Understanding and Appreciating the Capability to Recognize and Identify Blocks

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

Last Revised – October 27, 2018 (DRAFT)

> "It must be remembered that there is nothing more difficult to plan, more doubtful of success, nor more dangerous to manage than a new system. For the initiator has the enmity of all who would profit by the preservation of the old institution and merely lukewarm defenders in those who gain by the new ones." Niccolò Machiavelli[1]

Executive summary:

- The three primary structures that are commonly used by most to interact with an XBRL-based report is the Network, they Hypercube, and the Fact. A fourth structure is the Component.
- A Block is a useful notion created out of convenience that enables software to interact with the fragments of an XBRL-based financial report with increased effectiveness.
- Discovering the Blocks that make up an XBRL-based report is useful. Identifying the nature of each Block is also very helpful.
- The Block enables software engineers to create software that can automate tasks such as the reporting checklist.
- The Block plays a role in the perspective of a complete system by contributing to the chain of capabilities that is necessary to enable the creation of high-quality digital financial reports.
- These ideas have been perfected in XBRL-based financial reports submitted to regulators but these ideas benefit the broader area of accounting process automation.

[1] Good Reads, https://www.goodreads.com/quotes/88687-it-must-be-remembered-that-there-is-nothing-more-difficult

Copyright (full and complete release of copyright)

All content of this document is placed in the public domain. I hereby waive all claim of copyright in this work. This work may be used, altered or unaltered, in any manner by anyone without attribution or notice to me. To be clear, I am granting full permission to use any content in this work in any way you like. I fully and completely release all my rights to any copyright on this content. If you feel like distributing a copy of this work, you may do so without attribution or payment of any kind. All that said, attribution is appreciated should one feel so compelled. The copyrights of other works referenced by this document are established by the referenced work.

An XBRL-based report, such as a financial report, is composed of many fragments which make up the structure of the report. Each structural piece, or fragment, category has pros and cons depending on the nature of the structural piece.

The lowest level category of report fragment is the **Fact**. Facts provide individual pieces of information. As explained by the multidimensional model of an XBRL-based report[2], one Fact is differentiated from another Fact by the Characteristics of the Fact. Individual Facts is generally the most detailed level of interaction with the information that makes up a report.

The highest level category of a report fragment is the **Network**. Networks provide individual fragments of the technical structure of a report. The Network tends to be a technical oriented structure that are quite large and that have some business meaning but generally not enough precision in terms of meaning to be useful. Further, Networks often offer information in terms of the personal preference of how someone tends to want to organize information which could be well organize or poorly organized.

The second-highest level category of a report fragment is the **Hypercube** or often times called the [Table] in US GAAP or IFRS based financial reports. The Hypercube tends to be correlated to the Network on a one-to-one basis in the vast majority of US GAAP and IFRS based financial reports because each Network tends to contain only one explicitly defined Hypercube or one implied Hypercube (i.e. while no Hypercube is defined, all Concepts in the Network are grouped into one dynamically defined implied Hypercube). And so, the utility of the Hypercube tends to be comparable to the utility of the Network.

If there is more than one Hypercube in a Network, then that report fragment can be identified by combining the Network plus each Hypercube into what I refer to as a **Component**. But again, Components tend to have the same utility as a Network.

Enter the notion of what I call the **Block**. The Block is a fragment of a report that was created in order to make it possible to enable software applications to interact with the fragments of an XBRL-based report at a level that enables software engineers to create functionality that is useful to the professional accountants interacting with such reports.

This document explains what Blocks are and why they are so useful in solving two significant problems when working with an XBRL-based report: (2) working at a logical as contrast to a technical level and (2) identifying the fragment that you are working with.

[2] *Introduction to the Multidimensional Model for Professional Accountants*, http://xbrl.squarespace.com/journal/2016/3/18/introduction-to-the-multidimensional-model-for-professional.html

Understanding the Block

The notion of a Block[3] is one of the harder ideas to explain and understand in the conceptual model of a digital financial report[4]. Blocks are explained in detail in the document *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports* [5] and explained how they are used in the document *Guide to Building an Expert System for Creating Financial Reports* [6].

A Block is a unit of a report that was created in order to make interacting with and otherwise working with a report easier. Individual facts are many times too small working sets to be useful. Networks tend to be too large to work with. A Block is a useful unit of a report that makes doing certain things significantly easier.

Essentially, a Block is a fragment of a report that has the same *concept arrangement pattern*[7]. For example, there is a simple basic block:

Inventory Disclosure [Abstract]	Period [Axis]	
	2016-03-31	2015-03-31
Inventory Disclosure [Abstract]		
Raw materials and supplies	7,993,000	7,417,000
Work-in-progress	13,147,000	6,466,000
Finished goods	5,600,000	2,891,000
Inventories	26,740,000	16,774,000

The Block above has the concept arrangement pattern "roll up". I will explain more of the traits of a Block as we go along. Blocks are very consistent structures and makes working with an XBRL-based report at a logical level possible as contrast to working at the XBRL technical syntax level. Blocks make many, many things possible.

Note that then notion of blocks was inspired by MIT's Scratch application[8] and Blockly[9].

---

[3] YouTube.com, *Understanding Blocks*, https://www.youtube.com/watch?v=yI9yjD_T78I

[4] Charles Hoffman and Rene van Egmond, *Introduction to Conceptual Model of a Digital Financial Report*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part02_Chapter05.1_IntroductionToTheConceptualModelOfDigitalFinancialReport.pdf

[5] Charles Hoffman and Hamed Mousavi, *Putting the Expertise into an XBRL-based Knowledge Based System for Creating Financial Reports*, http://pesseract.azurewebsites.net/PuttingTheExpertiseIntoKnowledgeBasedSystem.pdf

[6] Charles Hoffman, *Guide to Building an Expert System for Creating Financial Reports*, http://xbrlsite.azurewebsites.net/2018/Library/GuideToBuildingAnExpertSystemForCreatingFinancialReports.pdf

[7] Charles Hoffman and Rene van Egmond, *Understanding Concept Arrangement Patterns, Member Arrangement Patterns, and Report Fragment Arrangement Patterns*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part02_Chapter05.7_UnderstandingConceptArrangementPatternsMemberArrangementPatterns.pdf

[8] MIT, Scratch, https://scratch.mit.edu/

A Block is a part of a Component whose line items all participates in the same *concept arrangement pattern*[10]. A roll up, roll forward, adjustment, and hierarchy are all types of concept arrangement patterns. Every XBRL-based public company financial report is essentially a set of Blocks. I estimate that there are about 754,430 blocks in the set of 5,734 public company reports that I analyzed. Of that total, 16% are roll ups, 5% are roll forwards, 24% are hierarchies, and 54% are text blocks[11].

