Purpose of Seattle Method

Charles Hoffman, CPA (June 16, 2023)

Effectively, the *Seattle Method* provides a fixed "container" or fixed logical schema[1] for representing (e.g., modeling) financial accounting, reporting, auditing, and analysis events, experience and other information in machine readable form that is also understandable to humans.

It starts with defining logical statements used to represent terms, sets of associations (a.k.a. structures), sets of assertions that govern the associations such as "wider-narrower" rules (a.k.a. general-special rules, type-subtype, class-subclass) or "has-part" rules, mathematical rules and constraints, consistency cross checks that verify high-level concept associations, other constraints and restrictions that describe what is permitted, and how to represent logical statements about facts within this logical scheme so that information is understandable to machine-based processes.

It is important to understand that any financial reporting scheme[2] can be represented using this consistent logical schema which represents a report "meta model" or "container" that each report "model" and report must fit. This enables software to effectively understand and interact with each individual report, a set of report for an economic entity, or a set of reports for any group of peers or other arbitrary grouping of reports.

A financial report is effectively a human-readable but also a machine-readable knowledge graph[3]. The *Seattle Method* leverages this representation plus foundational logic of accounting including the double-entry bookkeeping mathematical model, the fundamental accounting equation[4], and the use of the fundamental accounting equation by standards setters and regulators to define the core elements of financial reporting schemes[5]. This provides both the logical constraints and precise flexibility for financial report models and reports[6]. The accounting and reporting standards themselves provide additional details which must fit into the core higher-level logic. The rules of mathematics, set theory, model theory, and other such universal logic provides further logic used with the logical schema of a financial report.

You can think of the *Seattle Method* as a logic patterns language with exactly the appropriate level of flexibility in exactly the right areas such that things represented using that logic pattern language are always "computable" because the foundational logical schema or "container" of the logical statements that represent terms, structures, relationships, rules, assertions, constraints, and facts never change.

---

[1] Charles Hoffman, CPA, *Logical Schema of Financial Reports*, http://xbrlsite.com/seattlemethod/LogicalSchemaOfFinancialReports.pdf
[2] Charles Hoffman, CPA, *General Purpose Financial Reporting using XBRL*, https://digitalfinancialreporting.blogspot.com/2023/02/general-purpose-financial-reporting.html
[3] Charles Hoffman, CPA, *Financial Report knowledge Graph*, http://xbrlsite.azurewebsites.net/2021/Library/FinancialReportKnowledgeGraphs.pdf
[4] Wikipedia, *Accounting Equation*, https://en.wikipedia.org/wiki/Accounting_equation
[5] Charles Hoffman, CPA, *Comparison of Elements of Financial Statements*, http://xbrlsite.azurewebsites.net/2019/core/ElementsOfFinancialStatements.pdf
[6] Charles Hoffman, CPA, *Essence of Accounting*, http://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf

They are just logical patterns entirely known and understood by the logical schema and therefore by the Seattle Method.

As such, software can be created and then used to effectively reason over those terms, structures, associations, rules, assertions, constraints, and facts represented within different models because the sense-making machinery that is "baked-in" to the capabilities of the *Seattle Method* logical pattern language. For example, expert systems for creating financial reports can be built effectively[7].

The "fixed" way of defining the logical 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*[8]. Further, to take all this to the next level recognize the additional information that follows.

- Remember that a financial report is a specialization of the more general business report. While the focus of the Seattle Method is financial reporting; these same ideas can be applied to general business reporting per the *Standard Business Report Model*[9] (SBRM) which is a logical conceptualization of a general business report.
- Accounting and auditing working papers and schedules also follow the same fundamental logic of financial accounting and reporting. As such, these same ideas can be applied to such working papers and schedules which are simply more granular views of the same information. For more information, please see this prototype working trial balance[10] and other prototype accounting and auditing working papers and schedules[11].
- These ideas can be applied even more generally to the electronic spreadsheet. The *Seattle Method* enables the creation of an additional tool for professional accountants, the "modern spreadsheet"[12] or what I call a *Special Purpose Logical Spreadsheet for Accountants*[13].
- While the ideas in this document tend to focus more on the quantitative information contained within an accounting system these idea apply equally as well to qualitative information contained in financial reports that does not necessarily flow through the actual accounting system.

---

[7] Charles Hoffman, CPA, *Expert System for Cretaing Financial reports Explained in Simple Terms*, http://xbrlsite.azurewebsites.net/2022/Library/ExpertSystemForCreatingFinancialReports.pdf

[8] Ibid

[9] OMG, *Standard Business Report Model* (SBRM), https://www.omg.org/hot-topics/standard-business-report-model.htm

[10] *Working Trial Balance*, https://digitalfinancialreporting.blogspot.com/2023/05/modern-working-trial-balance.html

[11] *Semantic Accounting and Auditing Working Papers*, https://digitalfinancialreporting.blogspot.com/2023/05/semantic-accounting-and-auditing.html

[12] *Modern Spreadsheet*, https://digitalfinancialreporting.blogspot.com/2023/05/modern-spreadsheet.html

[13] Charles Hoffman, CPA, *Special Purpose Logical Spreadsheet for Accountants*, http://www.xbrlsite.com/2023/Library/SpecialPurposeLogicalSpreadsheetsForAccountants.pdf

- The *Seattle Method* will contribute to *The Great Transmutation*[14] of financial accounting, reporting, auditing, and analysis which is impacting the institution of accountancy globally.
- Other potential and already realized transmutations can help accountants, auditors, and analysts understand their transmutation. For example, the switch from paper blueprints to CAD/CAM and ultimately BIM[15] helps one see what might transpire for the institution of accountancy. This article, *A personal digital twin for healthcare*[16], provides a plethora of ideas.

And so, what exactly does the *Seattle Method* do? What is the purpose or function that the Seattle Method serves?

The *Seattle Method* both defines the "container" or "logical schema" into which a report model must fit and also specifies how a report model and report should be verified to evaluate precisely how well a report model and report "fits" into that logical schema.

