBRL	http://www.xbrl.org/2003/arcrole/general-special		Definition
essence-alias	Essence-Alias	XBRL	http://www.xbrl.org/2003/arcrole/essence-alias		Definition
all	LineItems-Hypercube	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/all	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
dimension-domain	Dimension-RootMember	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/dimension-domain	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
domain-member	RootMember-Member	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/domain-member	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
dimension-default	Dimension-Default	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/dimension-default	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
hypercube-dimension	Hypercube-Dimension	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/hypercube-dimension	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
domain-member	LineItems-Member	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/domain-member	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
domain-member	Member-Member	XBRL Dimensions	http://xbrl.org/int/dim/arcrole/domain-member	http://www.xbrl.org/2005/xbrldt-2005.xsd	Definition
summation-item	Item-Item	XBRL	http://www.xbrl.org/2003/arcrole/summation-item		Calculation
full-hasPart	Full-HasPart	SBRM	http://xbrlsite.azurewebsites.net/2016/conceptual-model/arcrole/full-hasPart	http://xbrlsite.azurewebsites.net/2016/conceptual-model/cm-arcroles.xsd	Definition

AssocationName: This is the ID portion of an arcole definition of a defined arcrole from an XBRL taxonomy schema and MUST be provided.

[63] XBRL International, XBRL Specification. Reference Roles, https://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_5.2.2.2.2

[64] XBRL International, XBRL Specification, Reference Roles, https://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_5.2.3.2.1

[65] XBRL International, Link Role Registry, https://specifications.xbrl.org/registries/lrr-2.0/

HumanReadableName: This is the human readable name of the association that can be used in software applications to avoid using the long, technical looking arcroles that are hard to read and understand by business professionals which MUST be provided.

Category: The category explains where the association role came from, who defined it and MUST be provided and can be used for sorting and filtering association roles.

Arcrole: The arcrole is the URI that has been defined within an XBRL taxonomy schema which defines the arcrole for the association and MUST be provided.

SchemaLocation: The physical schema location of the XBRL taxonomy schema which contains the arcrole definition for the association and MUST be provided.

StructureType: The structure type is the XBRL relation type to which the association role applies and MUST be either Presentation, Calculation, or Definition.

List Units

The list of units enables the use of a human friendly version of units to be provided to a software application user. Units can from the XBRL International Units registry[66] or can be defined by the software user. See the *Framework*[67] documentation for more information.

Key	FriendlyUnits	TechnicalUnits	UseUnitsRef
1	iso4217:USD	<measure>iso4217:USD</measure>	U-Monetary-USD
2	USD	<measure>iso4217:USD</measure>	U-Monetary-USD
3	iso4217:USD/shares	<divide><unitNumerator><measure>iso4217:USD</measure></unitNumerator><unitDenominator><measure>shares</measure></unitDenominator></divide>	U-PerShare
4	USD/shares	<divide><unitNumerator><measure>iso4217:USD</measure></unitNumerator><unitDenominator><measure>shares</measure></unitDenominator></divide>	U-PerShare
5	Shares	<measure>shares</measure>	U-Shares
6	Pure	<measure>pure</measure>	U-Pure
7	Gallons	<measure>gal</measure>	U-Gallons
8	Barrels	<measure>bbl</measure>	U-Barrels
9	iso4217:GBP	<measure>iso4217:GBP</measure>	U-Monetary-GBP
10	iso4217:EUR	<measure>iso4217:EUR</measure>	U-Monetary-EUR
11	iso4217:RUB	<measure>iso4217:RUB</measure>	U-Monetary-RUB

Key: The key is a unique key for the lookup list of units. (NOTE, this probably is not necessary.)

FriendlyUnits: The friendly units is a human readable version of units information that is presented to software users. This field should be less than 100 characters to make it fit will into software application interfaces.

TechnicalUnits: The technical units is the XBRL syntax into which the friendly units is converted when generating XBRL syntax.

UseUnitsRef: The use units ref field is the units reference code for a fact and the units identifier for units information within an XBRL instance. This field should be less than 255 characters, preferably even shorter and no more than 150 characters.

[66] XBRL International, Units Registry, https://specifications.xbrl.org/work-product-index-registries-units-registry-1.0.html

[67] Seattle Method, *Framework*, http://xbrlsite.com/seattlemethod/Framework.pdf

List Currency Codes

This lookup list provides a set of machine readable standard ISO 4217 currency codes[68] that a software application can use within a software application:

Code	CurrencyLabel	Enabled
AED	United Arab Emirates Dirham	FALSE
AFN	Afghanistan Afghani	FALSE
ALL	Albania Lek	FALSE
AMD	Armenia Dram	FALSE
ANG	Netherlands Antilles Guilder	FALSE
AOA	Angola Kwanza	FALSE
ARS	Argentina Peso	FALSE
AUD	Australia Dollar	FALSE

Code: The code is the actual ISO 4216 currency code.

CurrencyLabel: The currency label is the official ISO 4216 currency label.

Enabled: The enabled field allows software to enable or disable currency codes as deemed appropriate by software application users.

[68] Wikipedia, *ISO 4216 Currency Codes*, https://en.wikipedia.org/wiki/ISO_4217