If you do not understand what a Block is, I would encourage you to read the documents referenced thus far in this document. If you want some hands on experience with Blocks, download the Pesseract software application and work through the documentation related to Networks, Components and Blocks[12].

Examples Used in This Document

In the remainder of this document I am going to use examples from the document *General Ledger Trial Balance to External Financial Report*[13]. In that document you will see a rather small example, a larger example, and an XBRL-based report that was submitted to the SEC referred to. You can get to working commercial quality tools provided by XBRL Cloud for the examples I will reference.

Alternatively, you can download and install the Pesseract working proof of concept[14] and load the examples for yourself. I will try and provide direct references to examples as best that I can.

It may be the case that I reference XBRL-based financial reports submitted to the SEC and show those reports in the XBRL Cloud Viewer. You can get to pretty much any XBRL-based financial report submitted to the SEC from the XBRL Cloud EDGAR Dashboard[15]. Just press the blue button that says “Viewer” in the column marked “Interactive Reviewer”.

It can be hard to understand information if you cannot see things for yourself. At the same time, different software applications provide different functionality. Because of the differences in functionality, it can be hard to explain extremely detailed ideas. I will use the XBRL Cloud and Pesseract software tools.

---

[9] Blockly, http://xbrl.squarespace.com/journal/2014/7/14/blockly.html

[10] See page 11, http://www.xbrlsite.com/2015/Analysis/AnalysisSummary2014_PiecesOfReoprt.pdf#page=11

[11] I have a document that summarizes this information.

[12] Networks, Components, and Blocks, http://xbrlsite.azurewebsites.net/2018/Pesseract/12-NetworksComponentsAndBlocks.pdf

[13] General Ledger Trial Balance to External Financial Report, http://xbrlsite.azurewebsites.net/2018/RoboticFinance/TrialBalanceToReport.pdf

[14] Pesseract, http://pesseract.azurewebsites.net/

[15] EDGAR Dashboard, https://edgardashboard.xbrlcloud.com/edgar-dashboard/

Perspective

Why are Networks necessary? There are exactly two answers to that question: (1) because you WANT to represent a report fragment in separate Networks or (2) because you HAVE to represent a report fragment in a different Network.

So the "WANT to" is easy to understand. For example, you choose to put two fragments of information into two different Networks. Why would you do that? Because you can. Not much of a discussion is necessary here, this is simply a matter of preference. But what about the "HAVE to". Why would you have to construct two different report fragments within different Networks? The answer is conflicts.

Say, for example, that you wanted to represent the three computations[16] below within an XBRL-based report:

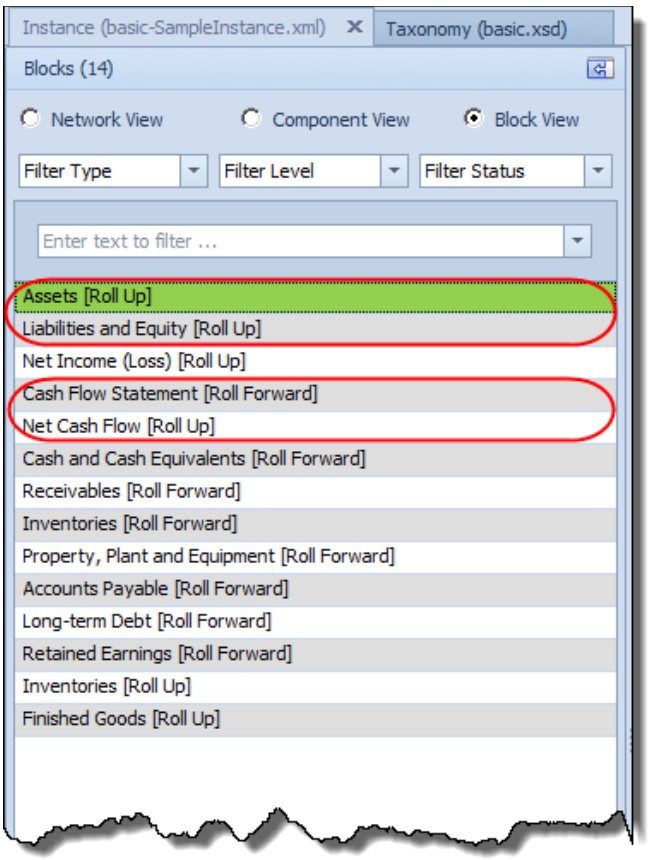
	2010	2009
TRADE AND OTHER RECEIVABLES		
Trade and Other Receivables, Net, by Component		
Trade Receivables, Net	8,790	6,431
Financing Lease Receivables, Net	2,498	1,263
Other Receivables, Net	1,305	1,096
Trade and Other Receivables, Net	12,593	8,790
Trade and Other Receivables, Net, by Net/Gross		
Trade and Other Receivables, Gross	18,280	13,472
Allowance for Doubtfull Accounts	-5,687	-4,682
Trade and Other Receivables, Net	12,593	8,790
Trade and Other Receivables, Net, by Current/Noncurrent		
Trade Receivables, Net, Current	6,340	5,701
Trade Receivables, Net, Noncurrent	6,253	3,089
Trade and Other Receivables, Net	12,593	8,790

To do that, you would need to use three different Networks. Why? Because there are three different sets of roll up computations and if you represented all three roll ups within one Network the three sets of computations would conflict with one another. And so, to avoid conflicts such as this (i.e. there are other types of conflicts) you use Networks.

[16] Business Use Case: Multiple Rollups, http://xbrlsite.azurewebsites.net/DigitalFinancialReporting/BusinessUseCases/2017-05-07/BUC06-MultipleRollUps/Index.html

The point here is to highlight the difference between a preference where you could do something and a requirement where you must do something. It is important to be able to distinguish between what is a preference and something that is required.

Next, the following might not be familiar to you. What you see is a list of the Blocks within a straight forward XBRL document[17]:



In that report there are 12 Networks, there are 12 Hypercubes, there are 12 Components because there is one Hypercube per Network, and there are 14 Blocks. Why are there 14 Blocks? The reason there are 14 Blocks is because there are two Components that contain two Blocks, all the other Components have only one Block.

In this representation the balance sheet is in one Network, but that Network contains two Blocks; “Assets [Roll Up]” and “Liabilities and Equity [Roll Up]”. And that makes sense; that is exactly what a balance sheet is: two roll ups and the totals for the roll ups are the same amount, the balance sheet balances.

[17] You can grab the ZIP archive with the XBRL instance and XBRL taxonomy here, http://xbrlsite.azurewebsites.net/2018/Presseract/Basic.zip

In this representation the cash flow statement is likewise in one Network, but again there are two Blocks: “Net Cash Flow [Roll Up]” and the “Cash Flow Statement [Roll Forward]”. That likewise makes sense, a cash flow statement is a roll up of the net cash flows from operating, financing, and investing activities to the total net cash flow. Then, the beginning balance is rolled forward to the ending balance using that total net cash flow and perhaps some other reconciling items.