While the focus of the *Seattle Method* is the logic of a financial report; the *Seattle Method* can also be used to verify that the physical format of the financial report model and report are consistent with the rules provided by XBRL International for XBRL-based digital financial reports in terms of the XBRL Technical Syntax. If some other physical technical formats were used, the XBRL technical syntax verification can simply be "swapped out" for that specific physical technical format.

Currently, the Seattle Method only supports the XBRL technical syntax format only because software applications only support that specific format.

Software Available

The *Seattle Method* was developed over a period of about eight years. During that time, I served as a consultant to a number of software vendors which created functionality which ended up being called the *Seattle Method*. Many aspects of the *Seattle Method* could be replaced by OMG's *Standard Business Report Model* (SBRM). While there could be some differences in terminology and scope; the logic of a business report is universal and really cannot be changed by OMG when they create SBRM. OMG could be more complete or perhaps even less complete or the OMG scope could be enhanced. For example, currently the *Seattle Method* only supports deductive reasoning. OMG could enhance SBRM to also include inductive reasoning and abductive reasoning. But what OMG cannot really do is simply ignore core logic of business reporting which is, as I have said, universal.

There were six different software developers/vendors that implemented functionality which became known as the *Seattle Method*:

- XBRL Cloud,
- 28msec,
- Pesseract (which is a working proof of concept developed by Hamed Mousavi[17],

[14] *The Great Transmutation*, https://digitalfinancialreporting.blogspot.com/2023/06/the-great-transmutation-of-financial.html
[15] *Using Difference Between CAD/CAM and BIM to Understand How to Create Financial Reporting Expert Systems*, https://digitalfinancialreporting.blogspot.com/2023/03/using-difference-between-cadcam-and-bim.html
[16] *A personal digital twin for healthcare*, https://patternslanguage.com/articles/f/a-personal-digital-twin-for-healthcare
[17] LinkedIn, Hamed Mousavi, https://www.linkedin.com/in/hamed-mousavi-286991b2/

- General Luca (which is a partially completed working proof of concept developed by Yury Volkovich[18]),
- Auditchain Pacioli[19],
- Auditchain Luca[20]

Of the six, all support putting report model and report information into the Seattle Method logical schema and all support some level of verification that what was put into that logical schema was consistent with what was deemed permitted.

However, only two support 100% of the current logical schema verification: Pacioli and Luca. Pesseract supports everything except XBRL Technical Syntax Verification and Type-subtype Associations Verification.

Summary of Verification

The following is a brief overview of the different categories of verification required by the Seattle Method. This summary is explained upon I the section which follows this section

1. Proper XBRL technical syntax used to represent logic
2. Proper and complete set of XBRL calculations (i.e. roll ups)
3. Proper and complete set of other mathematical relations expressed using XBRL Formulas (e.g., roll forward, adjustment, dimensional aggregation, variance, arithmetic)
4. Proper XBRL presentation relations associations (e.g., proper report model, supplement to XBRL technical syntax verification not covered by XBRL specification)
5. Proper relationships between high-level financial reporting concepts (e.g., not inconsistent, not contradictory high-level financial report line items
6. Proper use of a type of line item as if were some different type of line item (e.g., type-subtype or wider-narrower or general-special associations)
7. Proper logical mechanical structure of represented disclosures which are provided within report model
8. Proper reporting of each required disclosure consistent with financial reporting standards used (e.g., reporting checklist)
9. Other (e.g., manual checks and other additional constraints added such as XBRL US Data Quality rules)

The following are how each application summarizes the above nine categories:

[18] LinkedIn, Yury Volkovich, https://www.linkedin.com/in/yury-volkovich-66827849/
[19] Auditchain, *Pacioli Logic and Rules Engine*, https://docs.auditchain.finance/auditchain-protocol/pacioli-logic-and-rules-engine
[20] *World's First Standards Based Expert System for Creating Financial Reports*, https://digitalfinancialreporting.blogspot.com/2023/01/worlds-first-standards-based-expert.html

Auditchain Luca

https://dev.auditchain.finance/

LOCAL (this is a work in progress)

#	Verification Category	Result
1	XBRL Technical Syntax Verification	
2	Report Mathematical Computations Verification (XBRL Calculations)	✓
3	Report Mathematical Computations Verification (XBRL Formula)	✓
4	Report Model Structure Verification	
5	Fundamental Accounting Concept Consistency Crosschecks Verification	✓
6	Type-subtype (wider-narrower) Associations Verification	✓
7	Disclosure Mechanics Verification	✓
8	Report Disclosure Checklist Verification	✓
9	Other	

Using Pacioli Verification

#	Verification Category	Result
1	XBRL Technical Syntax Verification	✓
2	Report Mathematical Computations Verification (XBRL Calculations)	✓
3	Report Mathematical Computations Verification (XBRL Formula)	✓
4	Report Model Structure Verification	✓
5	Fundamental Accounting Concept Consistency Crosschecks Verification	✓
6	Type-subtype (wider-narrower) Associations Verification	✓
7	Disclosure Mechanics Verification	✓
8	Report Disclosure Checklist Verification	✓
9	Other	✓

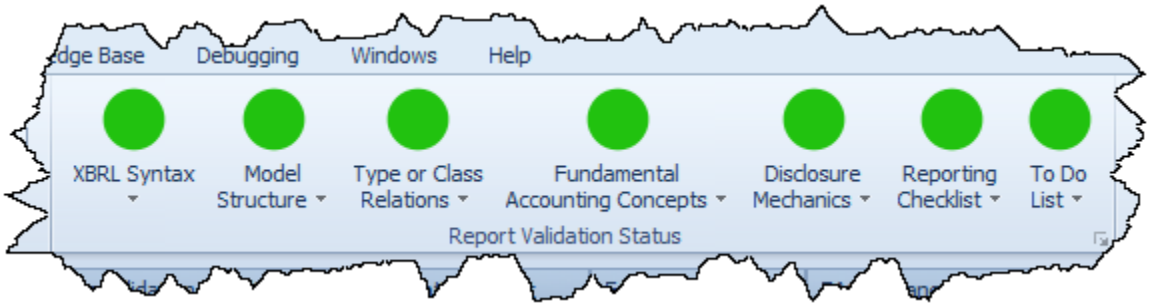
Auditchain Pacioli

https://auditchain.infura-ipfs.io/ipfs/Qmf1vKbKxCzTvtE3gLxaTgdu7vS5cWjPfnBHFqKdVC5cXH/

