1. Logic and Knowledge Graphs

This document was inspired by the paper *Logic and Semantic Networks*[1]. What I am trying to do is provide similar information that is approachable to business professionals such as accountants and use more modern terms.

1.1. Formalism

A formalism is the practice of using strict and complete methods to define and specify the important essence of a model. Every term in the formalism is given precise a definition. Every variable, parameter, rule, and factor are given a precise name and definition. A formalism stives to be complete and precise definition of the model and its mechanisms which are being described, ideally with no redundancy or gaps.

Logic and knowledge graphs (a.k.a. semantic networks[2]) are formalisms for describing information.

1.2. Information

There are specific differences between data, information, knowledge, and wisdom[3]: (note that I have added insight to match with the graphic which is shown next)

- **Data**: The basic compound for intelligence is data. Data are measures, observations, symbols, phenomenon, utterances, and other such representations of the world around us presented as external signals and picked up by various sensory instruments and organs. *Simplified: data is raw facts and numbers*.
- **Information**: Information is produced by assigning relevant meaning related to the context of the data to the data. *Simplified: information is data in context*.
- **Knowledge**: Knowledge is the understanding or interpretation, a justifiable true belief, of information and approach to act upon the information in the mind of the perceiver. *Simplified: knowledge is the interpretation of information*.
- **Insight**: Insight is the first step in putting information and knowledge to work for you.
- **Wisdom**: Wisdom embodies awareness, insight, moral judgments, and principles to construct new knowledge and improve upon existing understanding. *Simplified: wisdom is the creation of new knowledge*.

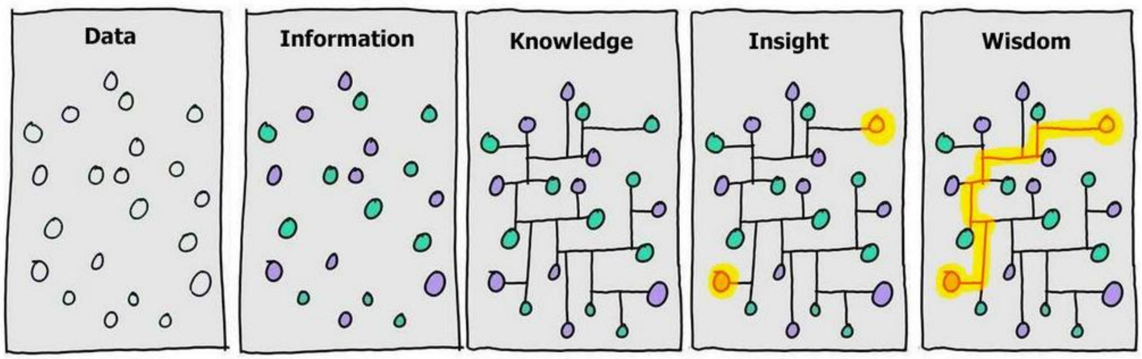
The following graphic perhaps provides the best visual explanation as to the difference between data, information, knowledge, insight, and wisdom[4] that I have run across:

---

[1] Amaryllis Deliyanni and Robert A. Kowalski, Logic and Semantic Networks, http://www.doc.ic.ac.uk/~rak/papers/logic%20semantc%20networks.pdf

[2] Wikipedia, *Semantic Network*, https://en.wikipedia.org/wiki/Semantic_network