So why couldn’t this report be represented so that there is a one-to-one correlation between Networks and Blocks? There is no reason that could not be done. How this information is represented is purely a matter of preference.

Where information occurs in the report is a matter of preference, habit, norms in many cases, best practices, and other such reasons. The meaning of the information contained in the report would not change if Blocks were represented in different Network combinations. The only imitation that you have is that you cannot create relations where you have physical conflicts between things like the different totals of receivables which we showed earlier in this section.

Disclosure Mechanics

Disclosure mechanics is the idea that each disclosure follows specific structural, mathematical, logical, accounting, and maybe even other rules.

So, for example, consider this inventory components disclosure:

Component: (Network and Table)	
Network	2101 - Disclosure - **Inventories Detail**
Table	Implied [Table]

Reporting Entity [Axis]	30810137d58f76b84afd http://standards.iso.org/iso/17442	
Unit [Axis]	USD	
	Period [Axis]	
Implied [Line Items]	2018-12-31	2017-12-31
Inventories [Roll Up]		
Finished Goods	600	700
Work in Progress	200	100
Raw Material	200	200
Inventories	1,000	1,000

Notice that this disclosure is a roll up. Would an inventory components disclosure ever be a roll forward? The answer is no. An inventory roll forward is a different disclosure. Here is what an inventory roll forward looks like:

Component: (Network and Table)	
Network	1230 - Disclosure - **Inventories**
Table	Implied [Table]

Reporting Entity [Axis]	30810137d58f76b84afd http://st
Unit [Axis]	USD
	Period [Axis]
Inventories [Roll Forward]	2018-01-01/2018-12-31
Inventories [Roll Forward]	
Inventories, Beginning Balance	1,000
Purchases of Inventory for Sale	2,000
Costs of Sales 2	(2,000)
Inventory Written Off	0
Inventories, Ending Balance	1,000

You may say, "Well, that looks like it has a total also." But that is not what is going on. Note the first line item "Inventories, Beginning Balance" and then the last line item "Inventories, Ending Balance".

A roll forward reconciles the same concept between two points in time. A roll up aggregates a concept, in this case, at a point in time[18].

So ask yourself a question: "How may roll ups have a total?" The answer is that 100% of roll ups have a total. And then ask yourself, "How many roll ups have to actually mathematically add up?" Well, clearly 100% of roll ups should add up mathematically.

You can ask yourself the same type of questions about roll forwards. These commonalities between roll ups and roll forwards are patterns. Software engineers love patterns. Why? Because patterns allow them to absorb some of the complexity of a task, have the software perform that task, and then provide useful functionality within software applications.

OK, so we say we identified a pattern for inventory components roll up. How can we check to see if that pattern holds? Empirical evidence. Look at the inventory components disclosure of financial reports of public companies. Well, I did exactly that. Here are the results[19]:

	July 3, 2016	June 28, 2015
	(in thousands)	
Finished goods	$ 44,264	$ 43,254
Work-in-process	24,573	16,020
Raw materials	34,491	33,889
	$ 103,328	$ 93,163

[18] To understand the difference between a roll up and a roll forward, see the document *Leveraging the Theoretical and Mathematical Underpinnings of a Financial Report*, page 4, Legers and Journals, Stocks and Flows, http://xbrlsite.azurewebsites.net/2018/Library/TheoreticalAndMathematicalUnderpinningsOfFinancialReport.pdf

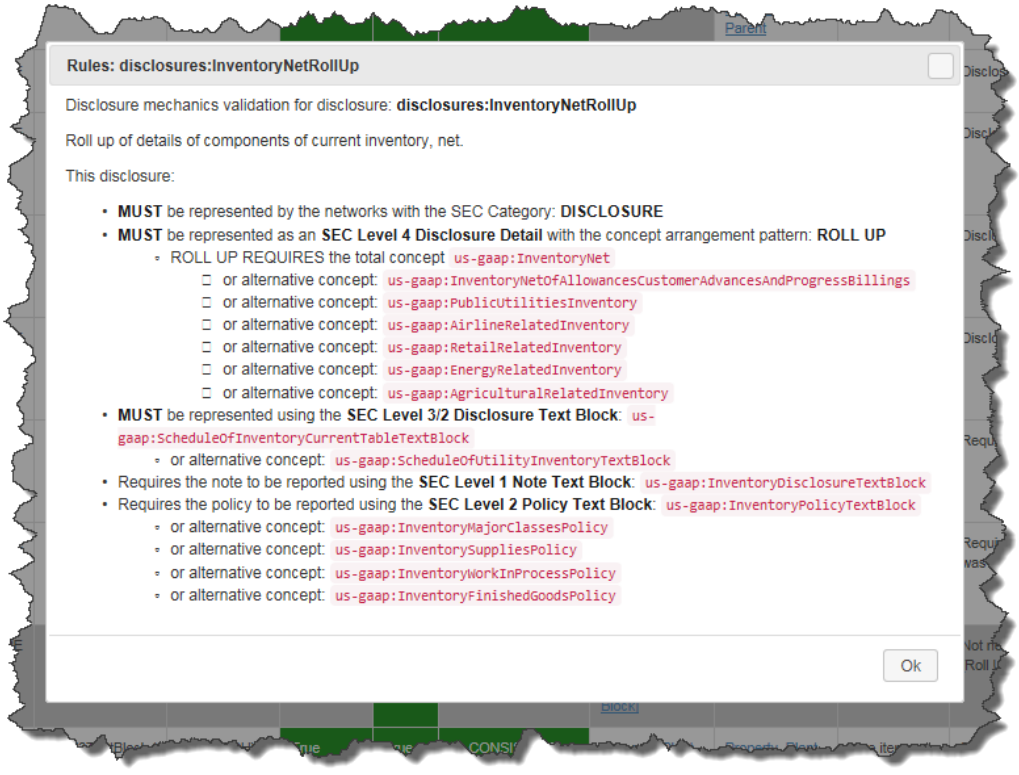
[19] Inventory Components Roll Up, http://xbrlsite-app.azurewebsites.net/DisclosureBestPractices/DisclosureBestPractices.aspx?DisclosureName=InventoryNetRollUp

What about other patterns? Well, OK; so if there is an inventory components roll up disclosure[20]; I know that there would also be an inventory note[21]. You would also expect that there would be an inventory policy[22].

Once you start noticing patterns you start noticing more patterns. You start documenting relationships[23]. You begin to automate the testing process using that information and you can analyze 100% of the population:

DisclosureFound	DisclosureConsistent	RepresentationConcept_TextBlock	RepresentationConcept_Detail	Count	Percent
FALSE	CONSISTENT	NOT-FOUND	NOT-FOUND	3,612	56%
TRUE	CONSISTENT	us-gaap:ScheduleOfInventoryCurrentTableTextBlock	us-gaap:InventoryNet	1,721	27%
TRUE	INCONSISTENT	NOT-FOUND	us-gaap:InventoryNet	1,061	16%
TRUE	INCONSISTENT	us-gaap:ScheduleOfInventoryCurrentTableTextBlock	NOT-FOUND	46	1%
TRUE	CONSISTENT	us-gaap:ScheduleOfUtilityInventoryTextBlock	us-gaap:InventoryNet	19	0%
TRUE	INCONSISTENT	us-gaap:ScheduleOfUtilityInventoryTextBlock	NOT-FOUND	7	0%
				6,466	100%
			Consistent with expectation	5,352	83%
			Inconsistent with expectation	1,114	17%
			Total	6,466	100%

Then you document these observations in pseudo code, for example:



[20] Inventory disclosure, http://www.xbrlsite.com/2018/10K/Notes/Compare_813.html

[21] Inventory note, http://www.xbrlsite.com/2018/10K/Notes/Compare_91.html

[22] Inventory policy, http://www.xbrlsite.com/2018/10K/Notes/Compare_347.html

[23] Understanding Disclosure Mechanics, http://xbrlsite.azurewebsites.net/2016/Analysis/UnderstandingDisclosureMechanics.pdf

Then, you figure out a scheme to represent those rules in machine-readable form. I tried numerous alternatives and settled on the XBRL syntax, specifically using XBRL definition relations[24].

I repeated this process for approximately 65 disclosures, summarizing the information in a document called *Disclosure Best Practices*[25]. I created rules for each of the approximately 65 rules. Then, I organized the individual rules linkbases with an XBRL taxonomy schema[26].

I then talked to two different software engineers about what I had put together and which I knew worked because I could get most of this to work using some Microsoft Access database applications that I created. But clearly, what I had was not commercial quality.

But, I convinced XBRL Cloud to implement these disclosure mechanics rules. I could run this for individual companies, for example here is the Microsoft 10-K checked against the approximately 65 disclosure mechanics rules[27].

I implemented the same thing in Pesseract[28], comparing the XBRL Cloud results with the Pesseract results[29] to make sure the implementations were consistent.

Note that while these rules are specifically created for US GAAP, the framework is independent of any profile[30].

IFRS works also, but I have not created a lot of rules yet. I created another example for the XASB reporting schema[31] that I create and maintain for testing. There is a specific thing that I will get back to with regard to this.

[24] XBRL definition relations rules for inventory components roll up, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/disclosure-mechanics/517-rules-def.xml

[25] Disclosure Best Practices, http://www.xbrlsite.com/site1/2017/Prototypes/DisclosureAnalysis/DisclosureBestPractices.pdf

[26] XBRL taxonomy schema for 65 rules, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/disclosure-mechanics/Disclosures_BASE2.xsd

[27] Microsoft Disclosure Mechanics Validation Results, http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html

[28] Microsoft Results in Pesseract, video, https://www.youtube.com/watch?v=V9bElBR7iFg

[29] Microsoft Results in Pesseract results screenshot, http://xbrlsite.azurewebsites.net/2017/Prototypes/DisclosureMechanicsExample/DisclosureMechanicsSummary.jpg

[30] XBRL-based Digital Financial Reporting Profiles and General Business Reporting Profile, http://xbrlsite.azurewebsites.net/2018/Library/Profiles-2018-10-22.pdf

[31] XASB reporting schema, Disclosure Mechanics Results, http://xbrlsite.azurewebsites.net/2017/Prototypes/XASB/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html

Reporting Checklist

I organized the disclosure mechanics rules with another set of helpful rules which I call the reporting checklist. The disclosure checklist rules were likewise represented using XBRL definition relations[32].

#	Disclosure	Category	Level	Pattern	Applicable	Found	Disclosure Consistent	Representation Concept [TEXT BLOCK]	Representation Concept [DETAIL]	Checklist Category	Reason
1	Entity Information		Level4Detail	HIERARCHY	True	True	CONSISTENT	NOT-EXPECTED	Economic Entity Name	Required disclosure	Disclosure always required
2	Document Information		Level4Detail	HIERARCHY	True	True	CONSISTENT	NOT-EXPECTED	Document Title	Required disclosure	Disclosure always required
3	Financial Highlights		Level3TextBlock / Level4Detail	HIERARCHY	True	True	CONSISTENT	Financial Highlights [HTML]	Revenues, Net	Required disclosure	Disclosure always required
4	Balance Sheet		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]
5	Assets [Roll Up]		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Assets	Part of disclosure	Disclosure always required
6	Liabilities and Equity [Roll Up]		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Liabilities and Equity	Part of disclosure	Disclosure always required
7	Income Statement		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Net Income (Loss)	Required disclosure	Disclosure always required
8	Cash Flow Statement, Direct Method		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Cash Flow, Net	Required disclosure	Disclosure always required
9	Statement of Changes in Equity		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Equity	Required disclosure	Disclosure always required
10	Significant Accounting Policies		Level1TextBlock	LEVEL 1 TEXT BLOCK	True	True	CONSISTENT	Significant Accounting Policies [Note]	NOT-EXPECTED	Required disclosure	Disclosure always required
11	Basis of Reporting		Level1TextBlock	LEVEL 1 TEXT BLOCK	True	True	CONSISTENT	Overall Financial Report Presentation and Display [HTML]	NOT-EXPECTED	Required disclosure	Disclosure always required
12	Nature of Operations		Level1TextBlock	LEVEL 1 TEXT BLOCK	True	True	CONSISTENT	Nature of Business [HTML]	NOT-EXPECTED	Required disclosure	Disclosure always required
13	Cash and Cash Equivalents Components		Level3TextBlock / Level4Detail	ROLL UP	True	True	CONSISTENT	Cash and Cash Equivalents Components [Schedule]	Cash and Cash Equivalents	Line item exists, then disclosure required	Required because line item gaap:CashAndCashEquivalents was reported
14	Reconciliation of Cash Summary		Level3TextBlock / Level4Detail	ROLL UP	True	True	CONSISTENT	Reconciliation of to Cash Flow Statement, Summary [Schedule]	Cash and Cash Equivalents, per Cash Flow Statement	Line item exists, then disclosure required	Required because line item gaap:CashAndCashEquivalents was reported
15	Reconciling Item of Cash and Cash Equivalents		Level3TextBlock / Level4Detail	HIERARCHY	True	True	CONSISTENT	Reconciliation of to Cash Flow Statement, Detail [Schedule]	Reconciling Item, Amount	Disclosure 'Reconciliation of Cash Summary' exists, then disclosure required	Required because disclosure 'Reconciliation of Cash Summary' was found and is consistent.
16	Receivables Details, Current and Noncurrent		Level3TextBlock / Level4Detail	ROLL UP	True	True	CONSISTENT	Receivables, Current and Noncurrent [Schedule]	Receivables, Net	Line item exists, then disclosure required	Required because line item gaap:ReceivablesNet was reported
17	Receivables Details, Gross, Net		Level3TextBlock / Level4Detail	ROLL UP	True	True	CONSISTENT	Receivables, Net and Gross [Schedule]	Receivables, Net	Line item exists, then	Required because line item gaap:Rec

What the reporting checklist does is organize the disclosures into a checklist which articulates whether a disclosure is always required, required if some line item is reported, required if some other disclosure is required, etc. The disclosures in the reporting checklist are the disclosures for which the disclosure mechanics rules are provided. So, the reporting checklist and the disclosure mechanics work together.