#	Verification Category	Result
1	XBRL Technical Syntax Verification	
2	Report Mathematical Computations Verification (XBRL Calculations)	
3	Report Mathematical Computations Verification (XBRL Formulas)	
4	Report Model Structure Verification	
5	Fundamental Accounting Concept Consistency Crosschecks Verification	
6	Type-subtype (wider-narrower) Associations Verification	
7	Disclosure Mechanics Verification	
8	Report Disclosure Checklist Verification	
9	Other	

Pesseract (working proof of concept)

http://pesseract.azurewebsites.net/



XBRL Cloud

http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/evidence-package/

http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html

Verification Summary

	A	M
XBRL Technical Syntax Rules	OK	OK
EFM Rules	OK	OK
XBRL-US Consistency Suite Rules	S	S
Model Structure Rules (US GAAP Taxonomy Architecture)	OK	OK
US GAAP Domain Level Rules	NS	NS
Fundamental Accounting Concepts and Relations Rules	OK	OK
XBRL-US Data Quality	OK	1
Notes Consistency	OK	OK
US GAAP Industry / Activity Specific Rules	NS	NS
Reporting Entity Specific Rules	OK	4
Reporting Entity Specific Roll Up Rules	OK	20
US GAAP Reportability Rules	NS	NS
Other Manual Review Tasks	NS	NS
Other Rules and Best Practice Tasks	OK	OK

Detailed Explanation of Verification Categories

The following is a detailed explanation of each verification category. Additional details and explanations are provided in the document *Essentials of XBRL-based Digital Financial Reporting*[21].

In the information below, the SFAC 6 prototype financial reporting scheme[22] will be used to provide examples. The Pacioli Technical Analysis of SFAC 6[23] will be used to make this information available to the reader of this documentation. These tests can be repeated by others using the metadata provided and the report model and report prototype reference implementation provided[24].

An additional verification report is provided by XBRL Cloud which only verifies certain aspects of this SFAC 6 reference implementation of a report model and report[25].

For each of the verification categories a number of examples will be provided to help the reader understand the purpose of the Seattle Method in the creation of provably high-quality XBRL-based financial reports.

Proper XBRL technical syntax used to represent logic

This category of verification assures that an XBRL-based report model and report are consistent with the XBRL technical syntax by using a fully compliant XBRL processor to test to be sure the XBRL syntax of such report and report model are consistent with the XBRL International conformance suites for XBRL, XBRL Dimensions, XBRL Formulas, and any other applicable modules of XBRL.

Providing and explaining all of the XBRL International conformance suites is beyond the scope of this explanation. There are literally thousands of tests to be sure XBRL processing is performed correctly. Inconsistencies with the conformance suite is a sign of problems. For more information, please see the XBRL International technical specifications which provides each conformance suite. For example, the conformance suite for the core XBRL 2.1 specification[26]. Hundreds of tests exist to test software to be sure the software is processing XBRL-based report models and reports appropriately.

Proper and complete set of XBRL calculations

This category of verification assures that an XBRL-based report model and report are consistent with XBRL calculations rules.

---

[21] Charles Hoffman, CPA, *Essentials of XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2021/essentials/EssentialsOfXBRLBasedDigitalFinancialReporting.pdf

[22] SFAC 6 Financial Reporting Scheme Prototype, http://xbrlsite.com/seattlemethod/golden/sfac6/sfac6_ModelStructure.html

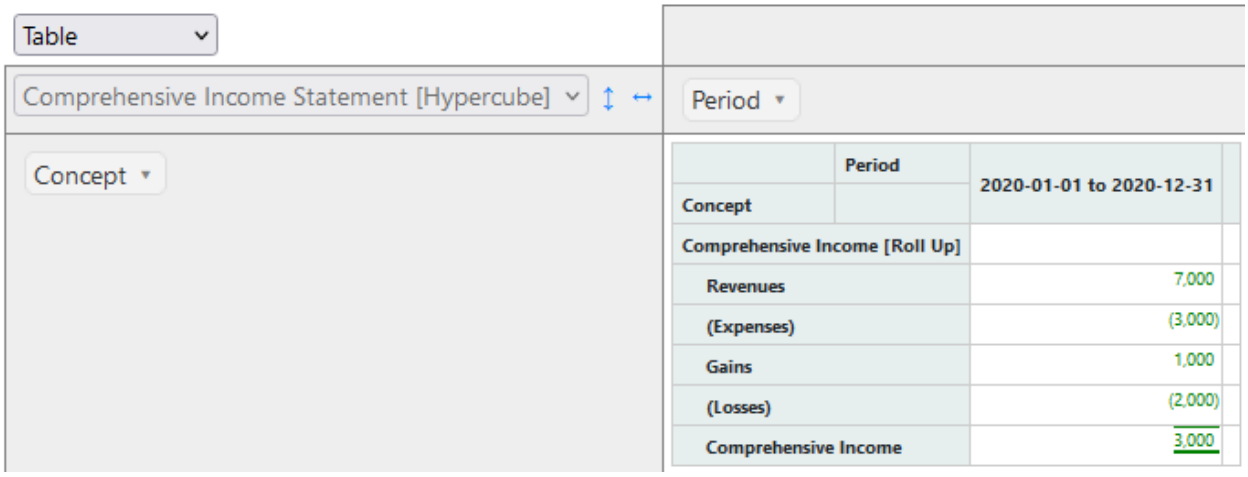
[23] Pacioli Technical Analysis of SFAC 6 prototype report, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/

[24] SFAC 6 report model and report reference implementation, http://xbrlsite.com/seattlemethod/golden/sfac6/instance.xml

[25] XBRL Cloud verification information, http://xbrlsite.com/seattlemethod/golden/sfac6/evidence-package/contents/index.html#Rendering-N0-RE6.html

[26] XBRL International, XBRL 2.1, https://specifications.xbrl.org/work-product-index-group-base-spec-base-spec.html

For example, this calculation which exists in the SFAC 6 financial reporting scheme prototype[27]:



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

Was verified using the provided XBRL calculations rule shown by the Pacioli verification of the XBRL calculation[28].