[3] Wikipedia, *DIKW Pyramid*, https://en.wikipedia.org/wiki/DIKW_Pyramid

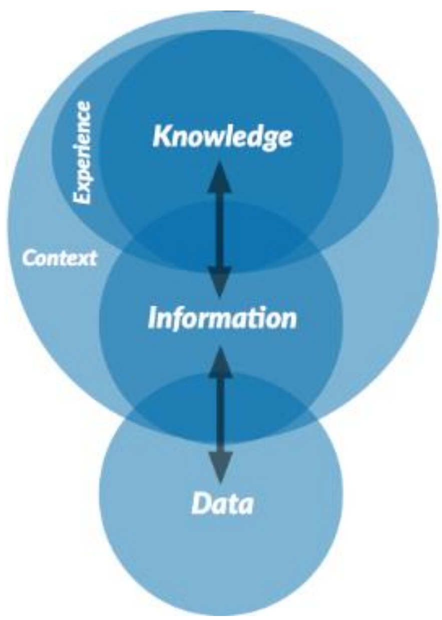


The difference between *data* and *information* is that data is the raw numbers and words where information is data in context. This is important to understand as most problems faced by accountants are an information problem, rather than a data problem. Getting data is easy. Knowing what that data represents and how the data fits together is more difficult. Representing information in the form that a machine such as a computer can understand and use that information safely and effectively is difficult.

Knowledge is a set of data and information and a combination of skill, know-how, experience which can be used to improve the capacity to take action or support a decision making process.

Insight and wisdom are related to putting information and knowledge to work for you.

The following graph created by Shawn Riley shows the important to understand differences between data, information, and knowledge[5].



[4] Tumblr, *Information isn't Power*, https://random-blather.com/2014/04/28/information-isnt-power/
[5] Shawn Riley, *Machine Learning versus Machine Understanding*, https://www.linkedin.com/pulse/machine-learning-vs-understanding-shawn-riley/

The important point to understand here is that it takes the skill and experience of human professionals to create information and knowledge and to put that knowledge into the proper context.

Our focus here is information, not data.

1.3. Logic

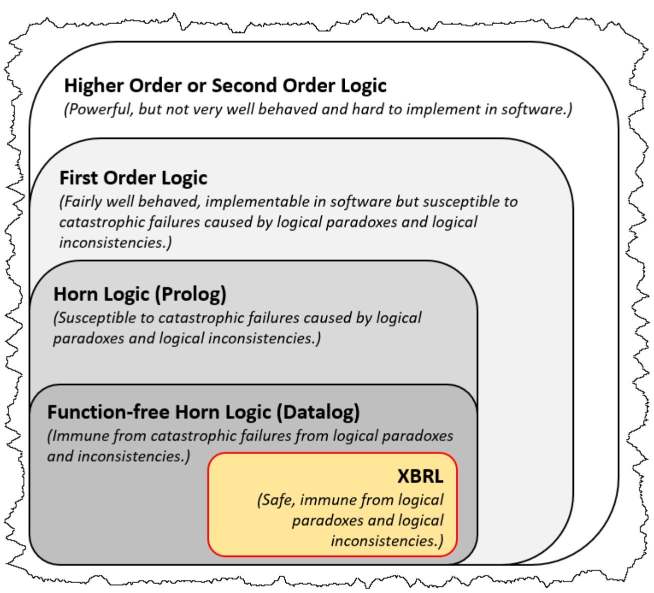
Logic is a set of principles that form a framework for correct reasoning. Logic is a process of deducing information correctly. Logic is about the correct methods that can be used to prove a statement is true or false. Logic tells us exactly what is meant. Logic allows systems to be proven.

In logic, a statement is a sentence that is either true or false. You can think of statements as pieces of information that are either correct or incorrect. And therefore, statements are pieces of information that you apply logic to in order to derive other pieces of information which are also statements.

A logical theory is a set of logical statements that formally describes some subject or system. Axioms[6] are statements that describe self-evident logical principles that no one would argue with. Theorems[7] are logical deductions which can be proven by constructing a chain of reasoning by applying axioms and the rules of logic in the form of IF…THEN statements.

A rule, or business rule or assertion, is a true statement with respect to some model of the real world that could possibly exist given some logical theory. You cannot create rules that are true in worlds that can never exist. A rule can be a mathematical expression. A rule is a type of logical statement.

Logic comes in many "flavors". The most powerful yet safest form of logic that can be implemented within software applications is DATALOG[8].