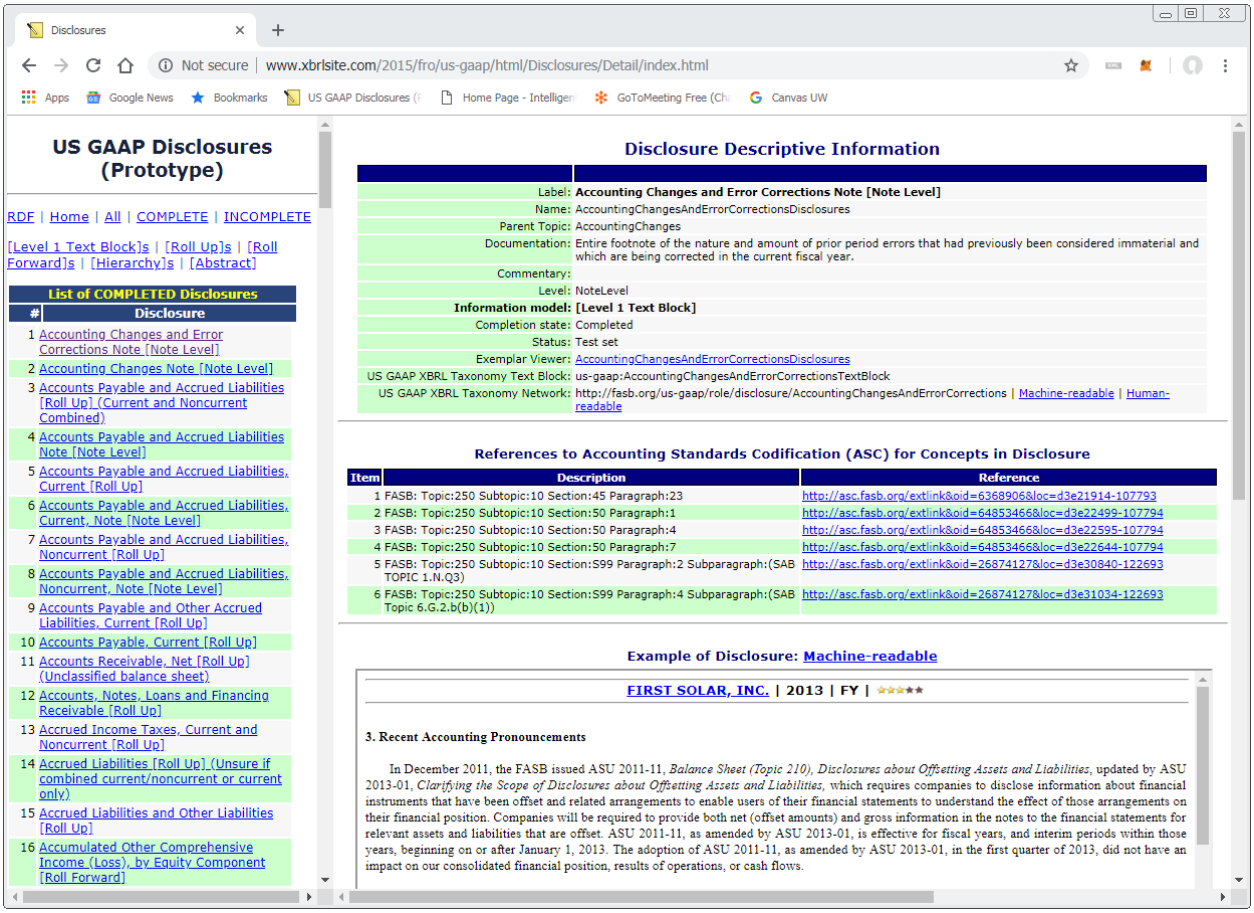
Disclosures

Key to making all of this work is inventorying 100% of disclosures which will be provided within a financial report and assigning each of these disclosures a unique identifier. Below is that inventory of disclosures, each of which has been uniquely identified[33]. Currently there are 959 disclosures within this library. Additional libraries exist for IFRS and for the XASB testing

[32] Reporting checklist, XBRL definition relations rules, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/reporting-checklist/ReportingChecklist-us-gaap-strict-rules-def.xml

[33] Disclosures, http://www.xbrlsite.com/2015/fro/us-gaap/html/Disclosures/Detail/index.html

reporting scheme which helps make sure that this functionality works for any reporting scheme which has a profile which is supported.



Again, these disclosures and all related information are provided in the form of an XBRL taxonomy schema[34].

Topics

Less important but still useful is the notion of topics. Topics are used to organize disclosures into a hierarchy[35]. Because a flat list of 959 disclosures can be overwhelming to work with, an XBRL taxonomy schema was created to define topics[36]. Topics enable disclosures to be organized within sets. Topics exist for each reporting scheme or profile supported.

[34] Disclosures XBRL taxonomy schema, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/Disclosures/disclosures.xsd

[35] PWC, *FASB accounting standards codification quick reference guide*, https://www.pwc.com/us/en/cfodirect/assets/pdf/accounting-guides/pwc_codification_quick_reference_guide.pdf

[36] Topics XBRL taxonomy schema, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/Disclosures/topics.xsd

Arcroles

The disclosure mechanics rules and the reporting checklist rules are expressed using XBRL definition relations. XBRL definition relations is general functionality to represent a relation between two things. Arcroles are used to define the type of relationship.

To articulate the disclosure mechanics and reporting checklist rules, arcroles[37] are used. Currently, the arcroles defined are proprietary. There are two sets of arcroles. The first set is general arcroles[38]. The second set is a set of arcroles that are specifically used for articulating the reporting checklist and disclosure mechanics rules[39].

Rather than using arcroles that are proprietary; a better approach is to have XBRL International create the arcroles, put them into the XBRL International Link Role Registry[40], and have a global standard reporting checklist and disclosure mechanics scheme.

Back to Blocks

So, you might be asking how all of this relates to Blocks. Here is the connection.

Blocks have nothing to do with personal preferences. Blocks have to do with mechanical, structural, mathematical, logical, and other rules that are universally the same for all economic entities that might create a financial report. Blocks factor out personal preferences. Blocks are purely mechanical; they are objective rather than subjective.

Yes, it is true that an economic entity can chose between disclosure alternatives. What goes into a Block can be different for different economic entities. However, once they choose; they MUST follow the rules related to representing that specific disclosures. Yes, it is true that an economic entity can organize the presentation of information to their liking. However, no economic entity can change the statutory or regulatory disclosure rules of the reporting schema to which they must comply.

The Block is the least common denominator that enables thousands of economic entities to consistently[41] represent information such that the information is usable by analysts.

---

[37] XBRL International, XBRL 2.1 Specification, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_3.5.3.9

[38] General arcroles, http://xbrlsite.azurewebsites.net/2016/conceptual-model/cm-arcroles.xsd

[39] Reporting checklist and disclosure mechanics arcroles, http://xbrlsite.azurewebsites.net/2016/conceptual-model/drules-arcroles.xsd

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

[41] Comparing Reporting Styles, Notes on Comparability, http://xbrl.squarespace.com/journal/2015/11/11/comparing-reporting-styles-notes-on-comparability.html

Prototype Theory

There are two approaches to identify something. The first approach is to use a unique identifier[42] that is guaranteed to be unique to some set of objects. This approach works great if the objects you are working with provide unique identifiers. What do you do if you don't have unique identifiers within the set of objects you are working with?

The second approach to identifying something is prototype theory[43]. According to prototype theory, objects can be defined by their resemblance to a unique prototype that is a best or most typical example of the object, sharing the maximum number of features or traits with that prototype. A prototype[44] consists of characteristic features.

As an example, one can understand that something is a "chair" by understanding as many traits as possible about the thing you are looking at, looking at the traits of a chair as defined by a prototype (the undisputed example), and then predicting whether the thing you are looking at is in fact a "chair" by comparing the traits you are observing with the traits of a chair.

This is important because neither the US GAAP XBRL Taxonomy, the IFRS XBRL Taxonomy, nor XBRL-based reports submitted to the SEC by public companies provide identifiers that explicitly and uniquely identify specific disclosures.

Essentially, the logical rules defined in machine-readable form[45] for and used in the process of checking disclosure mechanics also defines a prototype which can be used to identify disclosures within a report. Those rules are prototypes that can be used to query a report for a specific disclosure.

Block Detection and Identification

There are two distinct phases in the process of detecting and identifying a Block within a report:

1. Determining that some fragment of a report is an identifiable Block.
2. Determining the information that is being represented, the disclosure being provided, by that Block.

The first phase where you simply identify THAT some fragment of a report is a Block is straight forward and are applied Network by Network until all the Blocks in a Network are identified; and then all the Networks within a report have been identifies meaning that all the Blocks of a report have been identified.

---

[42] Wikipedia, *Unique Identifier*, https://en.wikipedia.org/wiki/Unique_identifier

[43] Understanding Prototype Theory, http://xbrl.squarespace.com/journal/2013/12/21/understanding-prototype-theory.html

[44] YouTube.com, *Semantics #4 - Prototype Theory*, https://www.youtube.com/watch?v=mff_sPnz_gs

[45] Disclosure mechanics rule, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/disclosure-mechanics/517-rules-def.xml

The second phase, understanding the information that is being conveyed by a Block of information, requires that additional metadata be created which is then used in that process of understanding what the Block represents.

Remember that we are discussing machine-based processes here. A professional accountant would generally have no problem identifying what information is being conveyed by a Block of information. We are trying to get a machine to understand the nature of the Block of information so that the machine can perform work for its human operator.

All Report Blocks

So a report can be broken down to some set of Blocks. That complete set of Blocks is the report. For example, the Microsoft 2017 10-K financial report can be broken down into 192 distinct Blocks. Each of the 192 Blocks must be verified to be correct. If you go look at the validation results of the Microsoft report[46]; you will notice that there are 70 lines in the validation results report.

Why are there only 70 lines and not 192, one report line per Block? There are four specific reasons:

1. Information is conveyed three times generally in a report: (1) the Level 1 Note Text Block which contains an HTML representation for an entire disclosure; (2) the Level 3 Disclosure Text Block which contains an HTML representation for a specific disclosure; and (3) the Level 4 Disclosure Detail which provided individual Facts for a disclosure or statement. Level 1 Note Text Blocks other than the one for the significant accounting policies is not considered in the validation process because where disclosures are presented is a matter of preference and is subjective therefore the validation process ignores the vast majority of Blocks that are Level 1 Note Text Blocks.
2. The validation report puts the Level 3 Disclosure Text Block and the Level 4 Disclosure Detail side-by-side on the report; and therefore many lines show two Blocks that make up both the Level 3 and Level 4 versions of the same disclosure.
3. The validation report lines that are completely gray in color are disclosures for which there are rules provided but the economic entity does not report that disclosure; so those lines can be ignored. (It is possible to turn those disclosures that are not applicable to the economic entity off using the radio buttons on the top of the report).
4. Rules do not exist for 100% of the Blocks that are provided in the report. If rules are not provided, then the second phase of determining what information the Block is

[46] Microsoft Disclosure Mechanics Validation Results, http://xbrlsite.azurewebsites.net/2017/Prototypes/Microsoft2017/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html

conveying cannot be performed. Missing rules means parts of the report are not being verified as being correct.

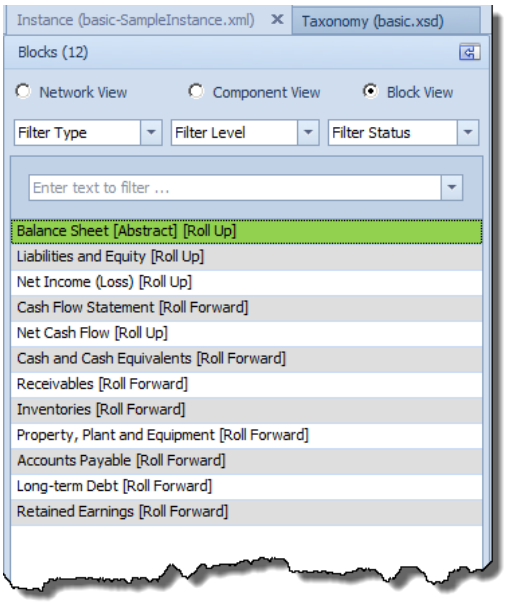
This is a critically important point to understand. The prototype reporting checklist and disclosure mechanics rules that I provide number about 65 or 70 disclosures depending upon how you count them (i.e. how disclosure alternatives are counted). There is not 100% coverage of US GAAP currently provided. What I am showing is a prototype.

But what if you did provide 100% coverage for a reporting scheme and what if 100% of the Blocks were covered by rules? That is exactly what is done with the XASB reporting scheme prototype[47].

The XASB report has 133 Blocks and the verification report has 133 blocks that are being verified to be correct. This is important to understand because it means that there is 100% coverage of the XBRL-based report by the verification rules.

Further, the XASB report prototype exercises all concept arrangement patterns found in US GAAP or IFRS based reports. That proves the capabilities of the system.

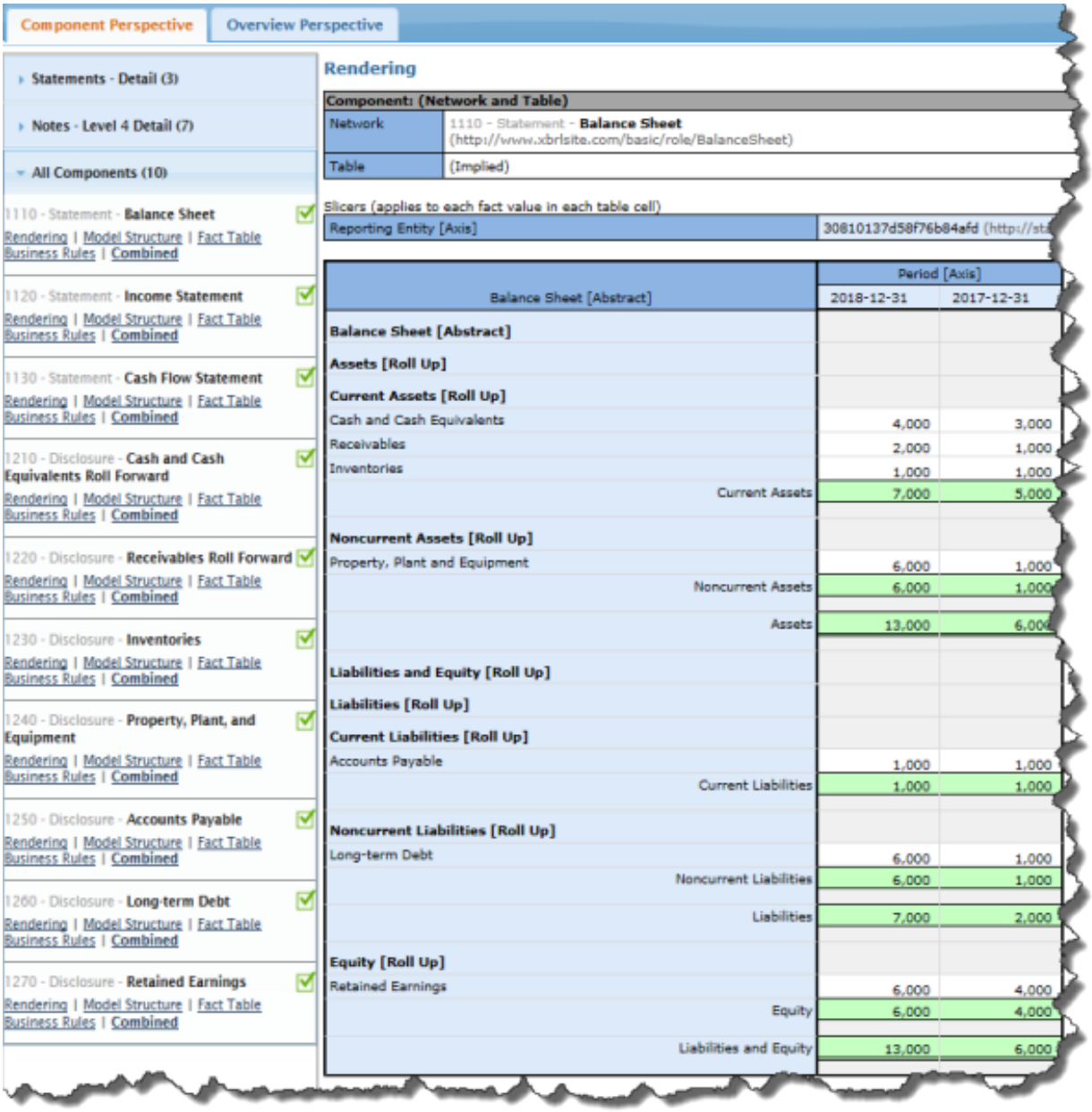
An easier report where 100% coverage can be better seen is to use what I call the basic example[48]. There are no text blocks used and the representation of the report is straight forward. Here is the Block view of the report where you see that there are 12 Blocks shown with the Pesseract software application:



[47] XASB reporting schema, Disclosure Mechanics Results, http://xbrlsite.azurewebsites.net/2017/Prototypes/XASB/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html

[48] Basic example, XBRL instance, http://xbrlsite.azurewebsites.net/2018/Prototypes/Basic/Basic-XASB-ConsistentRF/basic-SampleInstance.xml

Here is the XBRL Cloud evidence package that helps you see the Blocks in the report[49] if you don't have software where you can load the report:



The XBRL Cloud view of this information shows 10 Components, that software understands Blocks, but does not provide a separate listing for Blocks in this interface. Two of the Components have two Blocks (balance sheet and cash flow statement) while all the Components have one Block so there are a total of 12 Blocks the same as the Pesseract software's count of Blocks.

[49] Basic example, XBRL Cloud Evidence Package, http://xbrlsite.azurewebsites.net/2018/Prototypes/Basic/Basic-XASB-ConsistentRF/evidence-package/contents/index.html#Rendering-N0-Implied.html

Here is the XBRL Cloud reporting checklist and disclosure mechanics validation report[50]. You can see a list of 12 Blocks here. Note that 100% of the Blocks are represented correctly per the rules used to evaluate the report:

#	Disclosure	Category	Level	Pattern	Applicable	Found	Disclosure Consistent	Representation Concept [TEXT BLOCK]	Representation Concept [DETAIL]	Checklist Category	Reason
1	Balance Sheet		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]
2	Assets [Roll Up]		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Assets	Part of disclosure	Disclosure always required
3	Liabilities and Equity [Roll Up]		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Liabilities and Equity	Part of disclosure	Disclosure always required
4	Income Statement		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Net Income (Loss)	Required disclosure	Disclosure always required
5	Cash Flow Statement, Direct Method		Level4Detail	ROLL UP	True	True	CONSISTENT	NOT-EXPECTED	Net Cash Flow	Required disclosure	Disclosure always required
6	Receivables [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Receivables	Required disclosure	Disclosure always required
7	Cash and Cash Equivalents [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Cash and Cash Equivalents	Line item exists, then disclosure required	Required because line item basic:CashAndCashEquivalents was reported
8	Inventories [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Inventories	Line item exists, then disclosure required	Required because line item basic:Inventories was reported
9	Accounts Payable [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Accounts Payable	Line item exists, then disclosure required	Required because line item basic:AccountsPayable was reported
10	Property, Plant, and Equipment [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Property, Plant and Equipment	Line item exists, then disclosure required	Required because line item basic:PropertyPlantAndEquipment was reported
11	Long-Term Debt [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Long-term Debt	Line item exists, then disclosure required	Required because line item basic:LongtermDebt was reported
12	Retained Earnings [Roll Forward]		Level4Detail	ROLL FORWARD	True	True	CONSISTENT	NOT-EXPECTED	Retained Earnings	Line item exists, then disclosure required	Required because line item basic:RetainedEarnings was reported