#	Type	Name	Rule Expression
1	calculation	sfac6:ComprehensiveIncome (in ComprehensiveIncome) • ok:1 • failed:0	total=Revenues_4+(-Expenses_3+(Gains_2+ -Losses_1)) *1 instance:* *sfac6:ComprehensiveIncome[3,000] = sfac6:Revenues[7,000] + - sfac6:Expenses[3,000] + sfac6:Gains[1,000] + - sfac6:Losses[2,000]*

XBRL Cloud's representation of this same verification is shown below[29]:

	Rendered		Reported	Calculated	Balance	Decimals	Message
Comprehensive Income [Roll Up]							
Revenues	7,000	+	7,000	7,000	CR	INF	
(Expenses)	(3,000)	-	3,000	3,000	DR	INF	
Gains	1,000	+	1,000	1,000	CR	INF	
(Losses)	(2,000)	-	2,000	2,000	DR	INF	
Comprehensive Income	3,000		3,000	3,000	CR	INF	OK

Proper and complete set of other mathematical relations expressed using XBRL Formulas

This category of verification assures that an XBRL-based report model and report are consistent with all other mathematical computations not supported by XBRL calculations, but rather where XBRL Formula must be used to verify those mathematical computations.

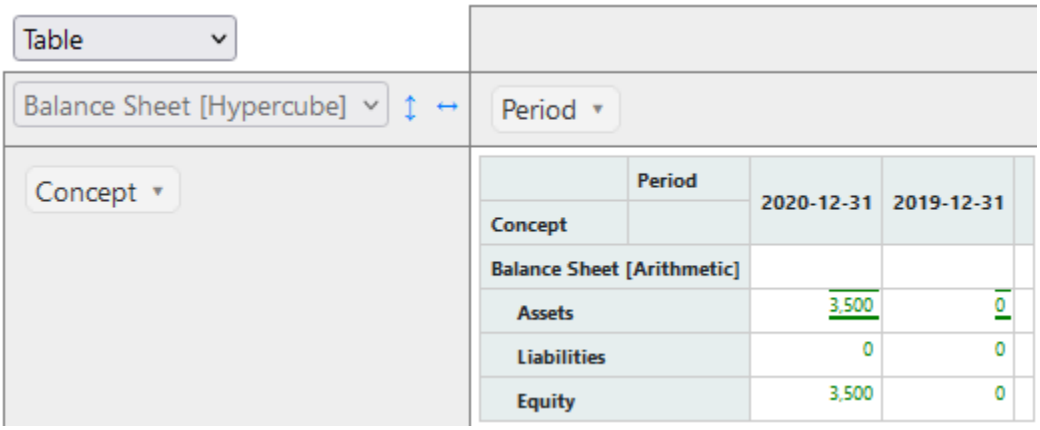
[27] Pacioli structure with calculation, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/35e817d0b0758e497560.html#4f547c55004cbb16aa20

[28] Pacioli verification of XBRL calculation, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/calculations.html

[29] XBRL Cloud verification of calculation, http://xbrlsite.com/seattlemethod/golden/sfac6/evidence-package/contents/index.html#Rules-ComprehensiveIncome-sfac6_ComprehensiveIncomeStatementHypercube.html

This includes roll ups, adjustments, dimensional roll ups, variance, general arithmetic, and other such mathematical associations.

For example, this “Arithmetic” type of mathematical relation exists within the balance sheet of the SFAC 6 reference implementation prototype report[30]:



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

For this “Arithmetic” relation in the SFAC 6 report in the balance sheet the following information verifies that the report information is consistent with the provided rule which is shown below using the Pacioli technical analysis[31]:

#	Type	Name	Rule Expression
1	valueAssertion	Arithmetic_BS01 • ok:2 • failed:0	Assets=Liabilities+Equity *2 instances:* *sfac6:Assets[0] = sfac6:Liabilities[0] + sfac6:Equity[0]* *sfac6:Assets[3,500] = sfac6:Liabilities[0] + sfac6:Equity[3,500]*

XBRL Cloud provides the following summary of other XBRL Formula based rules for the report which are provided and then text the report information[32].

#	Label	Result	Rule
1	$ComprehensiveIncome = ($Revenues - $Expenses + $Gains - $Losses) (Arithmetic_IS01)	Pass	$ComprehensiveIncome = ($Revenues - $Expenses + $Gains - $Losses)
2	0 = (Equity{T0} + (Revenue{P1} - Expenses{P1} + Gains{P1} - Losses{P1}) + (InvestmentsByOwners{P1} - DistributionsToOwners{P1})) + Liabilities{T1} - Assets{T1} (CONSISTENCY_sfac6_ElementsOfFinancialStatement)	Pass	0= (($Equity_BalanceStart + (($Revenues - $Expenses) + ($Gains - $Losses)) + ($InvestmentsByOwners - $DistributionsToOwners)) + ($Liabilities_BalanceEnd - $Assets_BalanceEnd))

[30] Pacioli information for balance sheet, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/830e87352633019b7a23.html#8fb33d5983660001f613

[31] Pacioli technical analysis for balance sheet, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/valueAssertions.html

[32] XBRL Cloud Business Rules, http://xbrlsite.com/seattlemethod/golden/sfac6/evidence-package/contents/index.html#BusinessRulesSummary.html

Proper XBRL presentation relations associations

This category of verification assures that the relations represented in the XBRL presentation portion of the report model are consistent with fundamental logic. A set of rules is provided, a matrix which establishes the allowed children for specific categories of report elements. This table below shows a matrix of the permitted and disallowed relations between the categories of report elements provides an example:

		Parent						
		Network	Hypercube	Dimension	Member	LineItems	Abstract	Concept
Child	Network	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL
	Hypercube	Permitted	Disallowed	Disallowed	Disallowed	Disallowed	Permitted	Disallowed
	Dimension	Disallowed	Permitted	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Member	Disallowed	Disallowed	Permitted	Permitted	Disallowed	Disallowed	Disallowed
	LineItems	Disallowed	Permitted	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Abstract	Permitted	Disallowed	Disallowed	Disallowed	Permitted	Permitted	Disallowed
	Concept	Disallowed	Disallowed	Disallowed	Disallowed	Permitted	Permitted	Disallowed

Report elements in an XBRL-based report model can be grouped into the following categories: Network, Hypercube, Dimension, Member, Line Items, Abstract and Concept. The relationship between the report element type of the parent and the report element type of the child in a set of XBRL presentation relations is documented in the matrix. For example, a “Network” (parent) may contain a Hypercube (child) report element.

