Representing a Logic System (a.k.a. Knowledge Graph) Using Global Standard XBRL

Understanding how to construct a simple non-complex logic system (a.k.a. knowledge graph, formal system) using the global standard XBRL technical syntax

By Charles Hoffman, CPA (August 5, 2024) (Work in Progress)

"I skate to where the puck is going to be, not where it has been." Wayne Gretzky, Canadian hockey star

What others refer to as a knowledge graph and explain as a technical artifact usually implemented using a specific technology preferred by the person doing the explaining; I see as a logical artifact, a logic system, that can be implemented using a variety of technical approaches.

This is what I mean.

Stardog, a software vendor, provides this video, *What is a Knowledge Graph?*[1] In that short three minute and ten second video, the narrators make the following comments:

- "A knowledge graph is a combination of your data and its meaning."
- "Knowledge graphs represent data intelligently as things. These things are also referred to as entities."
- "…context can be separate from the data itself…"
- "Knowledge graphs were built out of the Semantic Web and have a rich technical history…"
- "A flexible, semantic data layer for answering complex queries across data silos."
- "Connects any data source/location and virtualizes access."
- "Enriches real-world context into data."
- "Infer new relationships, patterns and insights into data."
- "Semantically searches data by meaning."
- "Knowledge graphs make use of RDF."
- "Infuses additional context into the data."
- "Knowledge graphs are a framework in a platform for capturing your data, wherever it lives."
- "Make use of that context to infer additional insights."

That explanation is fairly typical. That is typical about the explanation is the following.

[1] YouTube.com, *What is a knowledge Graph?*, https://youtu.be/6BotfiE5Np4

First, knowledge graphs are explained in terms of the technology that is used to implement the knowledge graph, not in terms of what a knowledge graph does.

Second, knowledge graphs are explained in terms of "data" and not in terms of what they really are which is "information".

Third, knowledge graphs are explained in terms of the problem that knowledge graphs solve which is the problem of businesses storing data within separate databases that are not interconnected and therefore not in context.

Fourth, and this is a pet peeve of mine, knowledge graphs are explained in terms of one typical implementation of a knowledge graph which is RDF[2]. Part of this is because each software vendor that explains knowledge graphs and who sells a product wants you to use their product and their implementation of a knowledge graph.

In this document, I am going to take a completely different approach to explaining the idea of what people refer to as a "knowledge graph". I am not going to explain the notion of a knowledge graph in terms of a common problem enterprises have which is a bunch of disjointed data silos. I am going to try and stay away from computer science and information technology technical things. I am going to provided two simple "knowledge graph" systems that will help a business professional get their head around what a knowledge graph is.

To achieve what I am trying to achieve, I am going to explain what a knowledge graph is by explaining what a knowledge graph actually does. A knowledge graph is a technical approach to representing a logic system or formal system of some sort.

Why do you Need a Logic System?

People refer to the notion of a knowledge graph or digital twin or logical twin[3] or I refer to them as professional knowledge graphs[4]. A logical digital twin of a financial report represented using XBRL is an example of such a system[5].

Gartner has created an entirely new market segment which they are calling **decision intelligence**[6]. What they have effectively done is establish a new market for what people have been calling knowledge products.

A knowledge product[7] is refined and actionable information that has been processed, organized, and/or structured in some way or put into practice in some way making the information super-useful. The information is ready to use. The knowledge is derived from expertise, research, lessons learned. Knowledge products allow the user of the knowledge product to make informed decisions or

---

[2] W3C, *Resource Description Framework (RDF),* https://www.w3.org/RDF/

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

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

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

[6] Gartner, *Market Guide to Decision Intelligence Platforms*, https://www.gartner.com/doc/reprints?id=1-2I5NX8E3&ct=240723&st=sb

[7] **Knowledge Products Offer New Business Models**, https://digitalfinancialreporting.blogspot.com/2024/02/a-new-business-model-is-emerging-which.html

better decisions. Knowledge products are part of a spectrum really. The full spectrum is provide in this list below:

- **Data product**: a reusable raw and unprocessed data asset, engineered to deliver a trusted dataset to a user for a specific purpose.
- **Information product**: organized, processed, and perhaps even interpreted data which provides context and meaning.
- **Knowledge product**: refined and actionable information that has been processed, organized, and/or structured in some way or put into practice in some way making the information super-useful.
- **Decision product:** tell a business professional what they need to do or actually execute an action making use of the information of the decision product.

In essence, information products, knowledge products, and decision products are logic systems.

If you have not thought about all this a lot; then it would serve the reader well to get a good grounding by becoming familiar with the information in *Problem Solving Systems*[8]. That document provides solid grounding for thinking about the notion of a logic system.

Logic System or Formal System

Logic is a formal set of principles and rules that form a framework for correct reasoning and communications. There are many different logics and there are many different technical implementations of logics. To keep this simple I am simply going to tell you that the logic that I am using which is a subset of first-order logic called Horn Logic[9] or DATALOG[10] that is safe to use and also very powerful when implementing computability.

The semantics of first-order logic are agreed to and very well understood. However, everyone and their bother has their own version of that terminology and none of those versions are both complete and explained in terms that a business professional can understand. And, therefore, I have to come up with my own set of terminology which I will provide now.

A **logic statement** is a proposition, claim, assertion, belief, idea, or fact about or related to a logic system. A logical statement is a piece of information. A logical statement is declarative (e.g. not a question).

A **logic system** (a.k.a. formal system[11]) is a group of interacting or interrelated logical statements that act according to a set of logical patterns to form a unified whole or conceptualization. It is the set of logical statements that defines a logic system. A set of logical statements forms a logical theory that explains the dynamics of the logic system.

A **logical theory** is a logic system that provides a formal, explicit, deliberate specification of the important specific details of an intended shared conceptualization between a set of stakeholders of a

[8] Problem Solving Systems, https://digitalfinancialreporting.blogspot.com/2024/03/problem-solving-systems.html
[9] Wikipedia, *Horn Clause*, https://en.wikipedia.org/wiki/Horn_clause
[10] Michelin, *An Introduction to Datalog*, https://blogit.michelin.io/an-introduction-to-datalog/
[11] Wikipedia, *Formal System*, https://en.wikipedia.org/wiki/Formal_system

logic system for an area of knowledge to achieve some specific set of goals/objectives. The logical theory describes the aim of the logic system.