The Pesseract application provides the same results, again using a slightly different interface. Note the count of 12 Blocks:

Primary Information

#	Disclosure	Category	Level	Pattern	Disclosure Found	Disclosure Consis...	Applicable	Representation Concept [TEXT BLOCK]	Representation Concept DETAIL
1	Accounts Payable [Roll For...	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Accounts Payable
2	Assets [Roll Up]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Assets
3	Balance Sheet	Statement	UNKNOWN	Component	True	CONSISTENT	True	-	-
4	Cash and Cash Equivalents...	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	NOT-FOUND
5	Cash Flow Statement, Dire...	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Cash Flow
6	Income Statement	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Net Income (Loss)
7	Inventories [Roll Forward]	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Inventories
8	Liabilities and Equity [Roll Up]	Unknown	Level4Detail	RollUp	True	CONSISTENT	True	NOT-EXPECTED	Liabilities and Equity
9	Long-Term Debt [Roll Forw...	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Long-term Debt
10	Property, Plant, and Equip...	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Property, Plant and Equipment
11	Receivables [Roll Forward]	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Receivables
12	Retained Earnings [Roll For...	Unknown	Level4Detail	RollForward	True	CONSISTENT	True	NOT-EXPECTED	Retained Earnings

In summary, what is being shown is that if a complete set of rules is provided the covers 100% of the Blocks being represented; then 100% of the report can be verified to be consistent with the expectations which are articulated within the machine-readable rules.

All this is enabled by the ability to break down an XBRL-based report into fragments that you can work with; we call these fragments Blocks.

[50] Basic example, reporting checklist and disclosure mechanics, http://xbrlsite.azurewebsites.net/2018/Prototypes/Basic/Basic-XASB-ConsistentRF/ReportingChecklistResults/Disclosure%20Mechanics%20and%20Reporting%20Checklist.html

Complete System

This document focused on the notion of Blocks. These Blocks empower the reporting checklist and disclosure mechanics functionality which is discussed within in this document. But, to verify an XBRL-based report completely you need additional capabilities. These additional capabilities are explained in detail in the document *Chain of Capabilities Necessary to Automate Accounting Processes*[51]. The following is a summary of the chain of capabilities necessary to make sure that an XBRL-based report has been created correctly. Please refer to the other document for details:

- XBRL technical syntax consistency
- Model structure consistency
- Reporting styles
- Continuity cross-checks
- Types
- Reporting checklist
- Disclosure mechanics
- Manual review of non-automatable tasks

As you can see the reporting checklist and disclosure mechanics functionality is included in the necessary chain of capabilities.

As is explained in the document *Leveraging the Theoretical and Mathematical Underpinnings of a Financial Report*[52]; a financial report can be explained using mathematics and logic.

If you think beyond individual financial reports and the need to exchange information, while I understand less about ideas such as digital distributed ledgers, smart contracts, triple-entry accounting[53]; those ideas will certainly play a major role in a complete system which leverages these technologies of a digital environment.

Part of this system must, most likely, include the independent audit. Visionaries such as Jun Dai and Miklos Vasarhelyi of Rutgers University in their papers *Imagineering Audit 4.0*[54] and *Toward Blockchain-Based Accounting and Assurance*[55], foresee new techniques that leverage technology to make audits better, faster, and cheaper.

Techniques for managing quality by manufacturing companies exist and have been effectively used to control quality. These same techniques can be used in accounting, reporting, auditing, and analysis in a digital environment.

---

[51] Charles Hoffman, *Chain of Capabilities Necessary to Automate Accounting Processes*, http://xbrlsite.azurewebsites.net/2018/Library/ChainOfCapabilities.pdf

[52] Charles Hoffman, *Leveraging the Theoretical and Mathematical Underpinnings of a Financial Report*, http://xbrlsite.azurewebsites.net/2018/Library/TheoreticalAndMathematicalUnderpinningsOfFinancialReport.pdf

[53] Charles Hoffman, *Digital Distributed Ledgers+Smart Contracts+XBRL*, http://xbrlsite.azurewebsites.net/2018/Library/DistributedLedgersSmartContractsXBRL.pdf

[54] Jun Dai and Miklos Vasarhelyi, *Imagineering Audit 4.0*, http://aaajournals.org/doi/abs/10.2308/jeta-10494

[55] Jun Dai and Miklos Vasarhelyi, *Toward Blockchain-Based Accounting and Assurance*, http://aaajournals.org/doi/10.2308/isys-51804

Lean Six Sigma[56] is a discipline that combines the problem solving methodologies and quality enhancement techniques of Six Sigma with the process improvement tools and efficiency concepts of Lean Manufacturing. Born in the manufacturing sector, Lean Six Sigma works to produce products and services in a way that meets consumer demand without creating wasted time, money and resources.

Specifically, Lean is 'the purposeful elimination of wasteful activities.' It focuses on making process throughout an organization faster, which effects production over a period of time. Six Sigma works to develop a measurable process that is nearly flawless in terms of defects, while improving quality and removing as much variation as possible from the system.

While financial report quality control is generally extremely high, it is also extremely manual in nature. The modern finance platform[57] will use the techniques of Lean Six Sigma to measure and automate accounting, reporting, auditing, and analysis processes.

Putting all these pieces together will make the old-school approaches to creating and auditing a financial report look barbaric.

Conclusion

The Block is a notion that is used as a matter of convenience. Employed correctly, Blocks can be used to enable functionality which can be achieved in no other way. While the technique of employing Blocks was shown in the narrow context of creating XBRL-based financial reports; these same ideas are applicable to the broader arena of accounting process automation.

Acknowledgements

Most of the ideas in this document come from discussions and feedback that I received over the past 15 or so years from many, many colleagues who are too numerous to list here. That input was critical to shaping the thoughts expressed in this document. Thank you to the entire XBRL community!

[56] *Comprehensive Introduction to Lean Six Sigma for Professional Accountants*, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.72_LeanSixSigma.pdf

[57] *The Modern Finance Platform*, http://xbrl.squarespace.com/journal/2018/7/15/the-modern-finance-platform.html