Understanding Logical Objects of XBRL-based Digital Financial Reports

Interacting with the high-level logical objects of an XBRL-based digital report models and reports

By Charles Hoffman, CPA (April 15, 2024) (Work in Progress)

You can consider an XBRL-based financial report from the perspective of the technical syntax of that report or you can consider that same report per the logic of the information represented within or by that XBRL technical format.

This document helps the reader consider the logical objects within an XBRL-based digital financial report model and report. Before reading this document, it is strongly suggested that the reader work through the document *Understanding and Leveraging the "Semantic Glue" of XBRL-based Financial Reports*[1].

"*Magic is when you command the elements to capture the flow of grace*." Carlos Santana.

Financial Report Logical Objects

An XBRL-based digital financial report can be interacted with per the XBRL technical format or syntax. But that same report can be loaded into software and then one can interact with the logic that is being represented by that XBRL technical format.

This section helps the reader understand the difference between the technical format and the report logic and helps the reader understand the logical objects used to work with such digital reports.

Assemblies (a.k.a. Organisms, Frames)

We will focus on the high level of a financial report, the "assemblies" or what the Atomic Design Method refers to as "organisms". An assemble or organism is a high-level logical artifact of a report. Every financial report can be broken down into a set or collection of these high-level assemblies/organisms.

Boundaries, Guardrails, Bumpers

Some have called them "boundaries". Others "guardrails". Still others "bumpers". But what software does is force users to stay within the borders of what is permitted, not letting them do things that are not permitted. This both helps the user get the task or process that they are trying to complete done (i.e. computer augmenting the skills of the software users) and makes sure the quality of the work product is where it needs to be.

Software understands the logic of the objects that it is working with. Software will not allow users to "color outside the lines". Why is this important? Think of 5,000 different reporting

[1] *Understanding and Leveraging the "Semantic Glue" of XBRL-based Financial Reports*, http://xbrlsite.com/2024/Library/UnderstandingAndLeveragingSemanticGlue.pdf

managers from public companies submitting XBRL-based reports to the U.S. Securities and Exchange Commission (SEC) using 50 different software applications. "Wild behavior" by accountants creating those reports cannot be permitted. Clear specifications, descriptions, and verification enable effective extraction and reuse of information.

Report Used to Explain Logical Objects

This document uses a very basic report model and report to help the reader make the distinction between report technical format and the report logic and then to explain the notion of logical objects or logical assemblies.

The following is information related to the report used in this document to show the logical objects of financial report models and reports:

Reporting scheme: (including all verification rules)

http://www.xbrlsite.com/seattlemethod/golden/common3/base-taxonomy/common_ModelStructure.html

Report model and report:

https://dev.auditchain.finance/storage/c5344192-b624-487d-8c22-11e42ba5963f/f1cc6cd2/instance.xml

Verification rules Used:

['http://xbrlsite.com/seattlemethod/cm/model-structure-rules-strict-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/disclosure-mechanics/dm.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/fac/REFERENCE/fac.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/reporting-checklist/dr-rules-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/type-subtype/typeSubtype-rules-BEST-def.xml']

Verification results (Auditchain Pacioli):

https://auditchain.infura-ipfs.io/ipfs/QmPxuHQ9d9WvyQd5dkmst1cJPg1Zp1zAGJGYxs2dXsfCuP/

#	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	✔

Verification results (Auditchain Luca):

#	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	

State properties of report model and report:

State Properties

Reporting scheme	2
Fact count	37
Structure count	7
Hypercube count	7
Hypercube (explicit)	7
Hypercube (implied)	0
Block count	9
Disclosure count	11
Pattern count	9
Set	
Roll Up	7
Roll Forward	2
Roll Forward Info	
Adjustments	
Variances	
Text Blocks	
Member Aggregation	
Arithmetic	
Validation inconsistencies depending on SBRM	0
XBRL Calculation	0
XBRL Formula	0
Model structure	0

Report Elements

	All	Local (Added)	Percent
Structure	7	7	100.00%
Hypercube	7	0	0.00%
Dimensions	0	0	0.00%
Members	0	0	0.00%
LineItems	1	0	0.00%
Abstracts	10	0	0.00%
Concepts	26	0	0.00%

Blocks

#	Block name	Patterns
1	Assets [Roll Up]	RollUp
2	Liabilities and Equity [Roll Up]	RollUp
3	Net Assets [Roll Up]	RollUp
4	Net Income [Roll Up]	RollUp
5	Net Income [Roll Up]	RollUp
6	Comprehensive Income [Roll Up]	RollUp
7	Net Cash Flow [Roll Up]	RollUp
8	Assets [Roll Forward]	RollForward
9	Changes in Equity [Roll Forward]	RollForward

Disclosures

Reporting Style - REFERENCE IMPLEMTATION

Approach

The approach used to explain the logical objects will be to first have a look at the technical report objects and then the logical report objects to let the reader see the difference between the technical and logical perspectives. The focus is on the high-level structures which hold report information. Just like the human body is not just a bunch of random organisms; a

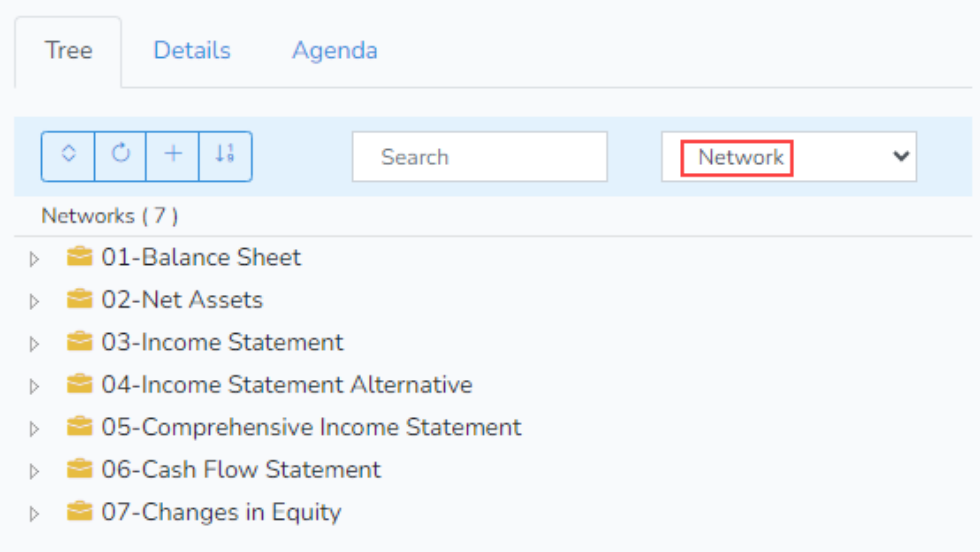
report model and report are not random assemblies of artifacts. There are patterns and patterns can be usefully employed.

Technical Oriented Objects

The following are the technical oriented assemblies of the report. Two of the technical oriented terms "network" and "hypercube" are explicitly defined by the XBRL technical specifications. The third term "component" is implied by XBRL.

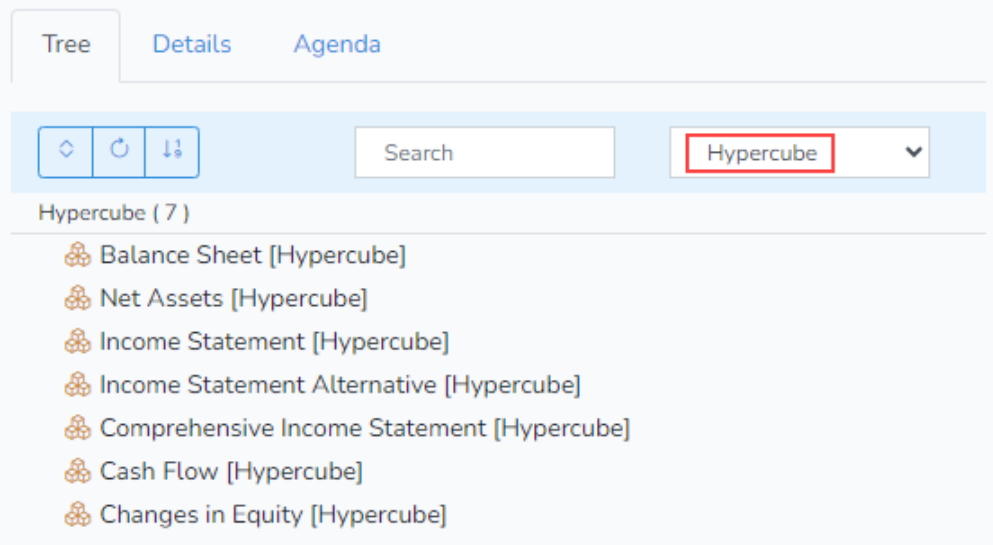
Networks

The following are the 7 XBRL networks that contain information provided within the report and report model:



Hypercubes

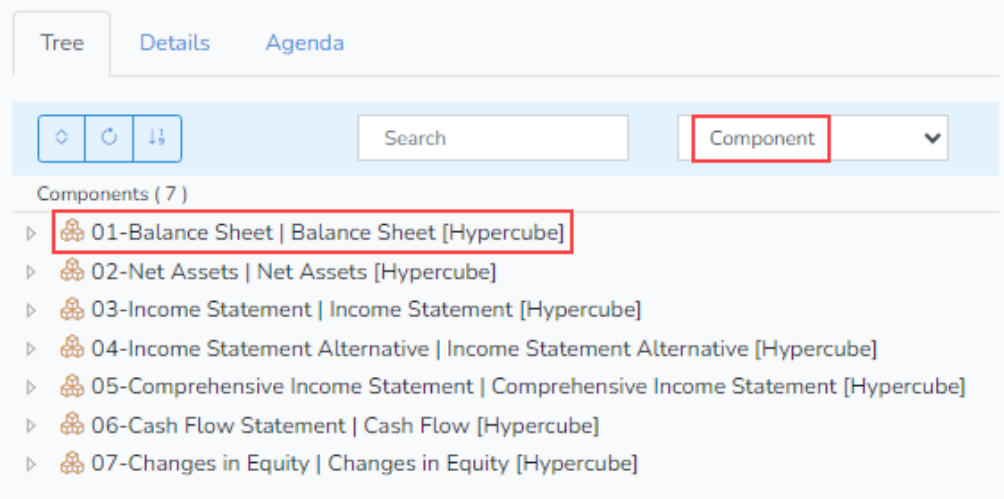
The following are the 7 XBRL hypercubes that contain information provided within the report and report model:



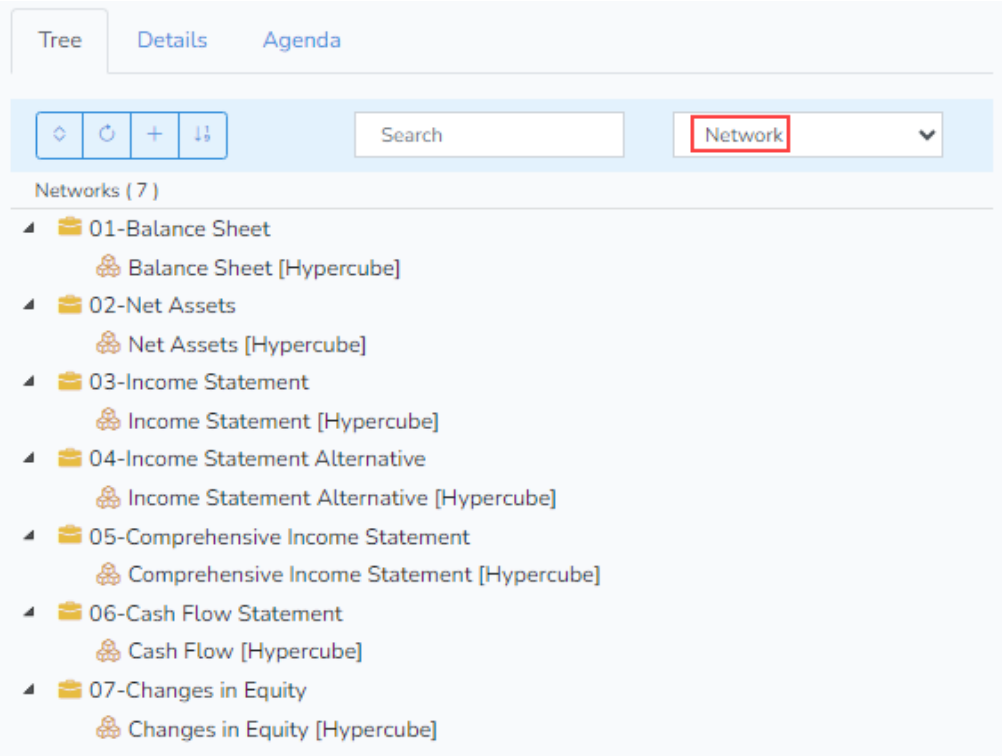
Note that there are the same number of hypercubes as there are networks because there is exactly one hypercube represented within each network.

Components

The following are the 7 XBRL hypercubes (assemblies) that contain information provided within the report and report model:



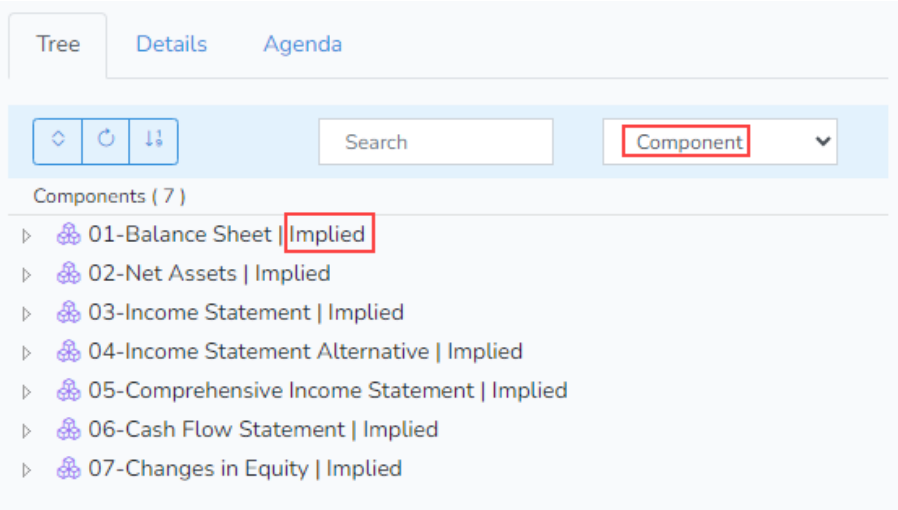
In the network view, if you expand each of the networks you see both the network and the hypercube:



While the notion of a "component" is not explicitly defined by XBRL, it is implied. A component is a combination of a network plus a hypercube within the network. The notion of a component is necessary because (a) any number of hypercubes could appear within a network and (b)

there is no guarantee that a hypercube used in one network cannot also be used within some other network. The notion of a component is used to uniquely identify an assembly there the hypercubes have exactly the same name. (i.e. both a network and a hypercube are required to explicitly define a unique assembly when two assemblies use a hypercube of exactly the same name.

Further, it may be the case that information is not explicitly defined within a hypercube. In this case, an implied hypercube is assumed that has no dimensional information. For example:



Note that the creator of a report can impact the presentation of an assembly of logical information by putting that assembly in whatever network they desire or whatever hypercube they may desire. However, where the assembly is provided does not impact the logic of that information.

Elements

What goes into networks and hypercubes is elements defined within XBRL. An element has the following attributes XML Schema attributes which have been supplemented by additional attributes added by XBRL and indicated by the "xbrli" namespace:

- id
- name
- type
- substitutionGroup
- abstract
- nillable
- xbrli:periodType
- xbrli:balance

The values of the attributes shown above determine the logical nature of the XBRL technical artifact. The following table helps one understand the relationship between the technical XBRL elements (which are XML Schema elements with two XBRL attributes added) and the logical report elements which the technical artifacts are used to represent).

	id	name	substitutionGroup	type	abstract	nillable	xbrli:periodType	xbrli:balance
Network	Per XLink specification							
Hypercube	namespace prefix + "_" + name	per XML Schema	xbrldt:hypercubeItem	xbrli:stringItemType	true	true	duration	
Dimension	namespace prefix + "_" + name	per XML Schema	xbrldt:dimensionItem	xbrli:stringItemType	true	true	duration	
Member	namespace prefix + "_" + name	per XML Schema	xbrli:item	nonnum:domainItemType	true	true	duration	
LineItems	namespace prefix + "_" + name	per XML Schema, includes "LineItem"	xbrli:item	xbrli:stringItemType	true	true	duration	
Abstract	namespace prefix + "_" + name	per XML Schema	xbrli:item	xbrli:stringItemType	true	true	duration	
Concept	namespace prefix + "_" + name	per XML Schema	xbrli:item	Any valid XBRL data type.	false	true	"duration", "instant"	"debit", "credit", or no value

Note that networks are specified using XLink and are extended links, not XML Schema elements. For more information, see the XBRL technical specification.

Alternative Technical Representation Approaches

When the creator of a report model represents information about that report model within the XBRL technical syntax they make certain choices. One of these choices is whether to use explicit hypercubes. Sometimes there is no choice; a report model creator MUST use an explicit hypercube when noncore dimensions are necessary to represent information effectively.

Whether the report model creator chooses to use a unique hypercube explicitly defined, choses to use the same one hypercube to represent each and every information assembly, or chooses not to use a hypercube in the report model representation at all; there is no impact on the meaning of the information being. All three report model creation approaches verify as one would expect:

#	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	✔

To understand logical objects, we will first show three different technical approaches to representing an XBRL-based report model and report to (a) highlight the different technical representation approaches and then to (b) show that the logic of what is represented logically is the same in all three different technical approaches.

An obvious question might be: "Why are three different technical approaches to representing the same information allowed." That is a very good question. That answer should be framed per a conscious understanding of the advantages and disadvantages of each of the three different approaches.

To become conscious of the differences, I would suggest *Essentials of XBRL-based Digital Financial Reporting*[2].

[2] *Essentials of XBRL-based Digital Financial Reporting (Platinum),* http://www.xbrlsite.com/seattlemethod/platinum/EssentialsOfXBRL_PLATINUM.pdf

Unique Explicit Hypercubes (i.e. every hypercube is unique; each logical fragment represented within hypercube)

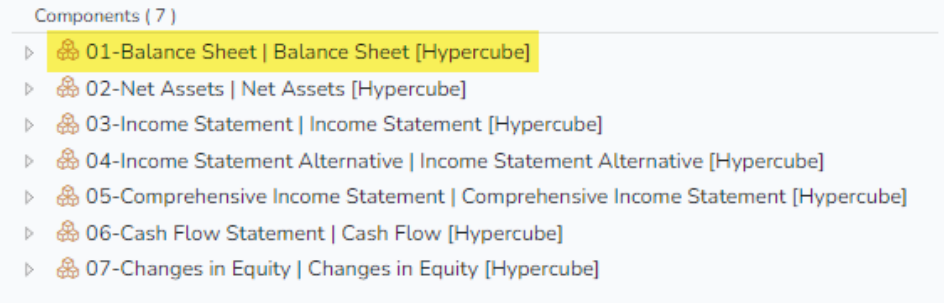
https://dev.auditchain.finance/storage/c5344192-b624-487d-8c22-11e42ba5963f/f1cc6cd2/instance.xml

https://auditchain.infura-ipfs.io/ipfs/QmPxuHQ9d9WvyQd5dkmst1cJPg1Zp1zAGJGYxs2dXsfCuP/

Verification rules)

['http://xbrlsite.com/seattlemethod/cm/model-structure-rules-strict-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/disclosure-mechanics/dm.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/fac/REFERENCE/fac.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/reporting-checklist/dr-rules-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/type-subtype/typeSubtype-rules-BEST-def.xml']

As you can see there are seven components; each with a unique network name (required by XBRL) and unique hypercube name (choice of report model creator or reporting scheme creator):



Component: (Network and Hypercube)	
Network	02-Net Assets (http://luca.auditchain.finance/report/role/NetAssets)
Hypercube	Net Assets [Hypercube]

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2023-12-31	2022-12-31
Net Assets [Roll Up]		
Assets	$ 0	$ 0
Liabilities	0	0
Net Assets	$ 0	$ 0

No Hypercubes (i.e. all hypercubes are implied)

https://dev.auditchain.finance/storage/c5344192-b624-487d-8c22-11e42ba5963f/bc5a8545/instance.xml

https://auditchain.infura-ipfs.io/ipfs/QmTHbW23nN1qjsHmXrQdVxXaPsA2iFHA7LmH2eiZJPSAup/

Verification rules (note that a different set of disclosure mechanics rules is used, a set that does not require the existence of the explicit hypercubes)

['http://xbrlsite.com/seattlemethod/cm/model-structure-rules-strict-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/disclosure-mechanics2/dm.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/fac/REFERENCE/fac.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/reporting-checklist/dr-rules-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/type-subtype/typeSubtype-rules-BEST-def.xml']

As you can see there are seven components; each with a unique network name (required by XBRL) and no explicit hypercube (choice of report model creator or reporting scheme creator); and so the hypercube is implied by the software application:

Component: (Network and Hypercube)	
Network	02-Net Assets (http://luca.auditchain.finance/report/role/NetAssets)
Hypercube	Implied

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2023-12-31	2022-12-31
Assets	$ 0	$ 0
Liabilities	0	0
Net Assets	$ 0	$ 0

Note that the naming of the hypercubes has no impact on the logic of the information being represented; only the technical artifacts that are used. But this can impact the extraction of information from an XBRL-based report.

Same Hypercubes (i.e. all assemblies use the same physical hypercube)

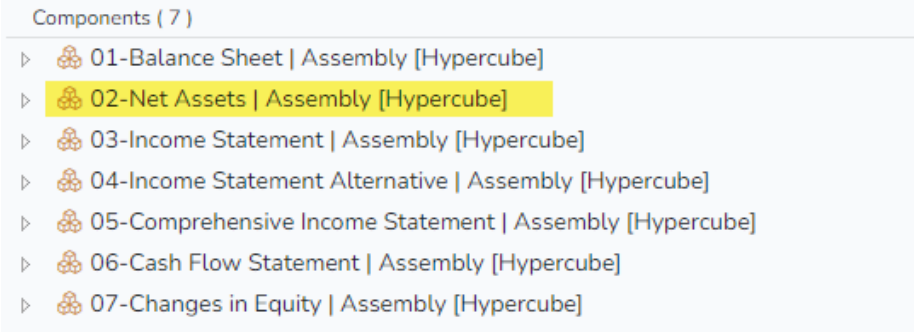
https://dev.auditchain.finance/storage/c5344192-b624-487d-8c22-11e42ba5963f/dcdf19ad/instance.xml

https://auditchain.infura-ipfs.io/ipfs/QmW4jJEgFyhcqe6LvruzEYNnzummxN4868V1SGYnehUwy5/

Verification rules (note that a different set of disclosure mechanics rules is used, a set that does not require the existence of the explicit hypercubes)

['http://xbrlsite.com/seattlemethod/cm/model-structure-rules-strict-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/disclosure-mechanics2/dm.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/fac/REFERENCE/fac.xsd','http://www.xbrlsite.com/seattlemethod/golden/common3/reporting-checklist/dr-rules-def.xml','http://www.xbrlsite.com/seattlemethod/golden/common3/type-subtype/typeSubtype-rules-BEST-def.xml']

As you can see there are seven components; each with a unique network name (required by XBRL) and each information fragment uses the same hypercube to represent information (choice of report model creator or reporting scheme creator):



Component: (Network and Hypercube)	
Network	02-Net Assets (http://luca.auditchain.finance/report/role/NetAssets)
Hypercube	Assembly [Hypercube]

Reporting Entity [Aspect]	GH259400TOMPUOLS65II | http://standards.iso.org/iso/17442
Unit [Aspect]	iso4217:USD

Concept [Aspect]	Period [Aspect]	
	2023-12-31	2022-12-31
Net Assets [Roll Up]		
Assets	$ 0	$ 0
Liabilities	0	0
Net Assets	$ 0	$ 0

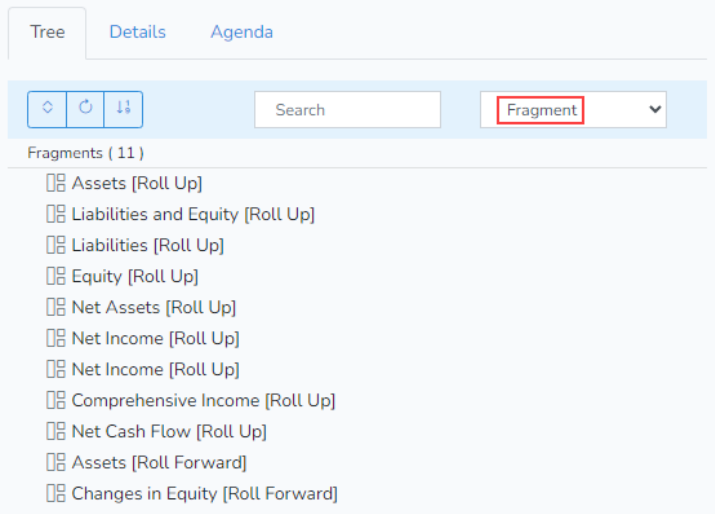
Again, the names of the hypercubes has no impact on the logic of the information being represented; but it does impact the extraction of information from the report.

Logical Oriented Objects

The following are the logical oriented assemblies of the report model and report. These logical artifacts are not defined by the XBRL technical specification.

Fragments (informal)

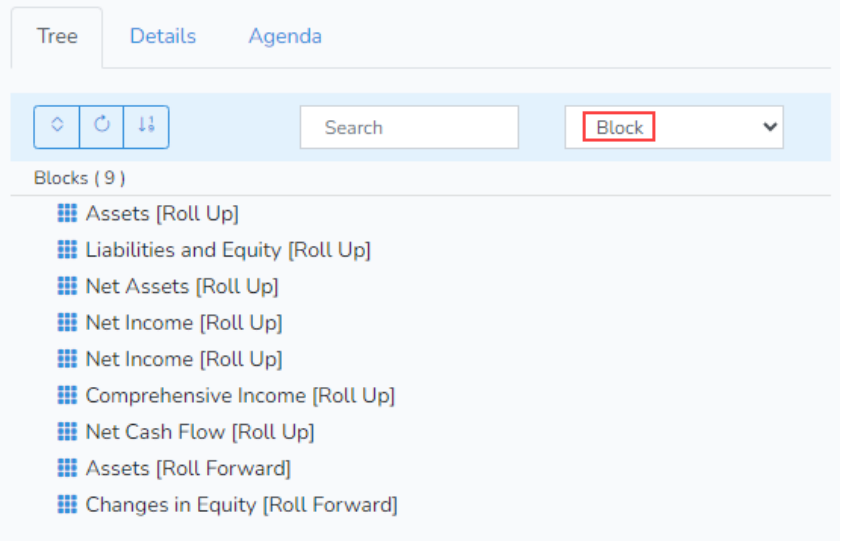
The following are the 11 logical fragments (assemblies) that contain information provided within the report and report model.



The reason there are 11 logical fragments is that two nested roll ups exist and each of the nested roll ups is considered an individual information fragment along with the combined grand total roll up.

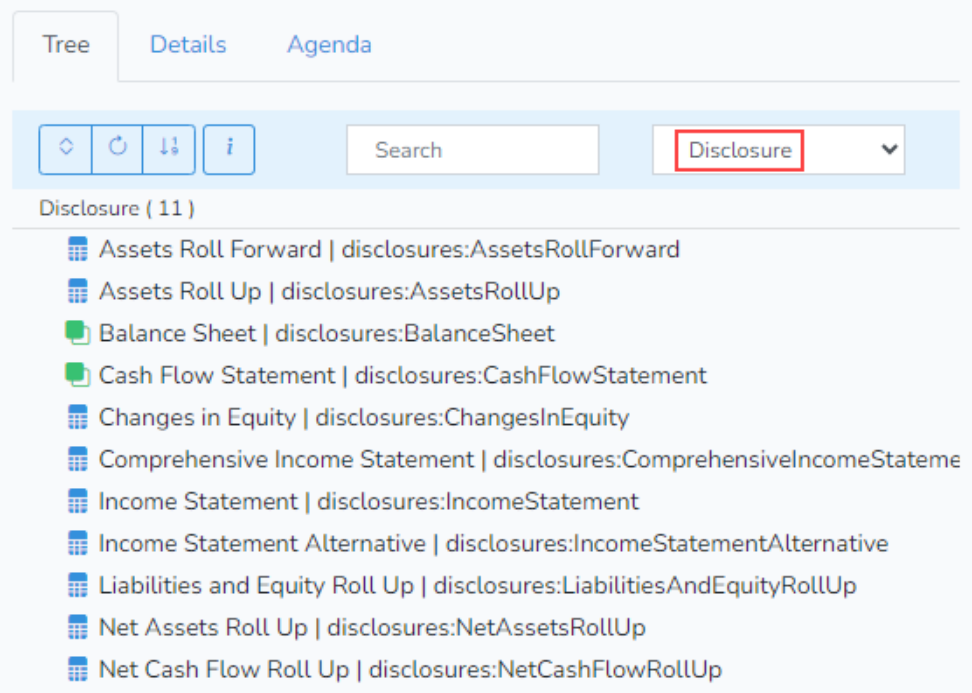
Blocks

The following are the 9 logical blocks (assemblies) that contain information provided within the report and report model.



Disclosures

The following are the 11 logical disclosures (assemblies) that contain information provided within the report and report model.



There are two more Disclosures than Blocks because the Balance Sheet Disclosure and the Cash Flow Statement Disclosure are combinations of multiple disclosures. The Balance Sheet Disclosure is a combination of the Assets Roll Up and Liabilities and Equity Roll Up Disclosures. The Cash Flow Statement is a combination of the Net Cash Flow Roll up and the Assets Roll Forward.

Infon (informal)

An infon is a useful unit of information. An entire report could be an infon, as could the set of any of the formal or informal logical oriented objects of a report.

Knowledge Brick (informal)

A knowledge brick is equivalent to or an alias of the term infon.

Topics

At times financial reporting schemes can be quite large, containing a very large number of disclosures. The logical object "topic" is a helpful tool for organizing disclosures[3].

[3] XBRL definition relations which define topics, http://www.xbrlsite.com/seattlemethod/golden/common3/disclosures-topics/topics-disclosures-definition.xml

-	Arcrole	Order
Definition View		
Disclosures		
Disclosure		0
Topic (Disclosure topic)		0
Face Statements	http://www.xbrl.org/2003/arcrole/general-special	1
Statement of Financial Position	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	1
Balance Sheet	http://www.xbrl.org/2003/arcrole/general-special	1
Balance Sheet, Classified	http://www.xbrl.org/2003/arcrole/general-special	1.1
Assets Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	1
Liabilities and Equity Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	2
Balance Sheet, Order of Liquidity	http://www.xbrl.org/2003/arcrole/general-special	1.2
Assets Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	1
Liabilities and Equity Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	2
Balance Sheet, Liquidition Basis	http://www.xbrl.org/2003/arcrole/general-special	1.3
Net Assets Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	1
Statement of Operations	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	2
Income Statement	http://www.xbrl.org/2003/arcrole/general-special	1
Income Statement Alternative	http://www.xbrl.org/2003/arcrole/general-special	2
Comprehensive Income Statement	http://www.xbrl.org/2003/arcrole/general-special	3
Statement of Cash Flow	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	3
Cash Flow Statement	http://www.xbrl.org/2003/arcrole/general-special	1
Net Cash Flow Roll Up	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	1
Assets Roll Forward	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	2
Statement of Changes in Equity	http://www.xbrlsite.com/seattlemethod/conceptual-model/arcrole/has-part	4
Changes in Equity	http://www.xbrl.org/2003/arcrole/general-special	1
Nature of Economic Entity	http://www.xbrl.org/2003/arcrole/general-special	2
Basis of Reporting	http://www.xbrl.org/2003/arcrole/general-special	3
Summary of Significant Accounting Policies	http://www.xbrl.org/2003/arcrole/general-special	4
Financial Accounts	http://www.xbrl.org/2003/arcrole/general-special	5
Broad Transactions	http://www.xbrl.org/2003/arcrole/general-special	6
Document and Entity Information	http://www.xbrl.org/2003/arcrole/general-special	7

Logical Twin

A "logical twin[4]" is an idea. It is a useful idea when the idea works reliably, predictably, repeatedly, robustly. I have used the term "professional knowledge graph[5]" in the past.

The following is a description of a logical twin (a.k.a. logical digital twin) for the purpose of formally describing the notion of a logical twin. The purpose of this description of specification is to create a reliable, practical, useful tool for professional accountants, financial analysts, financial regulators, and other such stakeholders. This specification will be driven by goals and objectives and capabilities related to satisfying those goals/objectives.

Characteristics of Logical Twin (Brainstorming)

The following is a summary of the characteristics of a logical twin:

[4] *Logical Digital Twin of Financial Reports*, http://www.xbrlsite.com/mastering/Part02_Chapter05.A0_LogicalDigitalTwin.pdf

[5] *Professional Knowledge Graph*, https://digitalfinancialreporting.blogspot.com/2023/12/professional-knowledge-graphs.html

1. A logical twin is a "real world model" (a model of something real from the real world) created mainly by humans, but perhaps with the assistance of a machine (supervised machine learning).
2. A logical twin contains definitions, information, and logic. (What do you mean by "definition", "information", and "logic"? Please explain each.)
3. A logical twin is complete, precise, and consistent (very high quality).
4. A logical twin is safe and controllable.
5. An XBRL-based report CAN BE a logical twin, but not all XBRL-based reports would be considered logical twins; for example, if the quality is low, there are lots of errors, the report is incomplete, the report is imprecise, the report is inconsistent.
6. Not every logical twin is an XBRL-based report.
7. A "knowledge brick" (a.k.a. infon, fragment) is an informal term, more of a helpful metaphor to help explain how logical twins work.
8. A set of XBRL-based reports can be a logical twin. For example, a period comparison of an economic entity's primary financial statements for ten years can be a logical twin and would be considered a knowledge product.
9. A logical twin can be an information product, a knowledge product, or a decision product.
10. Not all information products, knowledge products, or decision products are logical twins.
11. An XBRL-based digital general purpose financial statement that is proven to be complete, precise, and consistent can be considered a logical twin, an information product, and can be a set of knowledge bricks.
12. A logical twin is an object.
13. A logical twin is always multidimensional. Like XBRL, a logical twin has three core dimensions that always exist: Concept Aspect, Reporting Entity Aspect, and Calendar Period Aspect. In addition, zero to many additional noncore dimensions may be added.

Formal Definition

A logical twin is a simplified representation of some real-world logical system or process represented in a global standard machine-readable and machine-understandable form or physical technical format/syntax. One approach to implementing a logical twin is the XBRL global standard (if XBRL is used correctly).

Examples

This section contributes to formally defining a logical twin by providing examples of what are considered logical twins. This section can also be considered a set of business use cases for a logical twin. The following are specific examples of logical twins:

- One XBRL-based digital general purpose financial statement is a logical twin.
- A set of XBRL-based digital general purpose financial statements is a logical twin such as a set of 10 financial statements that are used to perform an entity comparison for the purpose of preforming a variance analysis or a set of 10 financial statements that are used to perform a peer comparison for the purposes of benchmarking.
- An entire repository of XBRL-based reports such as the complete set of XBRL-based reports submitted to the Securities and Exchange Commission and put into their EDGAR system can be considered a logical twin if done correctly.

- An unleavened discounted cash flow financial model[6] represented using XBRL is a logical twin.
- A trial balance of accounts output from an accounting system and represented using XBRL is a logical twin.
- An accounting schedule or audit schedule that summarizes some quantifiable information from an accounting system such as an aged accounts receivable trial balance represented using XBRL is a logical twin[7].
- An adjusted working trial balance[8] used for an audit is a logical twin.
- A general semantic spreadsheet[9] is a logical twin.

Ultimately, a conformance suite[10] will be created which shows positive examples and negative examples of XBRL-based logical twins.

PROOF of Logical Twin

The PROOF[11] that I have created provides a specific example of a logical twin. While the PROOF might look somewhat like a financial report, the PROOF is not about the precise nature of the specific representation, it is about (a) the capabilities of the representation and (b) the specific financial reporting functionality used as examples to demonstrate those specific capabilities.

[6] Representing Unlevered Discounted Cash Flow Model Using XBRL, http://xbrl.squarespace.com/journal/2018/9/4/representing-unlevered-discounted-cash-flow-model-using-xbrl.html

[7] Semantic accounting and auditing working papers, https://digitalfinancialreporting.blogspot.com/2023/05/semantic-accounting-and-auditing.html

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

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

[10] Conformance Suite, http://xbrlsite.com/seattlemethod/platinum-testcases/index.xml

[11] PROOF, https://digitalfinancialreporting.blogspot.com/2023/12/proof.html

Logical Object Discovery

The following section describes the process of discovering or understanding the logical objects that are represented within an XBRL-based digital financial report.

The technical objects are easy to discover because the objects are explicitly named within an XBRL-based report model or report. Networks and hypercubes are explicitly instantiated or implied within a report model or report. Components are easily identified as they are simply a combination of a network and a hypercube.

But Fragments, Blocks, and Disclosures are assemblies that are constituted by software applications that are processing the XBRL-based digital financial reports that understand those useful logical objects.

Report Model and Report

The full report model and report is the set of all fragments, or blocks, or disclosures. You can think of fragments, blocks, and disclosures as a view of a report similar to how a relational database differentiates a "table" and a "view" of a table. The technical artifacts are like the tables of a relational database and the logical objects are things that can be viewed.

As was explained in the "Elements" section of the Technical Oriented Objects; logically, a report has exactly seven different types of report elements: Network, Hypercube, Dimension, Member, LineItems, Abstract, Concept. There are logical relationships between these types of report elements that are permitted or disallowed.

It is worth pointing out again that there are well established good practices and best practices for creating a report model. This graphic shows the allowed and disallowed relationships between report elements used to construct the report model:

		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

Abstract report elements are used consistently to create containers for information represented. The following shows examples of such abstract containers. These abstract containers have no impact on the actual logic of information represented within an XBRL-based report; but the abstract containers are very helpful in enabling a human to read the report model.

Example *without* abstract container:

Concept [Aspect]	Period [Aspect]	
Assets	2022-12-31	€ 1,000
Liabilities	2022-12-31	500
Equity	2022-12-31	500
Revenues	2022-01-01 | 2022-12-31	1,000
(Expenses)	2022-01-01 | 2022-12-31	(1,000)
Gains	2022-01-01 | 2022-12-31	1,000
(Losses)	2022-01-01 | 2022-12-31	(1,000)
Comprehensive Income	2022-01-01 | 2022-12-31	€ 0
Equity, Beginning Balance	2021-12-31	€ 0
Comprehensive Income	2022-01-01 | 2022-12-31	0
Investments by Owners	2022-01-01 | 2022-12-31	500
(Distributions to Owners)	2022-01-01 | 2022-12-31	0
Equity, Ending Balance	2022-12-31	€ 500

Example 1 *with* meaningless abstract container:

Concept [Aspect]	Period [Aspect]		
Block 1 [Abstract]			
Assets	2022-12-31	€	1,000
Liabilities	2022-12-31		500
Equity	2022-12-31		500
Block 2 [Abstract]			
Revenues	2022-01-01 | 2022-12-31		1,000
(Expenses)	2022-01-01 | 2022-12-31		(1,000)
Gains	2022-01-01 | 2022-12-31		1,000
(Losses)	2022-01-01 | 2022-12-31		(1,000)
Comprehensive Income	2022-01-01 | 2022-12-31	€	0
Block 3 [Abstract]			
Equity, Beginning Balance	2021-12-31	€	0
Comprehensive Income	2022-01-01 | 2022-12-31		0
Investments by Owners	2022-01-01 | 2022-12-31		500
(Distributions to Owners)	2022-01-01 | 2022-12-31		0
Equity, Ending Balance	2022-12-31	€	500

Example 2 *with meaningful* abstract container:

Concept [Aspect]	Period [Aspect]		
Balance Sheet [Abstract]			
Assets	2022-12-31	€	1,000
Liabilities	2022-12-31		500
Equity	2022-12-31		500
Comprehensive Income [Abstract]			
Revenues	2022-01-01 | 2022-12-31		1,000
(Expenses)	2022-01-01 | 2022-12-31		(1,000)
Gains	2022-01-01 | 2022-12-31		1,000
(Losses)	2022-01-01 | 2022-12-31		(1,000)
Comprehensive Income	2022-01-01 | 2022-12-31	€	0
Changes in Equity [Abstract]			
Equity, Beginning Balance	2021-12-31	€	0
Comprehensive Income	2022-01-01 | 2022-12-31		0
Investments by Owners	2022-01-01 | 2022-12-31		500
(Distributions to Owners)	2022-01-01 | 2022-12-31		0
Equity, Ending Balance	2022-12-31	€	500

Example 3 with meaningful abstract container and separate explicit hypercubes:

Below you see the same logical information as the two other examples. The difference is that teach logical information assembly is represented within a separate XBRL network:

Component: (Network and Hypercube)	
Network	Network 1 (http://luca.auditchain.finance/report/role/Network1)
Hypercube	Implied

Concept [Aspect]	Period [Aspect]	
	2022-12-31	2021-12-31
Balance Sheet [Abstract]		
Assets	1,000	
Liabilities	500	
Equity	500	0

Component: (Network and Hypercube)	
Network	Network 2 (http://luca.auditchain.finance/report/role/Network2)
Hypercube	Implied

Concept [Aspect]	Period [Aspect]
	2022-01-01 | 2022-12-31
Comprehensive Income [Abstract]	
Revenues	1,000
(Expenses)	(1,000)
Gains	1,000
(Losses)	(1,000)
Comprehensive Income	€ 0

Component: (Network and Hypercube)	
Network	Network 3 (http://luca.auditchain.finance/report/role/Network3)
Hypercube	Implied

Concept [Aspect]	Period [Aspect]
	2022-01-01 | 2022-12-31
Changes in Equity [Abstract]	
Equity, Beginning Balance	€ 0
Comprehensive Income	0
Investments by Owners	500
(Distributions to Owners)	0
Equity, Ending Balance	€ 500

A fourth example would be to provide an explicit hypercube for each of the above logical information assemblies.

Again, the point here is that (a) where assemblies of logical objects are represented has no real impact on the logic of what is being represented; and (b) it is only the ability of a human to read the information which is impacted.

The best practice is to include abstract container objects wherever possible to make reading the information just a bit easier. Consistency is a good thing.

Fragments Discovery

Fragments are somewhat useful logical artifacts so they will be included. The best way to understand fragments is to have a look at a few fragments.

The following Liabilities and Equity [Roll Up] is a logical fragment of a report:

Concept [Aspect]	Period [Aspect]	
	2023-12-31	2022-12-31
Liabilities and Equity [Roll Up]		
Liabilities [Roll Up]		
Current Liabilities	0	0
Noncurrent Liabilities	0	0
Liabilities	✓ 0	✓ 0
Equity [Roll Up]		
Equity Attributable To Controlling Interests	0	0
Equity Attributable to Noncontrolling Interests	0	0
Equity	✓ 0	✓ 0
Liabilities and Equity	✓ 0	✓ 0

The Liabilities [Roll Up], which is part of the above fragment, is also a fragment of the report:

Concept [Aspect]	Period [Aspect]	
	2023-12-31	2022-12-31
Liabilities [Roll Up]		
Current Liabilities	0	0
Noncurrent Liabilities	0	0
Liabilities	✓ 0	✓ 0

Likewise, the Equity [Roll Up] is a fragment of the report:

Concept [Aspect]	Period [Aspect]	
	2023-12-31	2022-12-31
Equity [Roll Up]		
Equity Attributable To Controlling Interests	0	0
Equity Attributable to Noncontrolling Interests	0	0
Equity	✓ 0	✓ 0

As was pointed out in the fragments subsection of the prior section, the report we are looking at has a total of 11 logical fragments within the full report we are working with.

Report fragments are discovered by simply reading the report model information and then associating reported facts with that fragment object. "Parsing" the full report into a set of fragments is trivial. Fragments is an informal artifact, not really defined.

Blocks Discovery

Blocks are different from fragments in that a block is a formal unit of information. The following table explains how to identify blocks of information[12]:

#	Information Model Pattern (Concept Arrangement Pattern)[1]	XBRL Calculation Relations Exist?	Specific XBRL Formula Pattern Exists?	Member Arrangement Pattern[2] Exits?	Specific Report Date Dimension Exists?	Specific Reporting Scenario Dimensions Exist?	Originally Stated Label Role[3] Exists in XBRL Presentation Relations?	Restated Label Role[4] Exists in XBRL Presentation Relations?	Period Start Label Role Exists in XBRL Presentation Relations?	Period End Label Role Exists in XBRL Presentation Relations?
1	**Set**	Never	Never	Optional	Never	Optional	Never	Never	Never	Never
2	**Roll Up**	Always	Never	Optional	Never	Optional	Never	Never	Never	Never
3	**Roll Forward**	Never	**Always**[5]	Optional	Never	Never	Never	Never	**Always**	**Always**
4	**Roll Forward Info**	Never	Never	Optional	Never	Never	Never	Never	**Always**	**Always**
5	**Adjustment**	Never	Never	Never	**Always**	Never	**Always**	**Always**	Never	Never
6	**Variance**	Optional	**Always**[6]	**Always**	Never	**Always**	Never	Never	Never	Never
7	**Text Block**	Never	Never	Optional	Never	Never	Never	Never	Never	Never
8	**Member Aggregation**	Optional	**Always**	**Always**[7]	Never	Never	Never	Never	Never	Never
9	**Arithmetic**	Never	**Always**[8]	Optional	Never	Optional	Never	Never	Never	Never

Each information block in an XBRL-based report can be discovered per the rules in the table above. Using the above information, the list of blocks provided in the prior section would be discovered.

Disclosures Discovery

Once a block of information has been discovered, that block can be further identified as being a representation of a specific disclosure. This is done using the disclosure mechanics rules[13]. The disclosure mechanics rules specify the essence of a disclosure; enough information to uniquely identify each and every disclosure that could appear in an XBRL-based financial report[14].

#	Type	Name	Rule Expression
1	disclosure	disclosures:AssetsRollForward **Added** • detections:1	*Assets Roll Forward* • disclosures:AssetsRollForward **requires:** • Hypercube common:CashFlowHypercube • Concept Arrangement Pattern cm:RollForward ◦ **with** common:Assets *1 instance:* *In network 06-Cash Flow Statement:* *common:CashFlowHypercube is presented* *Detected block Assets, Beginning Balance [RollForward] with common:Assets*
2	disclosure	disclosures:AssetsRollUp **Added** • detections:1	*Assets Roll Up* • disclosures:AssetsRollUp **requires:** • Hypercube common:BalanceSheetHypercube • Concept Arrangement Pattern cm:RollUp ◦ **with** common:Assets *1 instance:* *In network 01-Balance Sheet:* *common:BalanceSheetHypercube is presented* *Detected block Assets [RollUp] with common:Assets*

[12] *Information Model Identification*, http://www.xbrlsite.com/mastering/InformationModelIdentification.pdf

[13] *Disclosure Mechanics Rules*, http://www.xbrlsite.com/seattlemethod/golden/common3/disclosure-mechanics/dm.xsd

[14] *Disclosure mechanics rules verification*, https://auditchain.infura-ipfs.io/ipfs/QmPxuHQ9d9WvyQd5dkmst1cJPg1Zp1zAGJGYxs2dXsfCuP/disclosures.html

Another example of the disclosure discovery can be seen using the Pesseract version of disclosure mechanics rules. The results of the disclosure discovery generates the following result in Pesseract:

Primary Information

#	Disclosure	Category	Level	Pattern	Disclosure Fo...	Disclosure Co...	Applicable	Representation Concept [TEXT BLOCK]	Representation Concept DETAIL
1	[AssetsRollForward]	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Assets
2	[AssetsRollUp]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Assets
3	[BalanceSheet]	Unknown	UNKNOWN	Component	True	CONSISTENT	True	-	-
4	[CashFlowStatement]	Unknown	UNKNOWN	Component	True	CONSISTENT	True	-	-
5	[ChangesInEquity]	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Equity
6	[ComprehensiveIncome...	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Comprehensive Income
7	[IncomeStatement]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Income
8	[IncomeStatementAltern...	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Income
9	[LiabilitiesAndEquityRollUp]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Liabilities and Equity
10	[NetAssetsRollUp]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Assets
11	[NetCashFlowRollUp]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Cash Flow

Each disclosure mechanics rule can be viewed within the Pesseract application:

Rules | Line of Reasoning

This disclosure: disclosures:AssetsRollForward

- MUST be represented using the Hypercube/[Table] named: common:CashFlowHypercube
- MUST be represented as a **Level 4 Disclosure Detail** with the concept arrangement pattern: cm:RollForward
  - cm:RollForward REQUIRES beginning/ending balance: common:Assets

In addition, the line of reasoning is shown so that a software application user can understand why or why not a disclosure was discovered:

Rules | Line of Reasoning

####Disclosure mechanics validation explanation for disclosure: disclosures:AssetsRollForward####

Level 4 Disclosure Detail

Looking for blocks with concept arrangement pattern: RollForward
Looking for Hypercube/[Table]: common:CashFlowHypercube
*FOUND Hypercube/[Table]: common:CashFlowHypercube in network:
Looking for Concept: common:Assets
*FOUND Concept: common:Assets in network:
Concept located in network: 06-Cash Flow Statement

CONCLUSION

Disclosure found in report: True
Disclosure mechanics are CONSISTENT because the Level 4 Disclosure Detail concept was FOUND.

END of disclosure mechanics validation explanation for this disclosure

Implementations

The following implementations are intended to help explain the logical artifacts of an XBRL-based report. Not all of these software applications are perfect. However, all reports created using the *Seattle Method* are interoperable with each of these software applications:

XBRL Cloud Evidence Package

http://www.xbrlsite.com/seattlemethod/golden/proof/reference-implementation/evidence-package/

Pesseract Working Proof of Concept

https://photos.app.goo.gl/cWeZYaMBEbmSSm7v8

Auditchain Pacioli

https://auditchain.infura-ipfs.io/ipfs/QmVdn6akCxSxB7yKb94qTFkG46UY4sNQPVRyQ9eyVC5eLK/

Auditchain Luca

https://digitalfinancialreporting.blogspot.com/2024/01/getting-started-with-auditchain-luca.html

Arelle

https://arelle.org/arelle/

UBmatrix XPE 2.5

https://sourceforge.net/projects/ubmatrix-xbrl/files/UBmatrix%20Processing%20Engine%202.5/2.500/

Conclusions

The following is a summary of the conclusions reached which we hope you have reached as a result of reading through this information:

1. Report creators can control which network, which hypercube, hypercube naming, and even whether they explicitly provide a hypercube to represent a logical assembly of information provided within an XBRL-based financial report.
2. Report creators have far less control over the logic of the representation because they must follow the rules of math, the rules of logic, and financial reporting logic when representing a report.
3. How logical artifacts are modeled in a report model impacts how those report artifacts can be viewed and otherwise used.
4. Care should be used when representing a report model. A report model creator should be conscious of their choices and how those choices will impact a user of the report which you have modeled.

Further Reading

The following is additional helpful information. The documents are arranged in no particular order.

Essentials of XBRL-based Digital Financial Reporting[15]: This document helps the reader understand important issues related to using XBRL to create XBRL-based financial reports effectively.

Accounting Basics (Brainstorming)[16]: This document contains a lot of information about business events, the notion of classic transactions, ACTUS, etc.

Essence of Accounting[17]: Relooks at some fundamental and foundational idea about accounting and reporting from the perspective of "digital".

Rules of Thumb[18]: Best practices and good practices relating to representing financial report information using XBRL.

Business Report Model in SQL[19]: Graphic of logical model from Access database.

Problem Solving Systems[20]: Explains the components of a problem solving system.

Standards Based Logical Twin Terminology[21]: Terminology used.

---

[15] Charles Hoffman, CPA, *Essentials of XBRL-based Digital Financial Reporting (Platinum)*, http://www.xbrlsite.com/seattlemethod/platinum/EssentialsOfXBRL_PLATINUM.pdf

[16] Accounting Basics (Brainstorming), https://xbrlsite.azurewebsites.net/2022/library/AccountingBasics.pdf

[17] Charles Hoffman, *Essence of Accounting*, https://xbrlsite.azurewebsites.net/2020/Library/EssenceOfAccounting.pdf

[18] *Rules of Thumb*, http://www.xbrlsite.com/mastering/Part04_Chapter07.G4_RulesOfThumb.pdf

[19] *Business Report Model in SQL*, http://www.xbrlsite.com/seattlemethod/platinum/proof/ref/BusinessReportModelInSQL.jpg

[20] *Problem Solving Systems*, https://digitalfinancialreporting.blogspot.com/2024/03/problem-solving-systems.html

[21] *Standards Based Logical Twin Terminology*, https://digitalfinancialreporting.blogspot.com/2024/03/standards-based-logical-twin-terminology.html