- **Conceptualization**: Set of declarative logical statements that describe what is permissible per a system. Conceptualizations describe/specify, enable creation/construction, enable review/verification, and enable extraction/analysis.
- **Model**: Set of logical statements that form/describe structures that are consistent with and permissible interpretations of that model. Models provide flexibility.
- **Term**: Logical statement that defines/describes an idea/thing which is used within the logical conceptualization and distinguishes one idea/thing from other ideas/things. Ideas/things tend to be nouns.
- **Structure**: Set of logical statements and type of term which describe assemblies of terms, associations, and restrictions. A structure is an assembly of terms, associations, restrictions that are consistent with and permissible per that structure (compound, decomposable). A structure is a compound idea/thing and can be decomposed. Structures provide the ability to identify, refer to, and work with sets of ideas/things.
- **Association**: Logical statement that describes a permissible interrelationship between an idea/thing within a structure. Associations tend to be verbs.
    - **Categorization (is-a)**: Logical statement that formally groups or classifies an idea/thing into a useful set. A type is an important category of association. (a.k.a. class, subclass, superclass, generalization, specialization, wider, narrower)
    - **Compositional (has-a)**: Logical statement that indicates that an idea/thing is part of some other idea/thing (a.k.a. part-of, has-part, property-of). Expressing important qualities or traits of a term. Category of association.
    - **Aggregational (summation):** Logical statement that indicates summation or aggregational associations that are mutually exclusive, completely exhaustive, no overlap.
    - **Navigational (parent-child)**: Logical statement is an informal set of relations which are used to navigate from one thing to another thing without defining formal meaning.
- **Constraint**: Logical statement that describes a rule, constraint, restriction, or assertion of a structure or model. Restrictions tend to be convertible into IF…THEN…ELSE types of logical statements. Restrictions can be connected together by joining restrictions using logical connectors (e.g. AND, OR, NOT, NOR, XOR, NAND).
- **Fact**: Logical statement about the numbers and words that are described by and instances of a model. Facts are differentiated from other facts using aspects (a.k.a. dimensions) which provide explicit context.

A conceptualization can be **complete** or **incomplete** (e.g. missing important logical statements); can be **consistent** or **inconsistent** (e.g. logical statements contradict one another); can be **precise** or **imprecise** (e.g. logical statements are not consistent with reality). A **properly functioning logic system** is said to be verifiably complete, consistent, and precise.

A conceptualization can be **adequate** or **inadequate** meaning that the conceptualization tested and validated to yield reliability and accuracy consistent with the specified purpose of the

conceptualization's stakeholders. A logic system is said to be adequate if it meets the aim of the stakeholders of the logic system.

A logic system has a level of complexity[12]. There are two groups of complexity: **complex** and **non-complex**.

- Non-complex systems are computable, they have computability[13].
- Complex systems are not computable.

Computability is defined as the ability to solve a problem in an effective manner. Fundamentally, this means that an algorithm can be defined and a machine or human can perform the algorithm and effectively solve a problem or perform a task.

To better understand logic systems; it is good to group logic systems into three specific groups: simple system, complicated system, or complex system.

- **Simple system**: The system is "non-complex" and therefore computable; clear and obvious for a non-subject matter expert to understand, and the set of elements, categories, and interaction patterns are fully understood. Control techniques can be used to eliminate all risk from the system.
- **Complicated system**: The system is "non-complex" and therefore computable; only clear and obvious for a subject matter expert in the area of knowledge to which the system relates to understand, and the set of elements, categories, and interaction patterns are fully understood. Control techniques can be used to eliminate all risk from the system.
- **Complex system**: The system is "complex" and therefore NOT computable; tend to lack clear boundaries, tend to be constantly changing and evolving, there tend to be large numbers of elements, categories, and interaction patterns which are not completely understood, the system seems to contradict itself on occasion, and the number of forces impacting the system tends to be large and the dynamics are not well understood. Control techniques cannot be used to fully eliminate all risk from the system.

While it is the case that a complex system can be divided into subsystems that might be simple, complicate, or complex; for the purposes of this document, we are going to stick with simple systems and complicated systems that are “non-complex” and therefore “computable”.

One important thing to understand is the notion of **atomic design methodology**[14]. Atomic Design Methodology introduces the notion of higher-level objects which are created by lower-level objects. In this way, a high-level model can be created for a logic system.