Below you see the model structure (e.g. XBRL presentation relations) for a network of the SFAC 6 example. The human readable representation can be used to understand the “parent” to “child” relations. Indentation is used to show the parents and the children. For example, the Abstract type report element “sfac6:BalanceSheetArithmetic” has three children “sfac6:Assets”, “sfac6:Liabilities”, and “sfac6: Equity”[33].

Model Structure	
Network	**11-Statement of Financial Position** *(http://luca.auditchain.finance/report/role/BalanceSheet)*
Table	Balance Sheet [Hypercube]

#	Label	Report Element Class	Period Type	Balance	Name
1	Balance Sheet [Hypercube]	Hypercube			sfac6:BalanceSheetHypercube
2	Balance Sheet [Line Items]	LineItems			sfac6:BalanceSheetLineItems
3	Balance Sheet [Arithmetic]	Abstract			sfac6:BalanceSheetArithmetic
4	Assets	Concept (Monetary)	instant	debit	sfac6:Assets
5	Liabilities	Concept (Monetary)	instant	credit	sfac6:Liabilities
6	Equity	Concept (Monetary)	instant	credit	sfac6:Equity

[33] Pacioli model structure relations human readable representation, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/830e87352633019b7a23.html#e38a7599ae597d7afcf8

The relations information can be represented in the form of a matrix that matches the allowed and disallowed example. For example, there are Pacioli verification of presentation relations within a model structure[34]:

Child	Parent						
	Network	Hypercube (a.k.a. Table)	Dimension (a.k.a. Axis)	Member	Line Items (a.k.a. Primary Items)	Abstract	Concept
Network	0	0	0	0	0	0	0
Hypercube (a.k.a. Table)	4	0	0	0	0	0	0
Dimension (a.k.a. Axis)	0	0	0	0	0	0	0
Member	0	0	0	0	0	0	0
Line Items (a.k.a. Primary Items)	0	4	0	0	0	0	0
Abstract	1	0	0	0	4	3	0
Concept	0	0	0	0	0	16	0

Proper relationships between high-level financial reporting concepts

Financial reporting standards define elements of financial reports and the high-level relationships between those elements. At the highest level is the accounting equation: Assets = Liabilities + Equity. From that high point other disaggregations are defined: Assets = Current Assets + Noncurrent Assets. Here is an example of the high level relations of financial reports[35]:

Line		Object Class	Period Type	Balance	Report ElementName
1	11.0-Balance Sheet Classified (BSC)	Network			http://www.xbrlsite.com/essence/role/BalanceSheetClassified
2	**Balance Sheet Classified [Hypercube]**	Hypercube			essence:BalanceSheetClassifiedHypercube
3	**Balance Sheet [Line Items]**	LineItems			essence:BalanceSheetLineItems
4	Assets [Roll Up]	Abstract			essence:AssetsRollUp
5	Current Assets	Concept (Monetary)	As Of	Debit	essence:CurrentAssets
6	Noncurrent Assets	Concept (Monetary)	As Of	Debit	essence:NoncurrentAssets
7	Assets	Concept (Monetary)	As Of	Debit	essence:Assets
8	Liabilities Equity [Roll Up]	Abstract			essence:LiabilitiesEquityRollUp
9	Liabilities [Roll Up]	Abstract			essence:LiabilitiesRollUp
10	Current Liabilities	Concept (Monetary)	As Of	Credit	essence:CurrentLiabilities
11	Noncurrent Liabilities	Concept (Monetary)	As Of	Credit	essence:NoncurrentLiabilities
12	Liabilities	Concept (Monetary)	As Of	Credit	essence:Liabilities
13	Commitments and Contingencies	Concept (Monetary)	As Of	Credit	essence:CommitmentsAndContingencies
14	Temporary Equity	Concept (Monetary)	As Of	Credit	essence:TemporaryEquity
15	Equity [Roll Up]	Abstract			essence:EquityRollUp
16	Equity Attributable To Parent	Concept (Monetary)	As Of	Credit	essence:EquityAttributableToParent
17	Equity Attributable To Noncontrolling Interest	Concept (Monetary)	As Of	Credit	essence:EquityAttributableToNoncontrollingInterest
18	Equity	Concept (Monetary)	As Of	Credit	essence:Equity
19	Liabilities and Equity	Concept (Monetary)	As Of	Credit	essence:LiabilitiesAndEquity

Part of verifying that a financial report is function properly is to check for inconsistencies, contradictions, and other such issues in these high-level financial reporting concepts. These consistency cross checks are important.