[6] Wikipedia, *Axiom*, https://en.wikipedia.org/wiki/Axiom

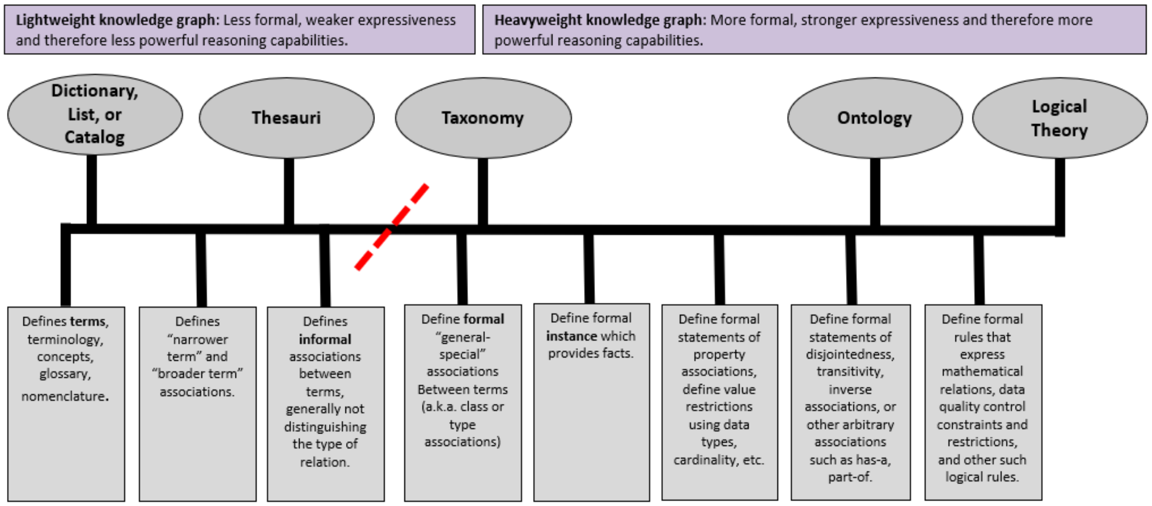
[7] Wikipedia, *Theorem*, https://en.wikipedia.org/wiki/Theorem

[8] Wikipedia, Datalog, https://en.wikipedia.org/wiki/Datalog

Relational databases follow DATALOG logic.

1.4. Tools for Representing Knowledge

There are a number of different tools that can be used to effectively represent knowledge in the form of a knowledge graph. Below you see a spectrum of such tools with the least powerful tools on the left and increasing in power to the right:



Inspired primarily by Deborah L. McGuinness, *Ontologies for the Modern Age*, Slide 4, https://www.slideshare.net/deborahmcguinness/ontologies-for-the-modern-age-mcguinness-keynote-at-iswc-2017

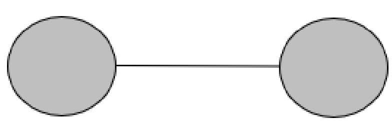
When representing knowledge, the right tool should be used for the job. A logical theory is the easiest way to enable business professionals to understand a logical system because business professionals have an innate understanding of logic.

1.5. Graphs

When I use the term graph, I am referring to the term in the context of graph theory[9] which is a discipline of mathematics. Wikipedia's definition of graph theory and graph is:

> In mathematics, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects.

This is a very simple graph:



Just like most other things, graphs have a jargon. In formal graph jargon, the circles are referred to as an edge and the line is referred to as a vertex.

Others in other areas use different terminology to refer to exactly the same idea. Here are synonyms for the notions of *edge* and *vertex*:

[9] Wikipedia, *Graph Theory*, https://en.wikipedia.org/wiki/Graph_theory

Edge	Entity	Node	Point	Report element
Vertex	Relationship	Line	Path	Association

A graph can have can have one or more paths between points; paths can have loops or cycles, circuits, as well as can have self-loops, and paths can go in one direction or both directions.

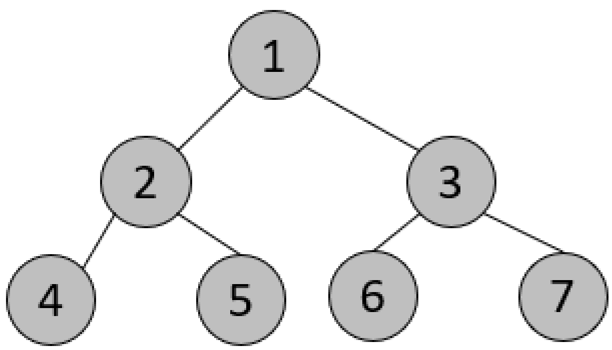
To better understand graphs, let's look at some subtle but very important differences between some different types of graphs.

If we take the time to consciously formalize the rules related to graphs and understand those rules these communications tools become more effective and they can even be understood by computer software applications.

1.6. Trees

A tree is a special type of graph. Most people are more familiar with trees than graphs. A tree is what is called an undirected graph because the items in a tree are connected by exactly one path. This is important to understand because it means that trees are safer than other types of graphs which can contain cycles. But trees have a limitation in that an edge can appear only once in a tree and a tree always has exactly one root edge. Also, because trees are undirected, they provide less information and so they are less powerful in terms of expressiveness.

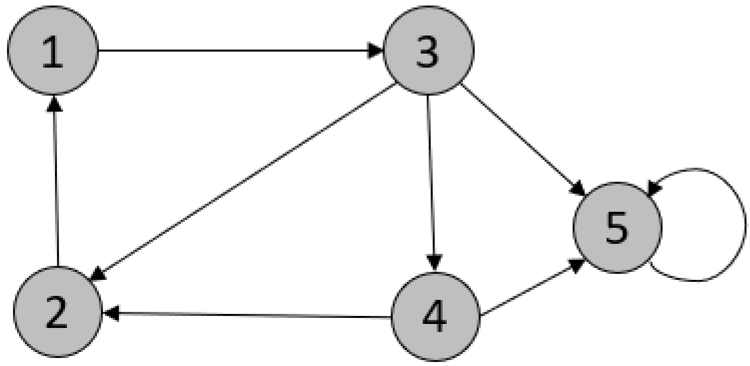
The following is an example of a graph that is also a tree:



Notice the root, node number 1 and that ever other node that appears is unique. Notice also that there is no direction associated with the lines that appear between the nodes.

1.7. Directed Graph

A directed graph is a special type of graph that provides a direction on each vertex. For example, below you see a directed graph:

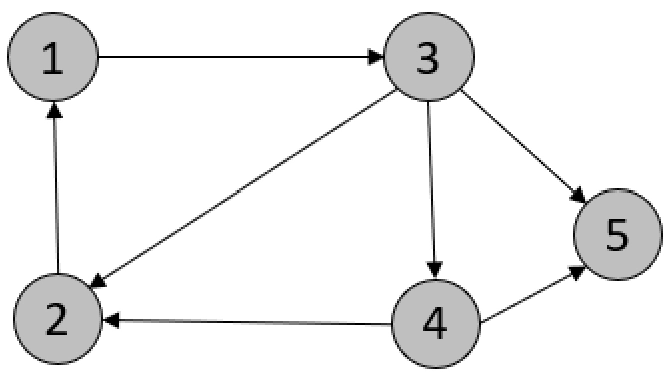


Note that each vertex (line) has an arrow that points in a specific direction; that is what makes the graph a directed graph. Note edge (node) number 5 which has an arrow that points to itself; that is a cycle. Cycles like that can cause issues such as causing an infinite loop. Those sorts of issues can be solved by using a directed acyclic graph that does not allow such cycles which we will cover next.

Directed graphs are more powerful than trees but because of the possibility of a cycles, they can be unsafe for certain things.

1.8. Directed Acyclic Graph

A **directed acyclic graph** (DAG) is an even more special type of graph that provides a direction on each vertex and you are guaranteed not to have any cycles in the graph. This makes the graph very save as there is not a possibility of creating infinite loops that can break software applications. For example, below you see a directed acyclic graph:



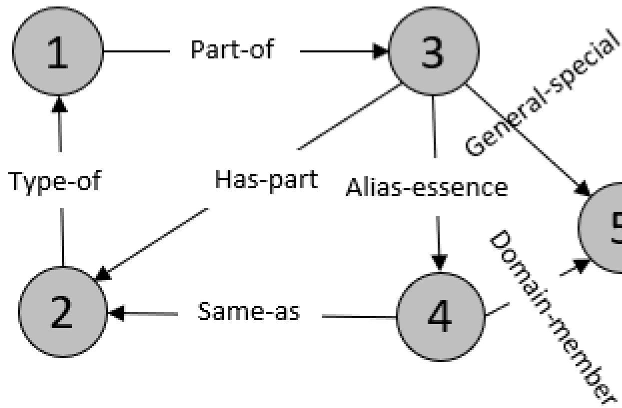
Note again that each vertex has an arrow which specifies a direction and that there are no cycles making this a directed acyclic graph. Note that there is not one specific edge (node) that can be considered the root of the graph.

But note that you don't have any information about the nature of the vertices (lines). What if information was provided about the relationships in the graph as communicated by the vertices (lines)?

1.9. Labeled Directed Acyclic Property Graph

A labeled directed acyclic property graph specifies a type of vertex for each association between any two edges. Specifying that feature, the nature of the relationship, provides additional information that is useful in working with a graph.

For example, below you see a labeled directed acyclic graph:



Note the labels that explain each vertex in the graph. You can, for example, query a graph for those relationship types. Labeled directed acyclic graphs have the most power in terms of expressiveness but are also very safe to use because they are guaranteed not to contain any cycles which can lead to catastrophic failure when read by a machine-based process.

1.10. Typed Directed Acyclic Property Graph

Now I am getting over my head, but this seems to have a profoundly important impact on functionality and query speed. There seems to be a difference between a "labeled property graph" and a "typed property graph". Also, there seems to be a critically important difference between RDF graphs and graph databases. Seems that RDF graphs are typed, but you cannot add properties. Seems that labeled property graphs are more flexibly, but that flexibility might not be needed and it impacts functionality and query speed.

This is maddeningly difficult for a business professional to understand. But, reading this article *Labeled vs Typed Property Graphs — All Graph Databases are not the same*[10] and understanding TypeDB[11] are important. Strongly typed graph databases seem very compelling. "TypeDB provides a strong type system for developers to break down complex problems into meaningful and logical systems. Through TypeQL, TypeDB provides powerful abstractions over low-level and complex data patterns."

This seems like incredibly important stuff but I don't really understand it as well as I would like to. And frankly, most software engineers don't seem to understand it well either which makes this problematic. Finally, how does something like PROLOG fit into this comparison.

---

[10] Medium, *Labeled vs Typed Property Graphs — All Graph Databases are not the same*, https://medium.com/geekculture/labeled-vs-typed-property-graphs-all-graph-databases-are-not-the-same-efdbc782f099

[11] Vaticle, Strongly Typed Database, https://vaticle.com/

1.11. Knowledge Graph

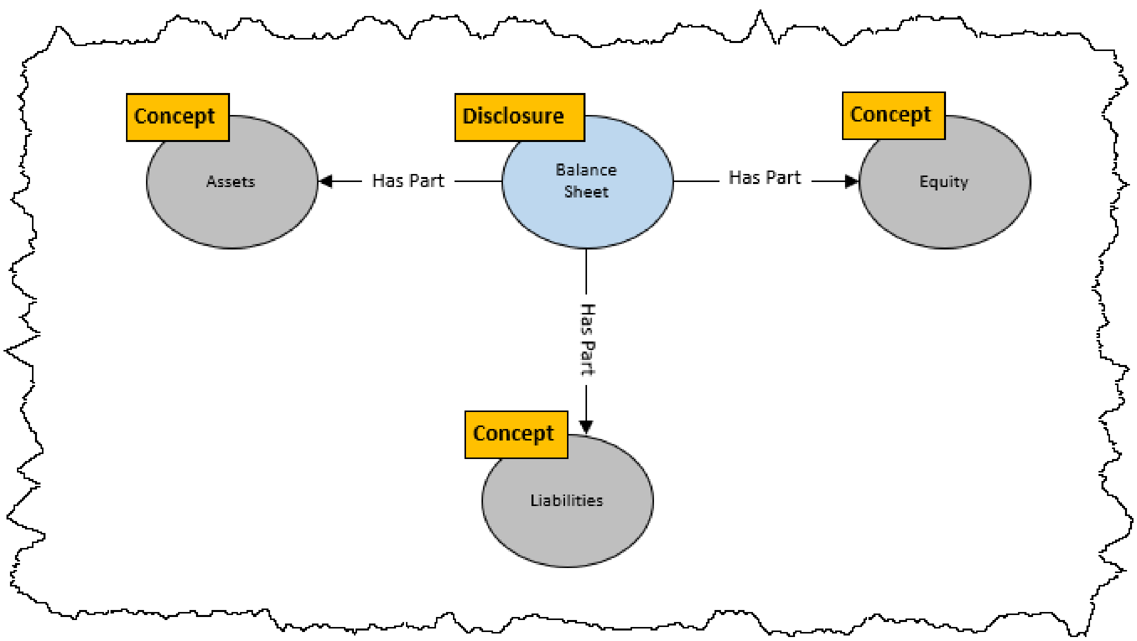
We communicate using knowledge graphs. When you go to a whiteboard and draw circles and squares and connect them with lines with arrows you are drawing a graph and communicating knowledge. Those circles, squares, lines, and arrows are intuitively understandable and very expressive. These informal knowledge graphs like this have been used by humans to communicate information for quite some time. The earliest documented use of knowledge graphs (a.k.a. semantic networks) was the third century CE.

Knowledge is the understanding or interpretation of information. Knowledge relates to terms, structures, associations, rules, facts, and skills acquired by a person through experience or education that relates to the theoretical or practical understanding of something.

A **graph**, in formal terms, is a set of vertices and edges. In less intimidating language, a graph is a set of nodes and the relationships that connect the nodes together. Graphs represent things as nodes and the ways in which those things relate to one another and rest of the world as relationships.

A graph is a general-purpose communications tool that allows us to model all sorts of scenarios in terms that are innately understandable to humans. One thing that can be represented in the form of a graph is knowledge.

This is a simple graph of knowledge, or a **knowledge graph**:

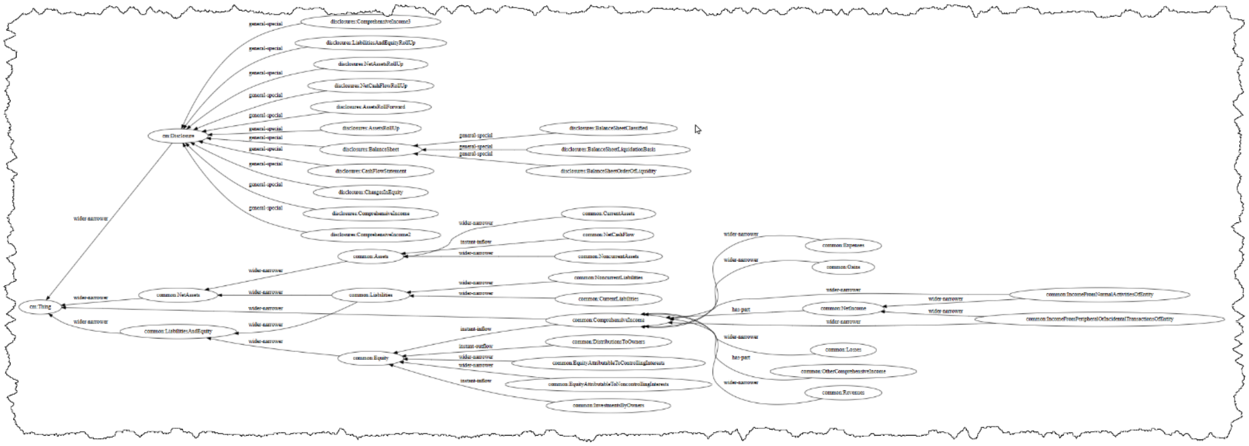


A knowledge graph[12], also known as a semantic network, represents a network of real-world things (entities)—i.e. objects, events, situations, or concepts—and illustrates the relationship between them. This information can be visualized as a graph structure.

[12] IBM, *What is a Knowledge Graph?*, https://www.ibm.com/topics/knowledge-graph

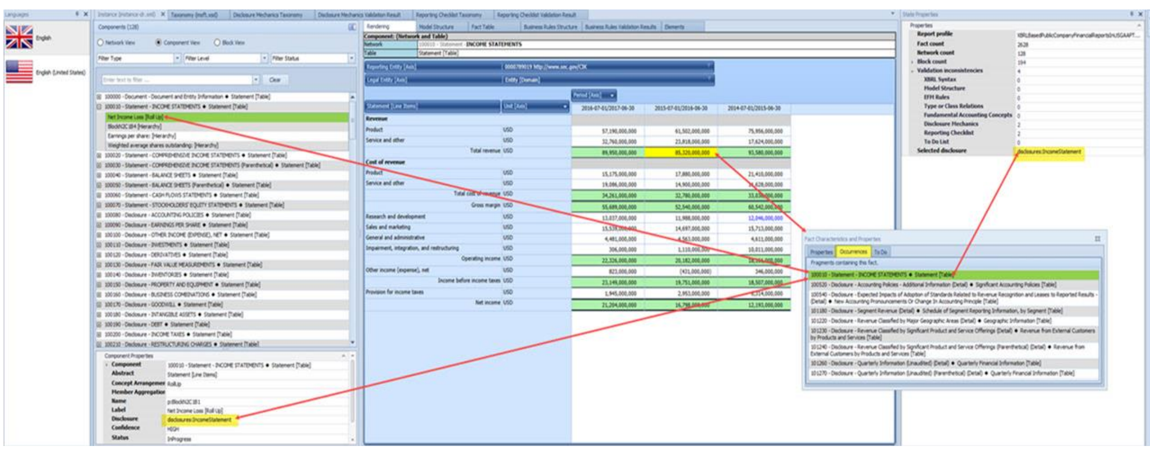
1.12. Example of Knowledge Graph

The following is an example knowledge graph that explains accounting information related to the four statement financial statement model[13]:



1.13. Financial Report Knowledge Graph

As explained in the document, *Financial Report Knowledge Graph*[14]; another example of a knowledge graph is a financial report. For example:



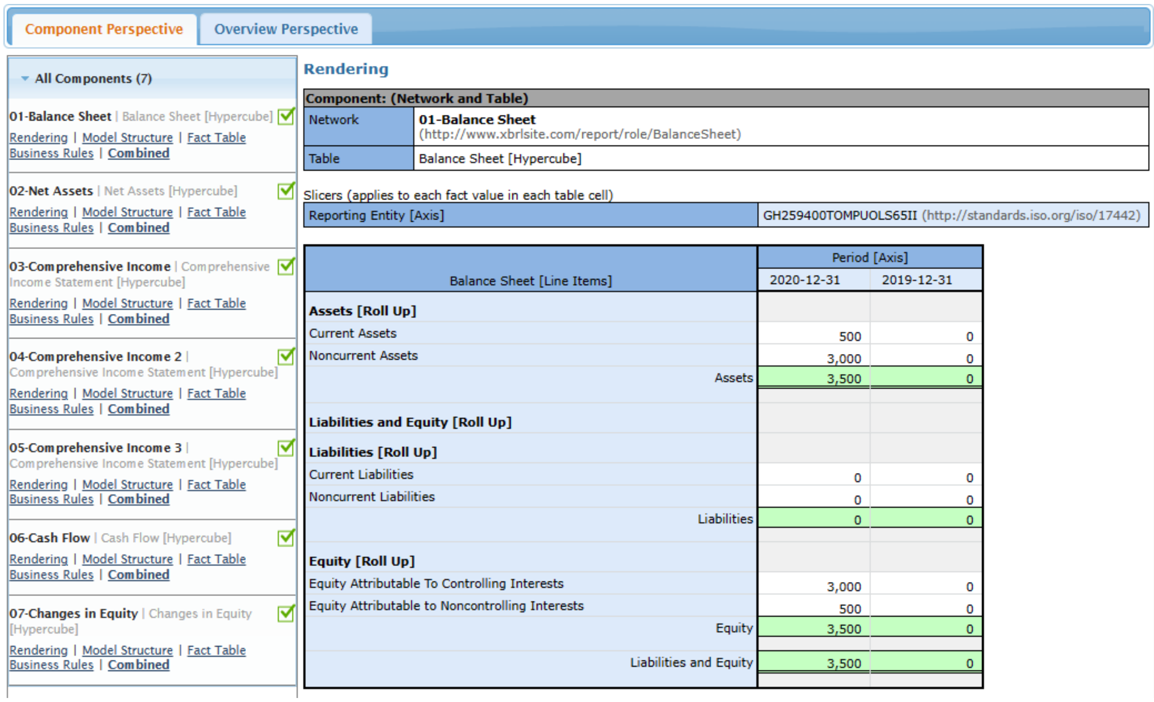
Here is an example of an XBRL-based digital financial report that has been converted into an HTML human readable rendering of that machine readable knowledge graph[15]:

---

[13] Auditchain, *Four Statement Model Knowledge Graph*, https://auditchain.infura-ipfs.io/ipfs/QmTtcz3rcEmsSYjkev3Xo9qHH2YUrnkGimyt9Wg2YV3Lr2/typeSubTypeGraph.html

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

[15] *HTML rendering of financial report knowledge graph*, http://www.xbrlsite.com/seattlemethod/golden/common2/reference-implementation/evidence-package/



Component Perspective
Overview Perspective
All Components (7)
01-Balance Sheet | Balance Sheet [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
02-Net Assets | Net Assets [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
03-Comprehensive Income | Comprehensive Income Statement [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
04-Comprehensive Income 2 | Comprehensive Income Statement [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
05-Comprehensive Income 3 | Comprehensive Income Statement [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
06-Cash Flow | Cash Flow [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
07-Changes in Equity | Changes in Equity [Hypercube]
Rendering | Model Structure | Fact Table
Business Rules | Combined
Rendering
Component: (Network and Table)
Network
01-Balance Sheet
(http://www.xbrlsite.com/report/role/BalanceSheet)
Table
Balance Sheet [Hypercube]
Slicers (applies to each fact value in each table cell)
Reporting Entity [Axis]
GH259400TOMPUOLS65II (http://standards.iso.org/iso/17442)
Balance Sheet [Line Items]
Period [Axis]
2020-12-31
2019-12-31
Assets [Roll Up]
Current Assets 500 0
Noncurrent Assets 3,000 0
Assets 3,500 0
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 3,000 0
Equity Attributable to Noncontrolling Interests 500 0
Equity 3,500 0
Liabilities and Equity 3,500 0