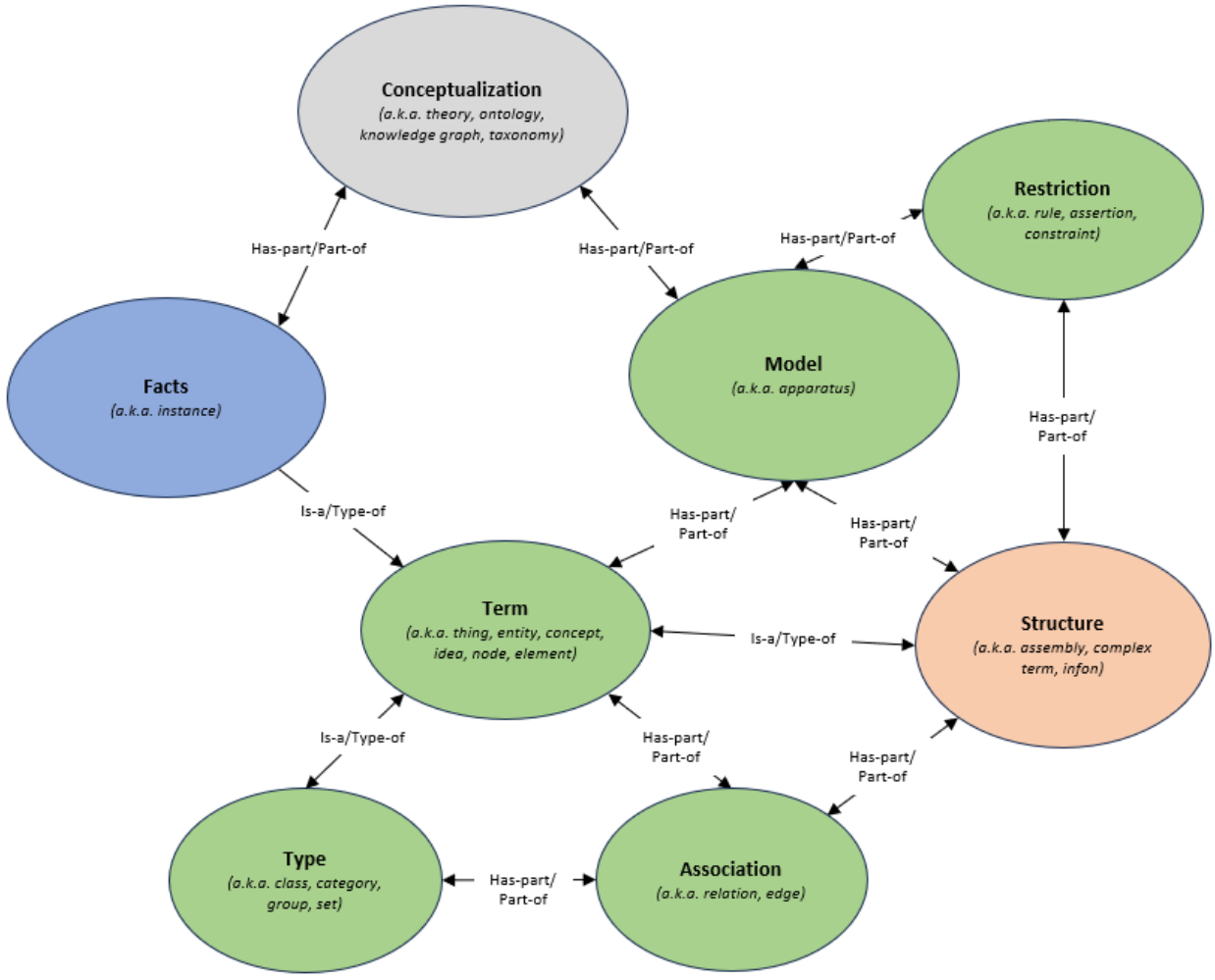
- **Atoms** are the basic building blocks. Atoms are indivisible elementary building blocks.
- **Molecules** are combinations of two or more atoms. These combinations of atoms take on their own unique properties, and become more tangible and operational than atoms. Molecules are decomposable into the atoms that make up the molecule.
- **Organisms** are assemblies of molecules functioning together as a unit. Organisms are more complex and sophisticated than molecules. Organisms might be constructed from other

[12] Jolly Contrarian, *Complexity*, https://jollycontrarian.com/index.php?title=Complexity
[13] Wikipedia, *Computability*, https://en.wikipedia.org/wiki/Computability
[14] Atomic Design *Methodology*, https://digitalfinancialreporting.blogspot.com/2023/12/atomic-design-methodology.html

organisms to create *compound organisms*. Organisms are decomposable into molecules and atoms.

- **Species**: There can be "species" of organisms. (I added this; not explicitly described by Atomic Design methodology, seems like it might be implied. Maybe a "template" is used to define a "specie".

This is my somewhat informal take on the relationship between the high-level objects of the logic system meta-meta-model[15] that is used to define a logic system: (meaning, this meta-meta-model should be defined using UML or some other formal modeling mechanism to make this crystal clear)



Every meta-model defined using the above meta-meta-model must fit 100% into that meta-meta-model. Further, every model defined using some defined meta-model must likewise fit within this meta-meta-model.

[15] Meta-meta-model, http://xbrlsite.com/2024/Library/meta-meta-model.pdf

Properly Functioning Logic System

A logic system that is flawed really has little value because it will lead to decisions that are wrong. The *Seattle Method*[16] provides detailed guidance related to what a properly functioning logic system is and how to make sure a financial report is a properly functioning logic system. What you need to understand and take away from this paragraph is that (a) properly functioning logic system is definable and (b) how to make sure such a logic system is properly functioning is knowable and repeatable processes for achieving this level of quality is knowable.

Requisite variety[17], in the context of computer science, refers to the principle that in order to effectively regulate (control) a system, the regulator (controller) of that system must possess a sufficient range of actions to counter act the variety of important potential disturbances that system might encounter. The principle of requisite variety ensures that the system's internal state remains as close as possible to the desired goal state of the system. Requisite variety requires that there be a balance or conscious "matching" between the potential "perturbance variety" or potential possible disturbances which may occur within a system and the "control variety" which is the information such as rules available to the system to make sure the "residual variety" is as close to 0 as possible, preferably equal to zero.

System complexity and computability[18] that can be achieved is a well understood science and art form. While information technology professionals might not understand the science and art of complexity and computability, those familiar with the disciplines of cybernetics, informatics, and Lean Six Sigma do[19]. Everything you need to achieve the necessary quality level is provided by the guidance of the *Seattle Method* as it relates to financial reporting.

Seattle Method

The *Seattle Method*[20] provides good practice guidance related to representing a logic system meta-model that conforms to the meta-meta-model of a logic system described above. I will not discuss the *Seattle Method* further in this document but I will use the *Seattle Method* guidance to construct two simple logic systems and will point you to a third complicated logic system.

A very important note to understand is that a financial report is a type of business report. Every financial report is a business report. It is not the case that every business report is also a financial report. The important point to understand is that the general ideas provided by the examples of financial reports (which is my personal area of knowledge) are also applicable to business reports in general and also to logic systems.

Business reports and financial reports are a type of logic system.