Below you see an example of the verification of the fundamental accounting concepts that make up a financial report[36]:

---

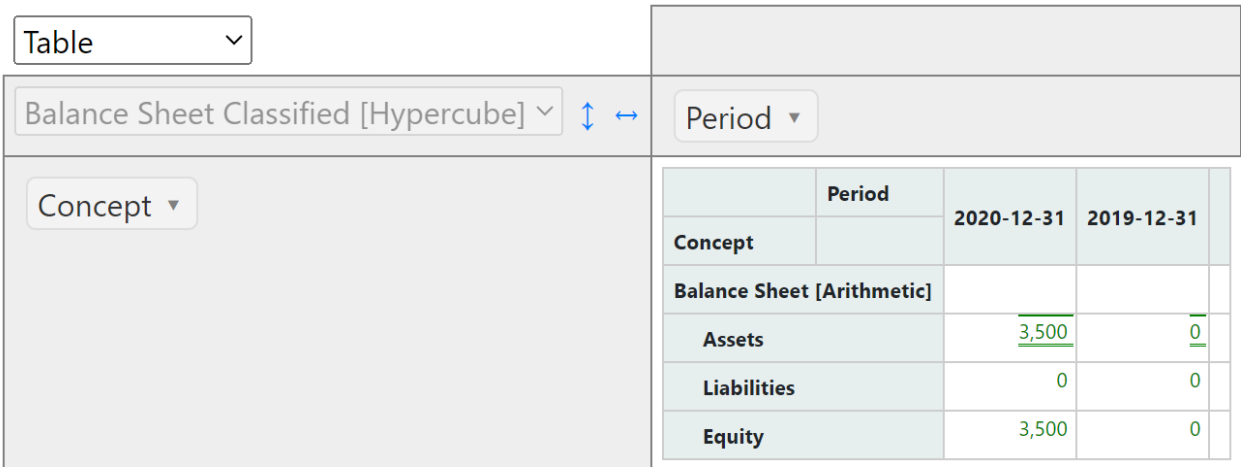
[34] Pacioli verification of model structure, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/modelStructure.html

[35] Example of high level relationships in financial reports, http://www.xbrlsite.com/seattlemethod/golden/essence/essence_ModelStructure.html

[36] XBRL Cloud, Fundamental Accounting Concepts Verification Report, https://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/evidence-package/USFACRenderingSummary.html

Balance Sheet [Line Items]	Period [Axis] 2017-06-30 Fact Value	Origin
Assets [Roll Up]		
Current Assets	159,851,000,000	fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]
Noncurrent Assets	81,235,000,000	fac:NoncurrentAssets[81,235,000,000] = fac:Assets[us-gaap:Assets[241,086,000,000]] - fac:CurrentAssets[us-gaap:AssetsCurrent[159,851,000,000]]
Assets	241,086,000,000	fac:Assets[us-gaap:Assets[241,086,000,000]]
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	64,527,000,000	fac:CurrentLiabilities[us-gaap:LiabilitiesCurrent[64,527,000,000]]
Noncurrent Liabilities	104,165,000,000	fac:NoncurrentLiabilities[104,165,000,000] = fac:Liabilities[us-gaap:Liabilities[168,692,000,000]] - fac:CurrentLiabilities[us-gaap:LiabilitiesCurrent[64,527,000,000]]
Liabilities	168,692,000,000	fac:Liabilities[us-gaap:Liabilities[168,692,000,000]]
Commitments and Contingencies	0	fac:CommitmentsAndContingencies[us-gaap:CommitmentsAndContingencies[0]]
Temporary Equity	0	fac:TemporaryEquity[0] = fac:LiabilitiesAndEquity[us-gaap:LiabilitiesAndStockholdersEquity[241,086,000,000]] - ((fac:Liabilities[us-gaap:Liabilities[168,692,000,000]] + fac:Equity[72,394,000,000]) - fac:CommitmentsAndContingencies[us-gaap:CommitmentsAndContingencies[0]])
Equity [Roll Up]		
Equity Attributable to Parent	72,394,000,000	fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]]
Equity Attributable to Noncontrolling Interest	0	fac:EquityAttributableToNoncontrollingInterest[0] = fac:Equity[72,394,000,000] - fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]]
Equity	72,394,000,000	fac:Equity[72,394,000,000] = fac:EquityAttributableToParent[us-gaap:StockholdersEquity[72,394,000,000]]
Liabilities and Equity	241,086,000,000	fac:LiabilitiesAndEquity[us-gaap:LiabilitiesAndStockholdersEquity[241,086,000,000]]

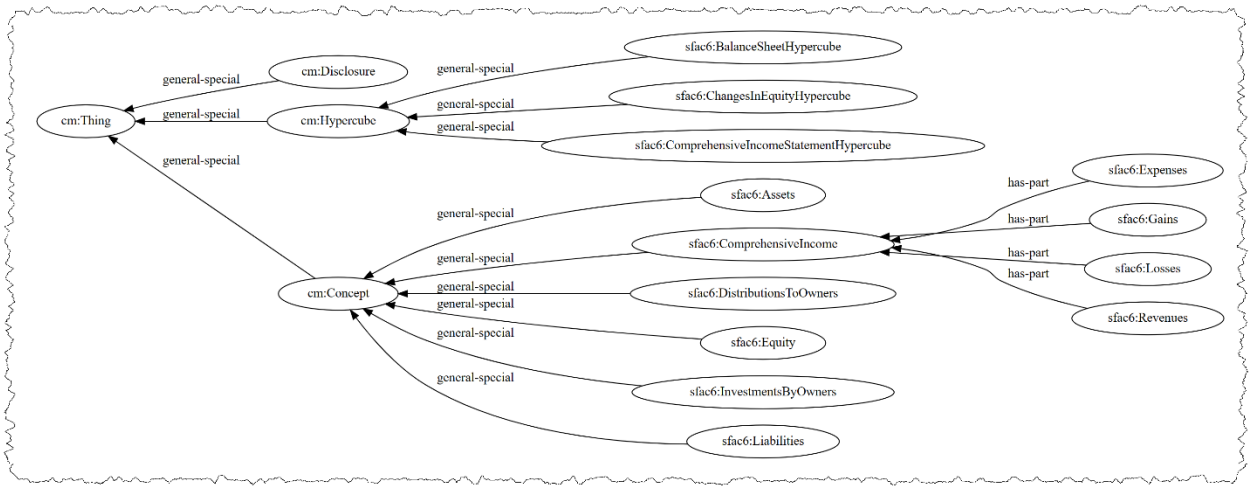
Here you see the verification of the accounting equation relation for the SFAC 6 example which we are using to provide examples[37]:



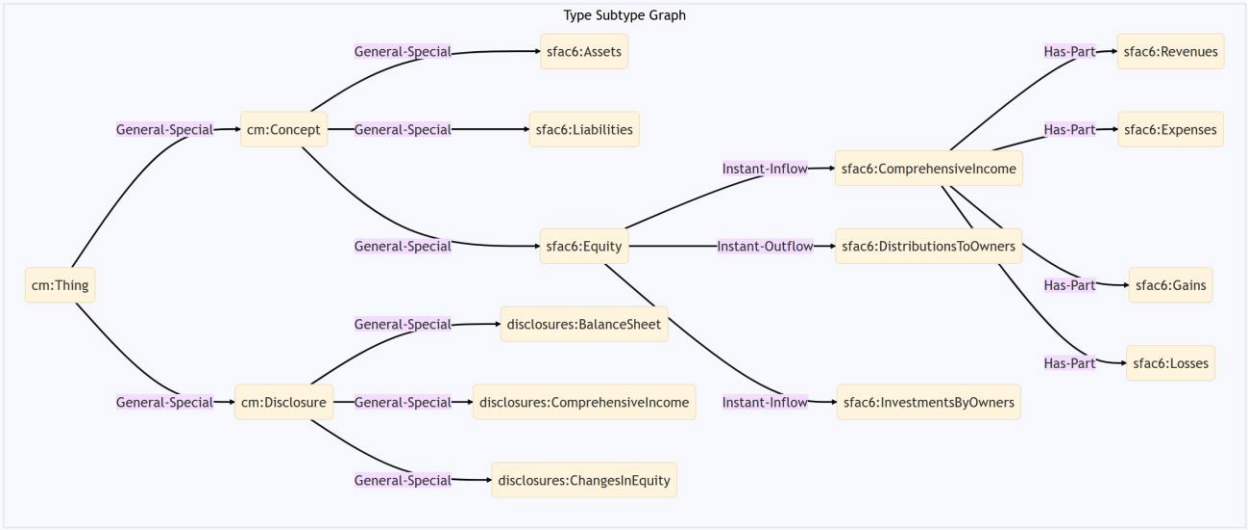
[37] Auditchain Pacioli verification report, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/09a7caeb8c183c28fc01.html#097ea0724d4e7bf68375

Proper use of a type of line item as if were some different type of line item

Building on the high level financial concept associations are the “type-subtype” (a.k.a. “wider-narrower” or “general-special” relations. Here is an example of those relations[38]:



Here is another view of the type-subtype assocations provided by Auditchain’s Luca software application:



What the type-subtype associations do is check to be sure the concepts used in mathematical relations match with what might be expected. For example, for the SFAC 6 report prototype, only the following concepts are part of “sfac6:ComprehensiveIncome”: sfac6:Revenues, sfac6:Expenses, sfac6:Gains, sfac6:Losses.

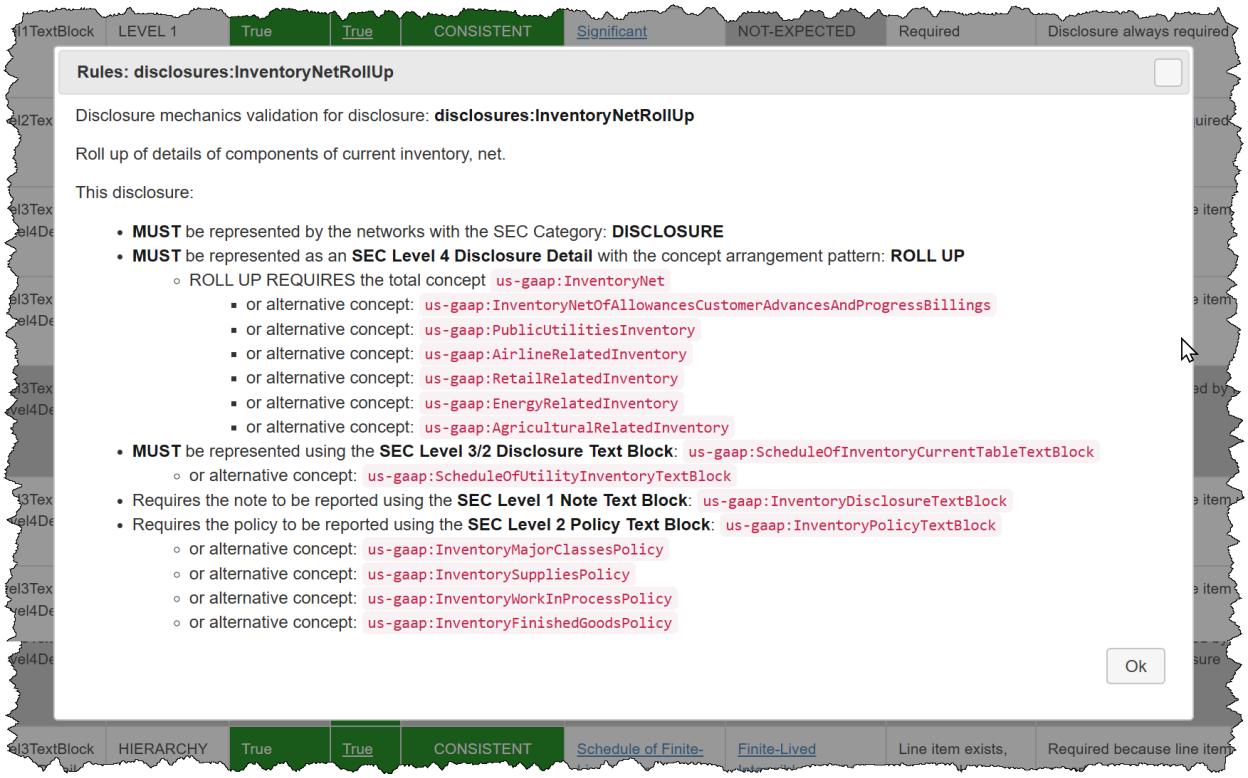
If a report creator erroneously used the concept “sfac6:InvestmentsByOwners” as part of sfac6:ComprehensiveIncome; then the software application would make the user of the software aware of this inappropriate use of the concepts provided for the financial reporting scheme.