---

[16] Seattle Method, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf

[17] *The Law of Requisite Variety*, https://digitalfinancialreporting.blogspot.com/2024/07/the-law-of-requisite-variety.html

[18] *System Complexity and Computability*, https://digitalfinancialreporting.blogspot.com/2024/07/system-complexity-and-computability.html

[19] *Informatics, Cybernetics, Lean Six Sigma*; https://digitalfinancialreporting.blogspot.com/2024/07/informatics-cybernetics-lean-six-sigma.html

[20] XBRLSITE.COM, *Seattle Method*, http://xbrlsite.com/seattlemethod/SeattleMethod.pdf

First Simple System: *Accounting Equation*

The first simple logic system that I will create is the accounting equation[21] which is a well understood logic system known to most business professionals because they received basic training in accounting and there is really no dispute as to the logic that underlies this logic system referred to as the accounting equation. Because of this, we can focus on constructing the logic system rather than arguing endlessly about the fundamental logic of that system.

While the accounting equation logic system is a very simple logic system; even simple systems have all the characteristics of any logic system. Below is a "pseudo code" representation of the logic of the accounting equation logic system:

- Model (Conceptualization)
  - Terms
    - Assets
    - Liabilities
    - Equity
  - Types
    - Assets is a type-of Concept
    - Liabilities is a type-of Concept
    - Equity is a type-of Concept
  - Structures
    - Balance Sheet
  - Restrictions (a.k.a. rules)
    - Assets = Liabilities + Equity
  - Associations
    - Assets part-of Balance Sheet
    - Liabilities part-of Balance Sheet
    - Equity part-of Balance Sheet
  - Properties
    - Assets has-property Balance with value Debit
    - Liabilities has-property Balance with value Credit
    - Equity has-property Balance with value Credit
    - Assets has-property PeriodType with value Instant
    - Liabilities has-property PeriodType with value Instant
    - Equity has-property PeriodType with value Instant
  - Facts
    - ABC company has Assets for-instant 2020-12-31 with value 5,000 USD
    - ABC company has Liabilities for-instant 2020-12-31 with value 1,000 USD
    - ABC company has Equity for-instant 2020-12-31 with value 4,000 USD

The following is a very basic rendering of the accounting equation logic system:

[21] Wikipedia, *Accounting Equation*, https://en.wikipedia.org/wiki/Accounting_equation

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

Result	Rule
Pass	$Assets = $Liabilities + $Equity

The following are representation of the accounting equation logic system represented in XBRL which was created by me and verified to be a properly functioning logic system, in RDF/OWL/SHACL which was prepared by someone else, and in Cypher created using Neo4j which stands in for ISO GQL until I can get that done using software which supports that standard syntax.

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

XBRL: (verified to be properly functioning logic system)

http://www.xbrlsite.com/seattlemethod/platinum/ae/ae_ModelStructure.html

http://www.xbrlsite.com/seattlemethod/platinum/ae/index.html

https://auditchain.infura-ipfs.io/ipfs/QmPeD6DrM8xsgwiSnwo6JR1njJmurDgBkQQf1kLAooJRwh/

RDF/OWL/SHACL: (not verified)

https://xbrlsite.azurewebsites.net/2024/sbrm/ae-report.ttl

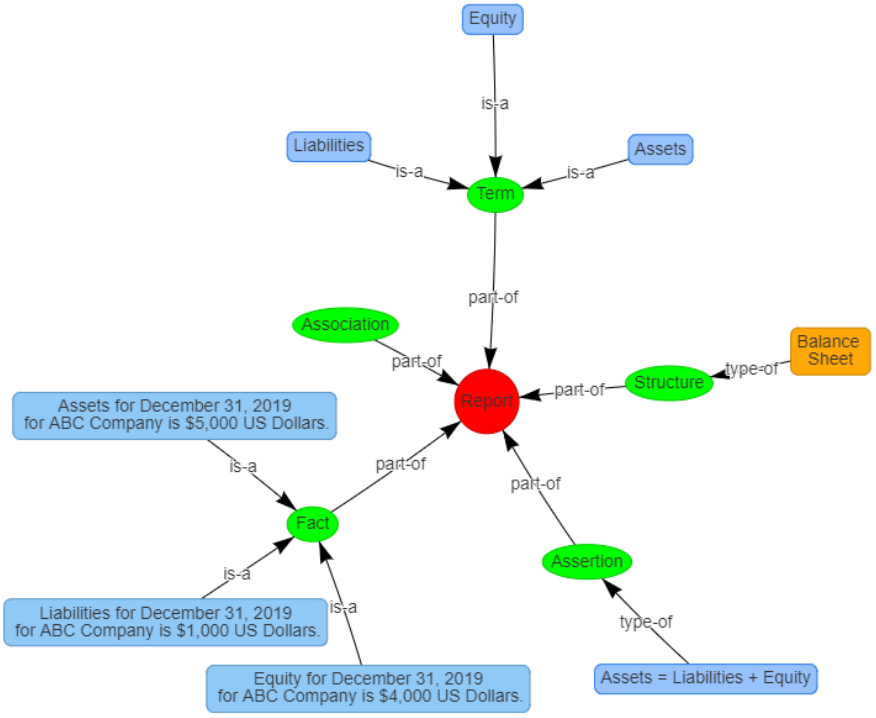
https://xbrlsite.azurewebsites.net/2024/sbrm/ae-model.ttl

https://xbrlsite.azurewebsites.net/2024/sbrm/sbrm.ttl

Cypher (GQL): (not verified)

http://www.xbrlsite.com/seattlemethod/platinum/ae/Cypher/Cypher.txt

Simple knowledge graph:

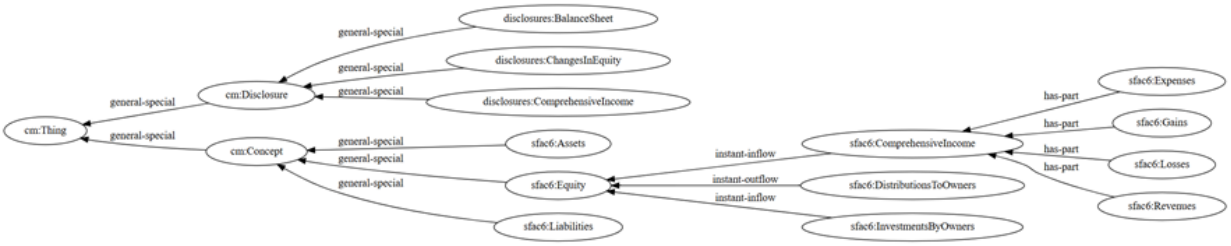


Second Simple System: *SFAC 6 Elements of Financial Statement*

The second simple system that I will create is a portion of the Financial Accounting Standards Board's *Statement of Financial Accounting Concepts No. 6, Elements of Financial Statement*[22]. This simple system defines 10 terms, 3 structures, 3 restrictions (rules), and while more complicated than the first example is still relatively simple to understand.

Again, SFAC 6 is a rather straight-forward and simple logic system which you would be hard pressed to find someone that disagrees with the logic. People may want to fight over the definitions of some of the terms or the boundaries of the system. For example, I am consciously leaving the discussion of "net assets" out of this example but do include it in my SFAC 8 example. Why? To keep the SFAC 6 example as simple as possible.

What the SFAC 6 example does beyond the accounting equation example is to (a) add two additional structures and (b) add two new types of mathematical rules.



[22] FASB, *SFAC 6 Elements of Financial Statement*, https://www.fasb.org/Page/document?pdf=aop_CON6.pdf&title=CON%206%20(AS%20AMENDED)

Assets = $3{,}500^{T1}$; 0^{T0}

Liabilities = 0^{T1}; 0^{T0}

Equity = $3{,}500^{T1}$; 0^{T0}

Revenues = 7,000

Expenses = 3,000

Gains = 1,000

Losses = 2,000

Comprehensive income = 3,000

Investments by Owners = 1,000

Distributions to Owners = 500

Assets = Liabilities + Equity

Comprehensive Income = Revenues - Expenses + Gains - Losses

$Equity^{T1} = Equity^{T0} + Comprehensive\ Income^{P1} + Investments\ by\ Owners^{P1} - Distributions\ to\ Owners^{P1}$

Balance Sheet

Balance Sheet [Line Items]	Period [Axis] 2020-12-31	2019-12-31
Balance Sheet [Arithmetic Expression]		
Assets	3,500	0
Liabilities	0	0
Equity	3,500	0

Changes in Equity

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

Income Statement

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

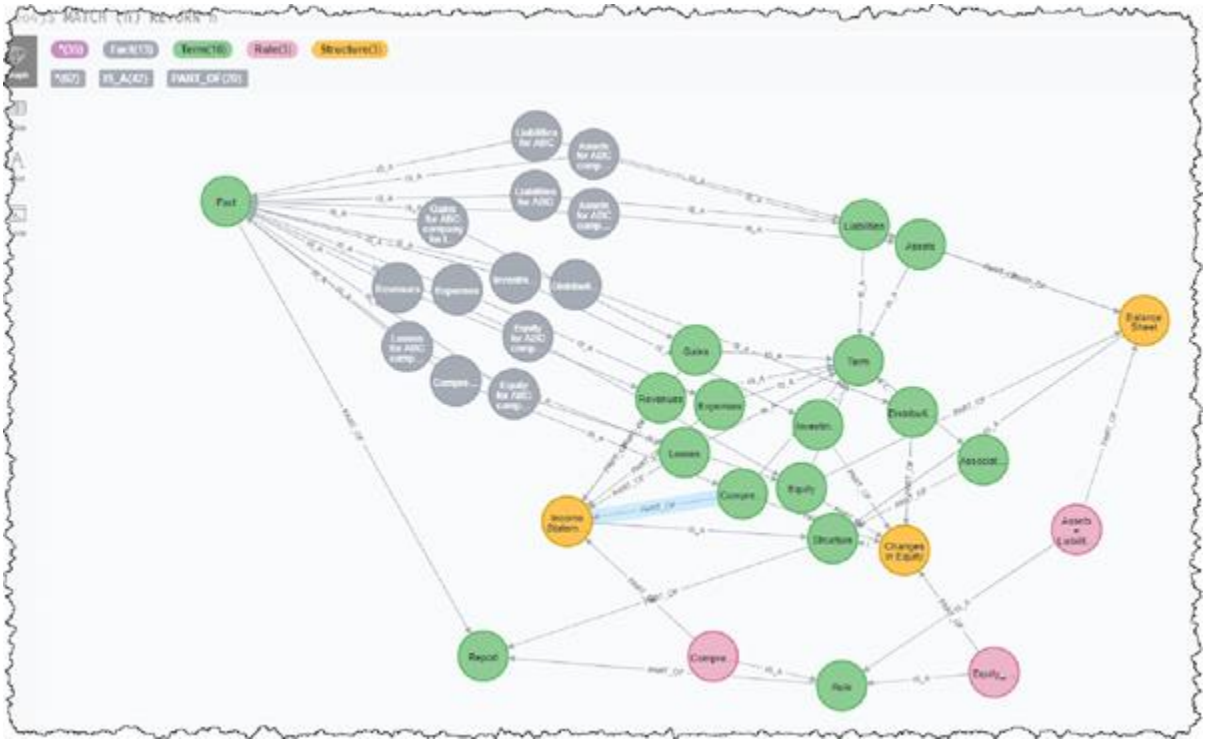
XBRL:

http://www.xbrlsite.com/seattlemethod/platinum/sfac6/sfac6_ModelStructure.html

Line		Object Class	Period Type	Balance	Report Element Name
1	**11-Statement of Financial Position**	Network			http://xbrlsite.com/seattlemethod/sfac6/role/BalanceSheet
2	**Balance Sheet [Hypercube]**	Table			sfac6:BalanceSheetHypercube
3	Balance Sheet [Line Items]	LineItems			sfac6:BalanceSheetLineItems
4	Balance Sheet [Arithmetic]	Abstract			sfac6:BalanceSheetArithmetic
5	Assets	Concept (Monetary)	As Of	Debit	sfac6:Assets
6	Liabilities	Concept (Monetary)	As Of	Credit	sfac6:Liabilities
7	Equity	Concept (Monetary)	As Of	Credit	sfac6:Equity
8	**21-Statement of Comprehensive Income**	Network			http://xbrlsite.com/seattlemethod/sfac6/role/ComprehensiveIncome
9	**Comprehensive Income Statement [Hypercube]**	Table			sfac6:ComprehensiveIncomeStatementHypercube
10	Comprehensive Income Statement [Line Items]	LineItems			sfac6:ComprehensiveIncomeStatementLineItems
11	Comprehensive Income [Roll Up]	Abstract			sfac6:ComprehensiveIncomeRollUp
12	Revenues	Concept (Monetary)	For Period	Credit	sfac6:Revenues
13	(Expenses)	Concept (Monetary)	For Period	Debit	sfac6:Expenses
14	Gains	Concept (Monetary)	For Period	Credit	sfac6:Gains
15	(Losses)	Concept (Monetary)	For Period	Debit	sfac6:Losses
16	Comprehensive Income	Concept (Monetary)	For Period	Credit	sfac6:ComprehensiveIncome
17	**31-Statement of Changes in Equity**	Network			http://xbrlsite.com/seattlemethod/sfac6/role/ChangesInEquity
18	**Changes in Equity [Hypercube]**	Table			sfac6:ChangesInEquityHypercube
19	Changes in Equity [Line Items]	LineItems			sfac6:ChangesInEquityLineItems
20	Changes in Equity [Roll Forward]	Abstract			sfac6:ChangesInEquityRollForward
21	Equity, Beginning Balance	Concept (Monetary)	As Of	Credit	sfac6:Equity
22	Comprehensive Income	Concept (Monetary)	For Period	Credit	sfac6:ComprehensiveIncome
23	Investments by Owners	Concept (Monetary)	For Period	Credit	sfac6:InvestmentsByOwners
24	(Distributions to Owners)	Concept (Monetary)	For Period	Debit	sfac6:DistributionsToOwners
25	Equity, Ending Balance	Concept (Monetary)	As Of	Credit	sfac6:Equity

Cypher:

http://www.xbrlsite.com/seattlemethod/platinum/sfac6/sfac6_Cypher.txt



Complicated System: *AASB 1060 Financial Reporting Scheme*

The complicated system example is about 20% of a full financial reporting scheme published by the Australian Accounting Standards Board, General Purpose Financial Statements – Simplified Disclosures for For-Profit and Not-for-Profit Tier 2 Entities.

What the AASB 1060 example does is add more terms, structures, restrictions (a.k.a. rules), types, and so forth. AASB is a real looking financial reporting scheme, although it really is only about 20% complete. It is only a working proof of concept due to lack of resources to actually complete the XBRL-based taxonomy and create real financial reports. However, 100% of the logic patterns that would ever exist within a real financial report exist within the working proof of concept.

Only an XBRL-based representation is provided, again due to lack of resources. But just as the accounting equation example was represented within RDF/OWL/SHACL and Cypher; so to can this working proof of concepts be represented using those two additional technical formats.

XBRL:

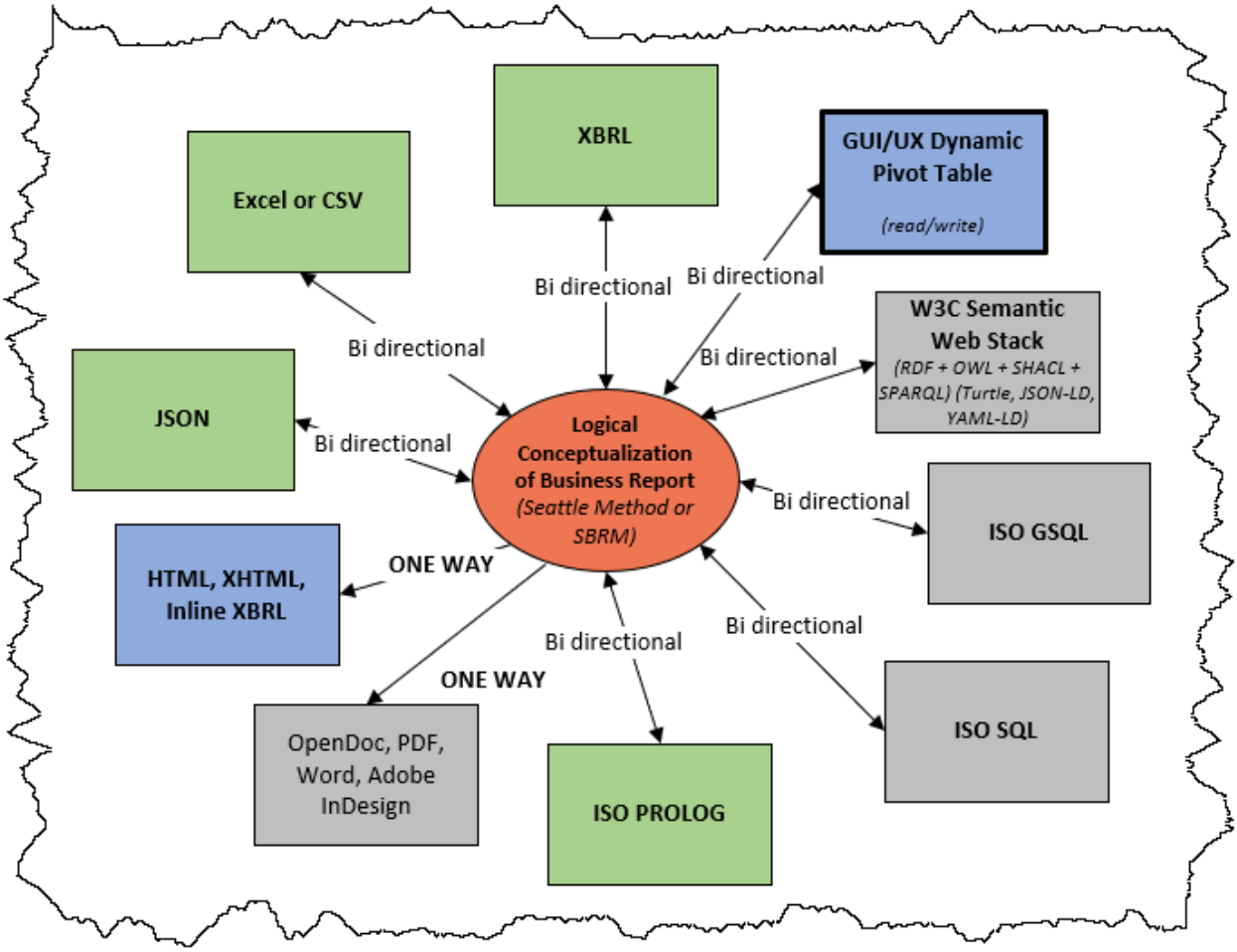
https://xbrlsite.azurewebsites.net/2021/reporting-scheme/aasb1060/base-taxonomy/aasb1060_ModelStructure2.html

Line	Label	Object Class	Period Type	Balance	Report Element Name	References
1	001-Documentation-Report Information	Network			http://www.xbrlsite.com/aasb1060/role/Documentation/ReportInformation	
2	**Report Information [Hypercube]**	Hypercube			aasb1060:ReportInformationHypercube	AASB 1060 31
3	**Report Information [Line Items]**	LineItems			aasb1060:ReportInformationLineItems	AASB 1060 31
4	Report Information [Set]	Abstract			aasb1060:ReportInformationSet	
5	Reporting Period End Date	Concept (Date)	For Period		aasb1060:ReportingPeriodEndDate	AASB 1060 31 c
6	Reporting Period	Concept (Date)	For Period		aasb1060:ReportingPeriod	AASB 1060 31 c
7	Presentation Currency	Concept (Text/String)	For Period		aasb1060:PresentationCurrency	AASB 1060 31 d
8	Level of Rounding	Concept (Text/String)	For Period		aasb1060:LevelOfRounding	AASB 1060 31 e
9	002-Documentation-Reporting Entity Information	Network			http://www.xbrlsite.com/aasb1060/role/Documentation/ReportingEntityInformation	
10	**Reporting Entity Information [Hypercube]**	Hypercube			aasb1060:ReportingEntityInformationHypercube	
11	**Reporting Entity Information [Line Items]**	LineItems			aasb1060:ReportingEntityInformationLineItems	
12	Reporting Entity [Set]	Abstract			aasb1060:ReportingEntitySet	AASB 1060 32
13	Reporting Entity Name	Concept (Text/String)	For Period		aasb1060:ReportingEntityName	AASB 1060 31 a
14	Reporting Entity Name Changes Information	Concept (Text/String)	For Period		aasb1060:ReportingEntityNameChangesInformation	AASB 1060 31 a
15	Type of Reporting Entity	Concept (Text/String)	For Period		aasb1060:TypeOfReportingEntity	AASB 1060 31 b
16	Domicile of Reporting Entity	Concept (Text/String)	For Period		aasb1060:DomicileOfReportingEntity	AASB 1060 32 a
17	Legal Form of Reporting Entity	Concept (Text/String)	For Period		aasb1060:LegalFormOfReportingEntity	AASB 1060 32 a
18	Country of Incorporation of Reporting Entity	Concept (Text/String)	For Period		aasb1060:CountryOfIncorporationOfReportingEntity	AASB 1060 32 a
19	Address of Registered Office of Reporting Entity	Concept (Text/String)	For Period		aasb1060:AddressOfRegisteredOfficeOfReportingEntity	AASB 1060 32 a
20	Reporting Entity Going Concern	Concept (Yes/No)	For Period		aasb1060:ReportingEntityGoingConcern	AASB 1060 14
21	100-Statement-Statement of Financial Position, Classified (Net Assets Approach)	Network			http://www.xbrlsite.com/aasb1060/role/Statement/StatementOfFinancialPositionClassified	
22	**Statement of Financial Position, Classified [Hypercube]**	Hypercube			aasb1060:StatementOfFinancialPositionClassifiedHypercube	AASB 1060 25 a
23	**Statement of Financial Position, Classified [Line Items]**	LineItems			aasb1060:StatementOfFinancialPositionClassifiedLineItems	AASB 1060 35
24	Net Assets [Roll Up]	Abstract			aasb1060:NetAssetsRollUp	
25	Assets [Roll Up]	Abstract			aasb1060:AssetsRollUp	AASB 1060 9
26	Current Assets [Roll Up]	Abstract			aasb1060:CurrentAssetsRollUp	AASB 1060 37
27	Cash and Cash Equivalents	Concept (Monetary)	As Of	Debit	aasb1060:CashAndCashEquivalents	AASB 1060 35 a
28	Trade and Other Current Receivables	Concept (Monetary)	As Of	Debit	aasb1060:TradeAndOtherCurrentReceivables	AASB 1060 35 b
29	Current Tax Assets	Concept (Monetary)	As Of	Debit	aasb1060:CurrentTaxAssets	AASB 1060 35 m
30	Other Current Financial Assets	Concept (Monetary)	As Of	Debit	aasb1060:OtherCurrentFinancialAssets	AASB 1060 35 c
31	Current Inventories	Concept (Monetary)	As Of	Debit	aasb1060:CurrentInventories	AASB 1060 35 d
32	Current Biological Assets	Concept (Monetary)	As Of	Debit	aasb1060:CurrentBiologicalAssets	AASB 1060 35 h
33	Other Miscellaneous Current Assets	Concept (Monetary)	As Of	Debit	aasb1060:OtherMiscellaneousCurrentAssets	AASB 1060 36
34	Assets or Disposal Groups Classified as Held for Sale	Concept (Monetary)	As Of	Debit	aasb1060:AssetsOrDisposalGroupsClassifiedAsHeldForSale	AASB 1060 35 r
35	Current Assets	Concept (Monetary)	As Of	Debit	aasb1060:CurrentAssets	AASB 1060 37
36	Non-current Assets [Roll Up]	Abstract			aasb1060:NoncurrentAssetsRollUp	AASB 1060 39
37	Trade and Other Non-current Receivables	Concept (Monetary)	As Of	Debit	aasb1060:TradeAndOtherNoncurrentReceivables	AASB 1060 35 b
38	Non-current Inventories	Concept (Monetary)	As Of	Debit	aasb1060:NoncurrentInventories	
39	Deferred Tax Assets	Concept (Monetary)	As Of	Debit	aasb1060:DeferredTaxAssets	AASB 1060 35 n
40	Other Non-current Financial Assets	Concept (Monetary)	As Of	Debit	aasb1060:OtherNoncurrentFinancialAssets	AASB 1060 35 c
41	Property, Plant and Equipment, Net	Concept (Monetary)	As Of	Debit	aasb1060:PropertyPlantAndEquipmentNet	AASB 1060 35 e
42	Investment Property	Concept (Monetary)	As Of	Debit	aasb1060:InvestmentProperty	AASB 1060 35 f

Conclusion

So, what is the point of this exercise? The point is to demystify specific aspects of what people are calling knowledge graphs and help business professionals learn about the logic systems that knowledge graphs represent.

First, knowledge graphs represent logic related to some logic system. That logic can be represented in numerous technical formats. Be suspicious of software vendors that tell you that you must use one specific technical format to represent a knowledge graph.

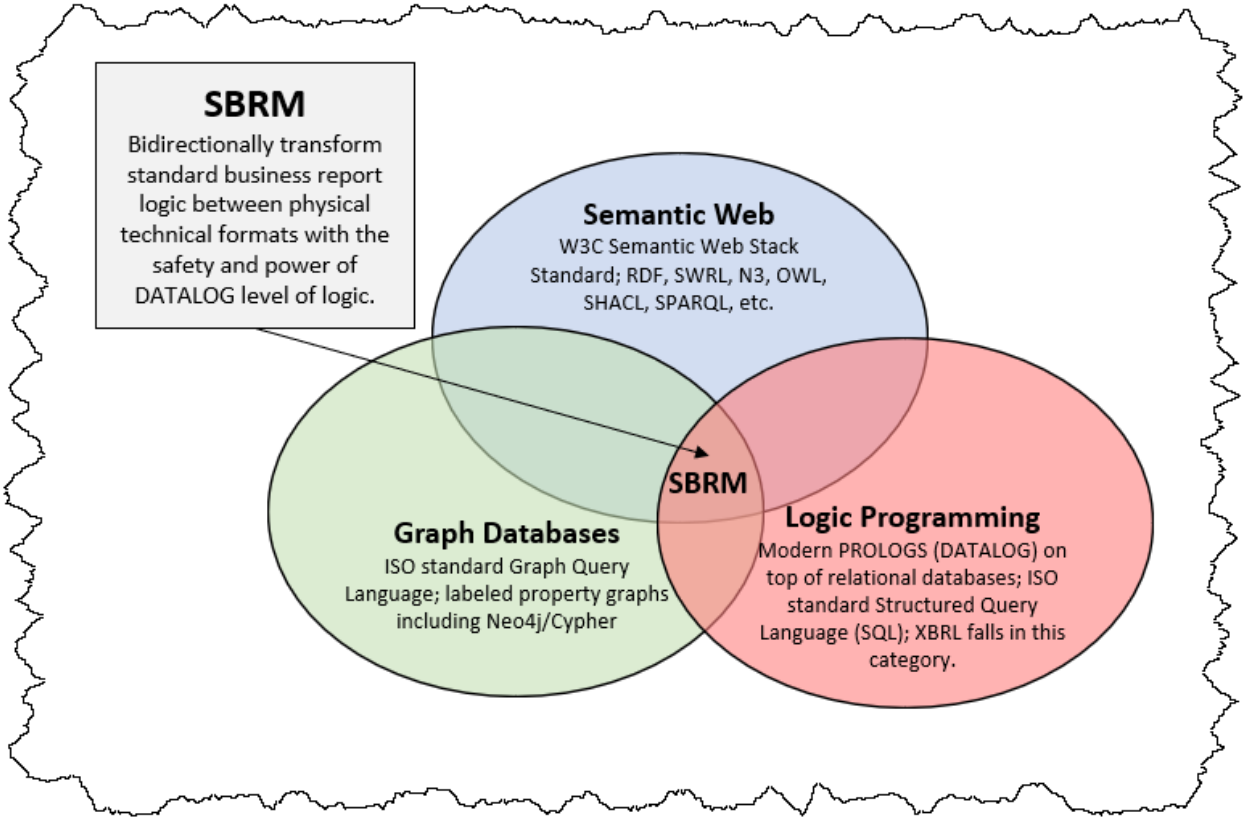


Second, a knowledge graph must be adequate to serve the goals and objectives of the stakeholders of the logic system. Remember; garbage in, garbage out. What have you done to make sure the information within our logic system is not garbage?

Third, global standards provide very significant leverage. Whenever possible, use global standards.

Fourth, there are three general approaches to implementing a knowledge graph or knowledge system. Each approach has a basket of "pros" and "cons". If any software vendor tells you only about the "pros" of their software produce, be very careful. Those "cons" exist. Software vendors what to sell you the product that they offer generally. It is the rare software vendor what will tell you that their product is not the best option for your needs and point you to another software product that is better. Such software vendors don't have your interests in mind; they simply want to sell you something they offer so that they can make money.

XBRL, the Semantic Web Stack, graph databases, Prolog, and other implementation approaches each have a set of "pros" and a set of "cons" that need to be understood and considered.



Fifth, thing of "Legos" and apply that idea to logic systems. Imagine many logic systems connected together and working together to answer questions, solve problems, and otherwise get things done. These "information bricks" will be combined into compound logic systems.

Finally, artificial intelligence and knowledge graphs offer new capabilities that were not available in the past. New modern approaches will become available. For example, modern approaches to accountancy[23]. Early implementations can be learned from and the technology can be understood. Don't try and understand the new paradigm using the map of the old paradigm; a paradigm shift has occurred (is occurring). Improve your digital proficiency[24] to make good choices.

[23] *Modern Accountancy*, https://digitalfinancialreporting.blogspot.com/2024/05/modern-accountancy.html
[24] *Digital Proficiency*, https://digitalfinancialreporting.blogspot.com/2024/05/digital-proficiency.html