[38] Pacioli Technical Analysis, Type-subtype associations graph, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/typeSubTypeGraph.html

Proper logical mechanical structure of represented disclosures which are provided within report model

You can think of the disclosure mechanics rules as a representation of the essence of a financial report disclosure. When a reporting economic entity constructs its financial report, the model constructed must be consistent with what is logical and what is expected for each specific disclosure. This category of verification checks to be sure each disclosure is within the boundaries of what is expected.

The following disclosure mechanics verification report shows a rather complex example of a disclosure mechanics rule[39]:



Explaining all the details of what is going on with this disclosure mechanics verification is beyond the scope of this document.

A much simpler example for the SFAC 6 prototype is provided here by Auditchain's Pacioli[40]:

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

[40] Auditchain Pacioli Disclosure Mechanics Rules Verification, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/disclosures.html

#	Type	Name	Rule Expression
1	disclosure	disclosures:BalanceSheet **Added** • detections:1	*Balance Sheet* • disclosures:BalanceSheet **requires:** • Concept Arrangement Pattern cm:Arithmetic ○ **with** sfac6:Assets • Concept sfac6:Liabilities • Concept sfac6:Equity *1 instance:* *In network 11-Statement of Financial Position:* *sfac6:Liabilities is presented* *sfac6:Equity is presented* *Detected block Assets [Arithmetic] with sfac6:Assets*

What the disclosure mechanics rules say is the following: “The Balance Sheet disclosure has an information pattern (concept arrangement pattern) of a cm:Arithmetic pattern and one of the concepts used in that information modal MUST be the concept “sfac6:Assets”. In addition, the concept “sfac6:Liabilities” and concept “sfac6:Equity” MUST also appear on the Balance Sheet or must be derived from other reported information.

Proper reporting of each required disclosure consistent with financial reporting standards used

Per financial reporting rules defined by a financial reporting scheme, specific financial disclosure are required to be provided given specific circumstances related to a reporting economic entity. For example,

- A financial reporting scheme might say that a balance sheet, an income statement, statement of comprehensive income, a cash flow statement, and a statement of changes in equity are always required to be provided within a financial report.
- Alternatively, a combine income statement and statement of comprehensive income may be allowed.
- Specific financial disclosures might always be required such as the “Basis of Reporting” and “Nature of Operations”.
- If specific line items are reported on the balance sheet or income statement; then one or more specific disclosures might be required. So for example, if the line item “Inventories” is reported, then a disclosure of the disaggregation of that inventory line item might be required and an inventories policy might be expected.

This category of verification tests the provided disclosures to be sure that the disclosure rules of a financial reporting scheme is being followed. While not all disclosure rules can be represented in machine readable form, many rules can be represented.

XBRL Cloud provides an example of what I refer to as a reporting checklist[41]:

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

Auditchain's Pacioli provides this version of a reporting checklist for the SFAC 6 financial report prototype that we are using to explain these verification rules[42]:

#	Type	Name	Rule Expression
1	disclosureCheck	require [disclosures:BalanceSheet] **Added** • ok:1 • failed:0	Require disclosure: • disclosures:BalanceSheet *1 instance:* *See link above*
2	disclosureCheck	require [disclosures:ChangesInEquity] **Added** • ok:1 • failed:0	Require disclosure: • disclosures:ChangesInEquity *1 instance:* *See link above*
3	disclosureCheck	require [disclosures:ComprehensiveIncome] **Added** • ok:1 • failed:0	Require disclosure: • disclosures:ComprehensiveIncome *1 instance:* *See link above*

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

[42] Auditchain Pacioli Reporting Checklist, https://auditchain.infura-ipfs.io/ipfs/QmdZhQeZg8PfU7Ce3ApkZTaJBK1FzzhkRrnVCoFg1jVBHv/disclosureChecks.html

Other

Any other additional constraints or restrictions could be added to a software application that may enable the automation of verification checks to be sure that a financial report is properly functioning in terms of financial reporting or accounting logic. A few examples of these other additional categories of verification might include:

- XBRL US Disclosure Quality Committee verification rules[43].
- Additional system specific rules required by a regulator such as the EDGAR Filer Manual (EFM) rules required by the U.S. Securities and Exchange Commission (SEC)[44], for example the specific ordering of information within a report model
- Additional technical format restrictions and constraints required by a specific technical format such as the European Single Electronic Format (ESEF) rules used by the European Single Market Authority (ESMA)[45], for example the use of a specific URI to identify economic entities
- Specific restrictions on the data types used to report facts defined using XML Schema PART 2 Data Types specification[46]
- Finally, a spell checker could be added!

The different types of additional verification which can possibly be added is almost endless. All that one really needs to do is to put the rules that need to be enforced by the verification into machine readable form.

Conclusion

While the final verification category might not be used; the prior eight verification categories are absolutely necessary in order to be sure a digital financial report has been correctly represented and proven to be a properly functioning logical system.

As is said, "garbage in, garbage out". Not providing machine readable rules and therefore not having automated verification processes not report errors because the report model is incomplete is not an effective approach to changing information electronically. Ignorance of these verification categories is no excuse, it is the job of a master craftsperson to understand these requirments.

For additional information please see *Essentials of XBRL-based Digital Financial Reporting*[47] and the *Seattle Method*[48].

[43] XBRL US, Disclosure Quality Committee, https://xbrl.us/data-quality/center/committee/
[44] SEC, EDGAR Filer Manual, https://www.sec.gov/edgar/filermanual
[45] ESMA, ESEF Filer Manual, https://www.esma.europa.eu/document/esef-reporting-manual
[46] W3C, XML Schema Part 2: Data Types Second Edition, https://www.w3.org/TR/xmlschema-2/
[47] Charles Hoffman, CPA, *Essentials of XBRL-based Digital Financial Reporting*, http://xbrlsite.azurewebsites.net/2021/essentials/EssentialsOfXBRLBasedDigitalFinancialReporting.pdf
[48] Charles Hoffman, CPA, *Seattle Method*, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf