XBRL as an Information System

The captain calls the engine room.

- *How much?*
- *Thirty!*
- *Thirty what?*
- *How much what?*

Reliable and transparent knowledge sharing is essential to operate complex systems, especially when the participants' expertise is different and sometimes even their motivations are conflicting. Accounting aims to solve this situation for thousands of years. Today, it can also utilize huge computer networks, the affordable storage, process and communication performance grew exponentially. Unfortunately, this does not automatically translate to better transparency. The ability to access millions of pages of plain or beautifully presented reports does not mean their content is valid, coherent and comparable. *The key is how we transfer our knowledge into computer systems*, considering the benefits and limitations of this process.

In this chapter, we will overview the fundamental concepts of this transfer process. The definitions will not be scientifically precise or follow very complicated research areas, but clear enough to give a coherent path from basic elements like "data" to all terms in any XBRL report. These terms will be defined "as we use here".

Transferrable Knowledge

A good analogy is photography. The traditional analogue photography uses a film that is sensitive to the light that makes molecular changes in the material during the exposure, then the film is developed and fixed. The resolution of the image is very high you can magnify very small parts of it. On the other hand, the physical object is the only one instance of that picture, you must repeat the procedure to create another copy; manipulations like trimming, storing or transferring mean physical operations. Although *an analogue photo is stored knowledge but is not "transferrable"*.

Digital images are different. They are just a long array of *numbers*, one of more of these numbers describe properties of a *pixel*, a huge matrix of these pixels is the *image* that can appear on your screen. *The resolution is fixed when you create the image*, when a digital camera converts the lights to this long array of numbers, you can't "magnify" it. On the other hand, *you can store, copy, edit, transfer this digital form without any physical operation*. You can use algorithms to change the values to sharpen, lighten, ... the picture, other algorithms can combine hundreds or even millions of pictures.

A less apparent property is that *you always need additional knowledge of how to organize and use the digital form* to create the image: how many dots are in one row, how many numbers describe a dot (one for a black and white image, 3 for an RGB, you may have another for transparency, etc.). It is even possible that those numbers have no connection with properties of light, like the artificially colored images of space probes.

An extreme example is a 3D MRI scan. The slice images are not created by recording light but reverse engineered from the measurements of a rotating sensor. It creates several images from the target area while advancing with a slow known speed. Then 3D algorithms combine the slices, identify surfaces of the organs or find blood vessels to gain situational awareness or even plan a surgery process and allocate resources.

Summary:

- When you use computers for knowledge management, you must consider that it is a fundamentally different, digital platform.
- The result of the transfer process is limited, less than your original knowledge: you lose all attributes that you don't transfer; you initially decide the granularity and lose all finer details; the digitized values have a finite precision and lose the rest.
- The exact transfer rules are necessary to regain the transferred knowledge together with the content, and it will still not be the same as your original knowledge before the transfer.
- However, the transferred knowledge allows digital storing, replication and algorithmic processing – none available in the original form.
- You can create a "familiar form of knowledge" from very different sources, like an image of your organs or a mathematical function.
- You can combine huge amount of transferred knowledge using its digital form and reliably extract knowledge that was not present in any item but "among" them, like an image on a vertical plane from a hundred horizontal slices.

Information System

Information systems manage transferrable knowledge. They contain

- their own terminology,
- rules to transfer a real-life situation into a model that represents it using the local terminology with a given precision,
- algorithms that allow to validate, analyze, predict or manipulate this model.

In this way, *information systems help their users to gain situational awareness and interact with complex "target" systems*. Creating the proper terminology is research (highest cost); then you need education to teach people to use them; they must use their knowledge to create and use these models. But when you fly an airplane, manage a factory or a hospital, there is no question: proper information systems are necessary to optimize the processes and avoid disasters.

Information systems in general have multiple layers, with different knowledge segments. Following the previous analogy, an information system transfers video streams of the plant security cameras to an office – this system consists of hardware and software components. The security officers know the layout of the plant, assume how people move from one place to another. They also know the "normal activities" at each location, the uniforms and even faces of workers; or look up schedules to verify if a known person should be at a given location or not.

The abstract nature of an information system extends its capabilities beyond of the represented real-life system. You can store the lowest layer of primary knowledge, sometimes with a lower resolution, like saving the snapshots of each camera in 5 seconds or keep the voice conversations for a month. This allows *"rolling back" or "speeding up" the time* within the information system to understand the events that lead to a disaster or realize opportunities for optimization, like change vehicle distribution or relocate depos to decrease waiting time or empty movements. The analysis can be centralized, the results can be reused, like aviation accident analysis and recommendations – *breaking the limits of physical distances*.

Information systems are not just software and not even have to be digital to achieve this. If you know the language and the current notation system of accounting, you can get an equivalent picture of the business of a medieval Venetian bank, the Osman Empire or Egypt. Of course, you will not know the color, the weight or the temper of the listed animals, the smell of the crops or anything about the people and vehicles that transported them. The model does not store that and you can't "magnify the image beyond the original resolution". However, you may detect someone cheating after transferring many years of books into digital form and run algorithms that can detect trends and find outliers. On the other hand, *any misunderstanding in the language or the notation system will lead to misinterpretation of the stored knowledge*.

The Big Picture of XBRL

XBRL (eXtensible Business Reporting Language) is an information system to create a transferrable knowledge "snapshot" of "any" business activity. The first version of XBRL was created in 2000, the 2.1 version in 2003 and stable since then, minor modifications to correct typos were published 20th February 2013. XBRL is the de facto global standard of financial reporting where the required content and structure of such reports is updated annually. It is the official standard to publish ESG (Environmental, Social and Governance) reports and used in many other areas. How can such an "old" standard be used both in a continuously changing and totally different environments?

The answer is that *the XBRL standard consists of two information systems*. *The first contains the terms and rules of creating "any" report*, which is a long list of "facts", each is a numeric, date, text, etc. value of a "concept" for a given "moment or period". The "concept" can be the income, share price, etc., but their definitions can be found in an external "taxonomy" to which the report only refers to. *The second information system allows defining a taxonomy*. Each taxonomy is a complete information system: it contains terms, presentation and validation rules, calculations.

So, in the case of US financial reports, the attributes of your transferrable business knowledge are defined in the US-GAAP taxonomy, which is updated annually according to the current requirements. However, your XBRL reports are always the same: they refer to the actual US-GAAP taxonomy version, and the list of facts refer to the concepts defined in that version. A software that handles all US reports just loads the referred taxonomy and handles the facts according to that description. If you happen to report to the EU, you should use the current version of the ESEF taxonomy, but the software works the same way, only now loads the referred ESEF taxonomy.

The role of the XBRL infrastructure and services is similar to the standard shipping container. The report layer ensures the external properties: file formats and general structure that allows basic validation, homogeneous storage and general content-related services – regardless of the actual content. The taxonomy acts like a standard documentation about the content: receivers can describe what they want to see in the container, from simple labels to mandatory content or even describe the requested relation among the labeled items in the container. This description is transparent to the XBRL handler infrastructure, thus general services can use the taxonomy to peek into and validate the content of the container like inspectors at a container station or port.

This means, a proper XBRL handler software can validate the report according to the required taxonomies while edited by the publisher, ensuring that only valid reports enter the system. The receiver can assume that all facts in all incoming reports contain only properly labeled facts and the report passed all validations in the taxonomies. When a taxonomy is updated via its description, the XBRL infrastructure will load and apply the new definitions, therefore it will adapt to the changes automatically for the new reports. A decade old report still refers to and can be validated against a decade old taxonomy.

Regardless of all its flexibility, XBRL cannot avoid the fundamental weakness of dealing with transferrable knowledge: all attributes, all possible values, etc. that you want to store in the report must be defined in a referred taxonomy. In practice, an American company creates a yearly report for the SEC. Most of the finance-related facts refer to concepts defined in the US-GAAP taxonomy, but that does not cover company identification, that comes from the DEI (Document and Entity Information) taxonomy. Similarly, the company itself, an alliance, a country, etc. may also create taxonomies. The publisher may decide to create one report and send it to several authorities. The XBRL standard supports this attitude, each receiver can use the segments (defined by the taxonomies) that they are interested in.

The task of a taxonomy creator organization is also hard. For example, the FASB is responsible to maintain the US-GAAP taxonomy with the aim to make the US economy sufficiently transparent. This requires defining all concepts, possible values, add all mandatory flags and validation rules to all "interesting" facts, so that companies will report and analysts can use them. Although anything that is not defined in the US-GAAP, may still appear in the report under additional (like, company or state level) taxonomies, but that will remain invisible for analysts using the US-GAAP. As a result, all such "primary" taxonomies tend to grow significantly over time and their creators require full control over them. Also, there are parallel organizations, like IASB creates the IFRS taxonomy for the same goal, the EU extends IFRS and creates the ESEF taxonomy, etc. The same applies to the ESG (environmental, social, and governance) taxonomies.

The situation may seem grim but still, XBRL is transferrable knowledge: through the taxonomies every value in every fact has an objective definition. If we provide translation rules from one taxonomy to another ("concordance"), the XBRL infrastructure can convert a report from its primary taxonomy to another one (also indicate the conversion errors or missing rules). FAC (Fundamental Accounting Concepts) by Charles Hoffman is a generic financial taxonomy over US-GAAP and IFRS with translation rules from them to FAC. The benefit of this approach is that accountants can start the general audit process using the FAC "view" of the report, but "drill down" to the underlying primary layer (US-GAAP or IFRS) when needed.

Knowledge Layers in XBRL

The cumulative knowledge represented by XBRL reports is a complex information system. As vaguely explained above, it has different knowledge layers, each must be properly managed by the XBRL infrastructure (and therefore, various participants like software developers, taxonomy managers, report creators, collectors, analysts). To understand its operation and use XBRL properly, one must be aware of all layers in sufficient details for their own tasks.

The following sections will describe the layers enough for a reliable, general picture. You will be able to look at any XBRL report through various tools, identify its segments, the connections of the elements, what such a report can and what it cannot accomplish. You would need much more details to manage a taxonomy or write XBRL software, but you will have a general understanding of their roles and, if necessary, explanations.

0. Abstract and simplified terms for Transferrable Knowledge.
1. The terms in the XBRL “report layer”. The definitions will be precise, focusing on understanding what they mean and where you find them in an XBRL report.
2. The “primary taxonomy layer” like US-GAAP, used to create an XBRL report. What it contains, how it affects the structure and content of the report.
3. The general content and aim of a “secondary taxonomy”, used to homogenize and analyze reports beyond the primary taxonomy, explained by the FAC.

Layer 0: Fundamental Terms of Transferrable Knowledge

This is the abstract area of ontology, semantic networks, knowledge graphs, etc. Each have their own, sophisticated terminologies but instead of going into any of them, we will only define a few “keywords” as they will be used in the subsequent layers.

Data

Data is the “atom” of transferrable knowledge, an unambiguous piece of information. It can be of various types, here are the most common ones with their specialties.

- **Numeric** data is obviously, a number – but not only that.
  When you transfer any numeric value, it is stored with a given precision using a given number of digits. You may think that a fraction like 1/7 or a constant like PI is exact, but that only hides the limitations of the actual computer that you use. In XBRL, you provide the precision of a numeric value for the validation rules.
  In most cases, the value represents a numeric attribute, given in a measurement unit. Without the unit, the data is ambiguous. In XBRL you often see the “shares” unit and currencies like “USD” or “EUR”, or a combined unit like USD/share.
- **Identifier** data is a string that identifies a knowledge item.
  We use identifiers to refer to a knowledge item, therefore, this string must be unambiguously assigned to exactly one item (not more), and whenever an identifier string appears, the referred item must be available. In a modular system, identified knowledge items come from different sources and although they assign unique strings, they may collide when using them (multiple sources defined the “green” identifier).

We use “namespaces” to avoid identifier collision, an identifier can be local: “localIdentifier”, or external: “namespace:identifier”. In an XBRL report, you will both see local identifiers like “UsdPerShare” or “Fact_1234” and external ones like “us-gaap:Assets”, “ifrs:Assets” (note the avoided name collision) or “ISO4217:USD” (dollars).

- **Category** – when the value can be one of a list, like color is green / blue / …
  To avoid ambiguity, the possible values must be properly defined knowledge items and the value is a reference to an identifier. This is how you refer to US dollars by the “ISO4217:USD” identifier.
- **Date** – or anything you could display in a calendar.
  Storing dates can also be complicated. Date can be represented by a single number (in Unix, the time elapsed since January 1, 1970.) numeric or alphanumeric strings according to a date format template. As mentioned before; to retrieve the original knowledge, you need the rules you used to store it.
  XBRL generally uses the ISO 8601: “2015-10-30” for dates or “2015-10-30T08:00:00.000Z” for time values.
- **Text** – plain or formatted character streams: names, addresses, descriptions.
  With free texts we can store and share knowledge, but this is not “transferrable” as it requires a human to understand it. The “unambiguous” nature of text data only means that the information system stores it “as is”. The system can use algorithms from simple search and comparison operations to training Large Language Models (LLMs) – but this is still not “transferred knowledge”.
  Texts have an additional parameter, the language, in XBRL it can appear as xml:lang ="en-US”. This allows separation of texts by the language or automatic translation to a target language.
- **Stream** – binary stream of various types: images, audio, video, documents, etc.
  It works like texts: the system must know its type and type-dependent attributes, can apply algorithms on it, but does reliably “look into” them for knowledge. For example, you can upload an image of a person, but it is ambiguous, not “the” image, you can upload hundred more images of the same person. However, the height of the same person is unambiguous, e.g. 175cm or 68.9 inches, the same measurement in different units.

Attribute

The attribute is an identified placeholder of a Data with specified type, it provides a context to a Data. In other words, in transferred knowledge, Data appears as “the value of an Attribute”. Its type must match the type required by the Attribute, you can’t provide a Text data to a Numeric attribute, or in the case of categories, the identifier of the English language as the value of a Currency.

In XBRL, us-gaap:Assets and us-gaap:Liabilities are both attributes of “Monetary” type, thus a company can report the Data 1000 with unit ISO4217:USD for us-gaap:Assets and 500 with unit ISO4217:EUR as us-gaap:Liabilities. The dei:DocumentPeriodEndDate attribute is of “date” type, and can have the value of 2023-12-31. This example also demonstrates that Attributes have external identifiers and in almost all reports, you will find attributes from multiple sources (taxonomies).

Most attributes are singular, can have one Data as value, like the examples above. Other cases the value of the attribute is a collection of Data. An XBRL report contains a collection unit and context definitions, and a relatively large collection of facts.

Knowledge Item

In a general information system, we use knowledge items to represent an entity in the real world: a person, a classroom, a book, etc., in this case, we often use the term “digital twin”. However, the knowledge item is not necessary to be a twin of something real, like the buildings and airplanes in a computer game; or even less tangible (you will see the XBRL Context term as an example of that).

So, when you transfer your knowledge, you collect data as values of attributes to create “virtual property sheets”. These tables are the knowledge items, each “row” is an attribute - value pair for this item. Sometimes you give an identifier to this property sheet to refer to it, e.g. in XBRL, you define the measurement units by filling up their property sheets, assign unique identifier to each and refer to them later in the report. Sometimes you don’t and can only search for the knowledge items by their content (values of specific attributes) finding zero, one or many.

Aspect

This is a radical difference from the traditional top-down knowledge representation approach. There, you define a hierarchy of “types” or “classes”, the attributes are their members, and you must enroll your knowledge items into them. You must know this classification before starting the transfer process and it is hard to change later.

The bottom-up approach is that we define Aspects as a collection of mandatory and optional Attributes. If a Knowledge Item has all the mandatory Attributes of an Aspect, then it “belongs to it” and can be handled accordingly. For example, the Location Aspect requires Latitude and Longitude Attributes; any Knowledge Item with a Location Aspect can be displayed on a map. An entity in an XBRL report does not have such Data but may have a state or even address. An algorithm can generate Latitude and Longitude, the entity now “gained” the Location Aspect and appears on the map.

In a similar way, all XBRL reports are submitted using a “primary taxonomy”, like US-GAAP or IFRS, knowing nothing about the FAC taxonomy. However, we have concordance rules that translate the primary values and add FAC Attributes to the Knowledge Item “property sheets”. In this way, these items “gain” Aspects defined in the FAC taxonomy, thus the related rules can validate, the user interfaces display them.

Knowledge Unit

A Knowledge Unit is a network of closely related Knowledge Items. In XBRL, Knowledge Unit is an XBRL report, or all versions of the US-GAAP taxonomy. The latter is an important note: an XBRL report refers to the then-current version of the US-GAAP taxonomy, the infrastructure must find and connect the right version.

Summary

So, what do you do when you create an information system and transfer your knowledge into it? (You may select a topic and try these steps as a practice.)

1. Create an attribute set that you want to collect. Some of these attributes are your own, some come from an external source; some may be mandatory. You ensure that the attribute naming scheme ensures safe, unique identification.
2. Check the data types that you will collect. If you have numeric values, you look for the measurement units; with texts you identify the languages you want to use; for categories you define or import the identifiable items you want to refer to.
3. Identify the Knowledge Units that you want to create, select a proper format for them. (Hint: in simple cases, you may find a spreadsheet quite sufficient.)
4. Formalize your knowledge by creating "virtual property sheets" of Knowledge Items, by storing the value of their local attributes (numbers, dates, texts, ...).
5. Assign identifiers to those that you want to refer to and use them in the attribute Data cells to register such references.
6. Store the registered Knowledge Items into identified Knowledge Units.

Congratulations, you have created a "snapshot" of your knowledge in a transferrable form!

If you used a digital platform to do this, you can easily create new snapshots by repeating step 4-6. With proper segmentation of the knowledge layers, you may even change the attribute definitions among snapshots without needing to modify the already created ones.

The XBRL standard provides you with all preliminary steps: you get the attributes (and many other services) from the taxonomies, you know what Knowledge Units you will create and what formats you can use. You also benefit from decades of professional application and countless existing XBRL reports. You only need to understand the fundamental concepts of the standard, explained in the next sections.

Layer 1: General XBRL Terms

Although it may be tempting, please do not open an XBRL report at this time. The standard allows multiple syntaxes (XML, JSON, iXBRL and CSV), all can store any XBRL report but support different use cases with slightly different internal structure. The next sections will explain the abstract terms, then build up the structure of an abstract XBRL report and finally point out the most important differences of the syntaxes.

The goal here is to prepare you to understand, look for information in an XBRL report and to confidently use an XBRL tool – you will need an XBRL reference to manually edit a report.

Contrary to the previous section, now we start with the top level and move down to the smallest details.

Entity

The Entity is the organization that uses XBRL to report about its operation.

As learned above, an entity is a Knowledge Item with a unique identifier that we can use to refer to it. In the US, the U.S. Securities and Equities Commission (SEC) assigns a unique Central Index Key (CIK) to the registered entities, you can use this identifier when sending a report to the SEC. It is important to know that CIK is not the only possible identifier, in the EU the Legal Entity Identifier (LEI) is generally used, S&P also provides a global identifier, etc.

The same company can be registered at multiple organizations and use their identifiers. The conclusion here is that the external identifier “cik:0000021344” or “lei: 57VG5X0E00X0QJU7CQ58” are apparently different identifiers but both refer to Coca Cola Co. However, each report collector organization states the preferred entity identification scheme.

It is easy to see that in the XBRL information system, the same Entity can publish multiple Reports (various report types over multiple years). The XBRL syntax allows the opposite: one Report could contain knowledge about multiple Entities. However, this is generally a validation error and you should refer to only one Entity in a Report.

Report

In abstract terms, an XBRL Report is a Knowledge Unit, a collection of coherent and valid Knowledge Items that together form a “snapshot” of the operation of one Entity, according to the requirements of an authority, like “a properly filled 10-K report of Coca Cola Co. for the 2023 business year”.

Technically, the report itself can be one XML, JSON or XHTML file, or one JSON index and a set of CSV files (in the CSV syntax). However, as the XBRL allows you to create and use your own taxonomy – if you used this option, your taxonomy is part of the report.

The abstract content of a report:

- Reference to external Knowledge Units (definitions of the XBRL standard and the primary taxonomies).
- Identifiable Unit definitions (used by the numeric Data).
- Identifiable Context definitions (from analysis perspective, these are the primary Knowledge Items of the report)
- Facts (you can think of them as the rows of the property sheet of a referred Context: a valid Attribute – Data pair in that Context).

Unit

Measurement units are typical examples of “well-known things” – and just like Entity identification, that can be a problem, to ensure that all reports refer the same way to the same thing.

In XBRL, there are some internal pre-defined units, like xbrli:shares to indicate the number of shares, or xbrli:pure that indicates no measurement unit. XBRL recommends common standard sources for externally managed units, like ISO4217 for currencies.

You can find simple units like currency for Attributes like us-gapp:Assets, or combined ones like currency / shares to store the price.

So in the XBRL report you collect all possible measurement units, assign a local identifier to them that you will use with all numeric Data. You refer to the actual units via the recommended external identifier like ISO4217:USD. The benefit of this approach is obvious already in one report: you can simply and reliably collect all Data given in dollars, check if and where you used other currencies, etc. When dealing with multiple reports, you can rely on the standard external identifiers, you will find all Data reported in USD in all reports.

Context – primary

The Context is the key term in the XBRL report and a bit more abstract than the others.

As you recall the definition of the Knowledge Item, the natural scenario is that it is a “digital twin” of an external entity (building, vehicle, person), or the primary form of a virtual entity (like vehicles in a game) – but sometimes it is not. In these cases, the Knowledge Item is just a virtual property sheet that we create following our knowledge transfer needs. Context is such a Knowledge Item: a locally identified collection of attributes and their values, related to an entity and a given time instant or period.

As you already know, each Report contains one or more locally identified Context definitions. “Context A” contains data of Coca Cola Co. at 2023-01-01; “Context B” collects data representing the continuous operation of Coca Cola Co., from 2023-01-01 to 2023-12-31; “Context C” is the snapshot of Coca Cola Co. at 2024-01-01.

The Context refers to one Entity, so technically one Report could contain Context definitions referring to different Entities. This is prohibited, all Context references must point to the same Entity, using the same external identifier. In practice, this is not a limitation, the report is created by the Entity who is responsible for its content, there is no reason to report anything from another Entity. On the other hand, the syntax of the XBRL standard allows collecting Contexts from different Entities into a single Report – an XBRL summary of year 2023 from all beverages companies is technically possible.

The flexibility of the Context time allows store knowledge in finer granularity: you can create monthly snapshots or week-based periods, there is no technical limitation. XBRL reports generally contain both summary Contexts with many Attribute-value pairs for official milestones (quarters, years), and smaller or more technical details like some data on weekly operation.

This approach has an important technical benefit: you find all time information in the Context definitions. It is easy to check if a report contains anything about an arbitrary moment or period and collect those Data – even if you must deal with thousands of reports of different types.

Context – dimensions

With primary contexts you can change the “resolution” of transferred knowledge in time. But what if you want to increase the granularity by a different factor? You may want to segment your knowledge related to transport activities by states or regions, marketing performance metrics by age groups, split travel data to road, rail, air, …

You may create Attribute "clones" for those segments, but that practically multiply these sets while all such clones represent the same information, just for a different segment. Imagine that you want to store the values of the travelDistance Attribute of your company for 10 years in an XBRL report. You create the related 10 primary contexts; all refer to your company for the subsequent years. You can draw a bar chart displaying these values. Now, you want to separate Rail, Road, Maritime, Air. You can create Attributes like travelDistanceRail, ..., but later you may need travelCost, etc. separated the same way. Later you may want to split by "Private" and "Business" – yet another multiplier… not convenient. What you really want is just to split those bars. You want to keep the bars of the total chart, create a split chart by coloring parts of the big bar by the means, or create a chart displaying Rail only. All while collecting the same, single Attribute, travelDistance, for different… sub-contexts.

XBRL provides an elegant solution to this problem: you can add these factors to the Context definition as optional, custom Dimensions. You already have the primary Context that you used to collect data for your company by year, perhaps many others apart from travelDistance. Now, you create new contexts, the company and the year is the same, but you add the "TravelMode" custom dimension and the different "Rail", "Road", … values for each. You can use these contexts to collect travelDistance independently, and use them as you like, e.g. drawing those charts. The "virtual property sheets" of these "finer" Contexts will likely be much smaller, but you can still collect any Attribute that you want to store in this more granular form. Similarly, if you want to further split these Contexts to Private and Business segments, you can add that too, so you create a Context where TravelMode = Rail and TravelExpense = Business, etc. Of course, a Context can contain only one of the possible values for each Dimension.

With custom Context dimensions, XBRL gives you a fully flexible environment to store your knowledge in theoretically any granularity (within practical technical and size boundaries). This structure allows you to use the same Attribute set that you use in the primary contexts. Adding new dimensions changes a 1-dimensional list to a 2D grid, a 3D cube, etc., the formal name of this structure is "hypercube" (that you will find in the XBRL reference). This results a huge "theoretical cell count" but you only create the Contexts for those that have any value in it, like you would do in real life: create a property sheet in your catalogue only when you want to write something on it.

Fact

Obviously, the main goal of any XBRL report is to contain a lot of Facts. The previous terms finally allow us to define the structure of a Fact and explain how they are handled. In simple terms, a Fact is a "row" of the "virtual property sheet" of a Context, a properly given Attribute – Data pair. It is also apparent that this is a bottom-up structure, a simple list of Facts populates the complex Context hierarchy of the report by referring to them.

A Fact consists of the following elements.

- Context reference (mandatory local identifier): connects the Fact to exactly one Context; the Attribute – Data pair belongs to that Knowledge Item. As you know, all the complex structural details (entity, time, dimensions) are in the Context definition, the Fact does not contain such information.

- Concept reference (mandatory external identifier): identifies an Attribute definition as “taxonomyName:conceptId”. The XBRL report assigns the local taxonomyName to an actual Taxonomy when imports it, and you will find the conceptID in that Taxonomy.
- Data: the value of that Attribute in the referred Context. As you recall from the definition of Data, in some cases, you must provide additional information depending on the Data type to avoid ambiguity.
  - For Numeric Data, those are *Unit reference* (local identifier of a Unit definition), and *precision* by giving the number of decimals to consider: value of 2 means 1.23xxx, -6 is 123,xxx,xxx.xxx (rounded to million). INF means “infinite precision”, the value must be handled as is.
  - For Text Data, you should provide the *language*.
  - Identifiers (references and categories) must be resolvable.
  - Date values most be in the ISO 8601: “2015-10-30” format.

Of course, there are less apparent factors that are also required to have a valid XBRL report; an XBRL processor software is responsible for checking them.

- Conflicting facts. As an XBRL report is just a text file with a long list of Facts, it is possible that more Fact items “point to the same cell of the virtual property sheet”: the Context and the Concept reference is the same. A careful reader may still point out the “ambiguity” of the actual Data, and that is correct. You are allowed to give multiple Text Data for the same “cell” for different languages (for obvious reasons), or multiple Numeric Data if their Units are different (like, monetary values in different currencies for convenience).
  The software may check if the Numeric Data versions are the same within the given precision; no such service exists for Texts. A complete match among Facts is a likely copy-paste mistake and not relevant but any Fact collision that leads to real Data conflict is an error and must be resolved.
- The Concept definition contains the expected Data type, and in Numeric case, the type of the measurement unit. Trying to give a Date to a Numeric value or providing “elapsed days” in USD is a conflict between the Data and the Concept definition.
- The Concept definition also contains if it should be given for a period or for a date instant. If the Fact refers to an “instant” Concept (share price at a given date) and a “period” Context (2023-01-01 / 2023-12-31), that is a conflict between the Concept and the Context.
- Checking the coherence of multiple values is a more complex task: the system should “understand the context of the Attributes” but the XBRL core does not assign meaning to the Concepts. It is very likely that the actual value of an instant Concept at the beginning of each year comes from the value of the same Concept and another period Concept (representing the change) for the previous year, but this relationship is given in the Taxonomy (described in the next section).
  However, granularity is a relationship understood on this level. If you have the same numeric Concept given for each month and the containing year, the sum of the monthly ones should be equal to the year value (in the range of the precision of that value). Similarly, if you have a dimensional distribution of a Concept (like

in the example of TravelDistance), the sum of the values for the TravelMode segments (Road, Rail, etc.) should be the same as the total value in the Context not split by TravelMode dimension.

Knowledge in XBRL

XBRL is not new, for example, the US SEC collects almost 300 report types in XBRL format, close to 800,000 reports. It processes many of them and through the EDGAR API, you can access around 100 million data points about more than 17 thousand companies. (Rough numbers based on internal technical tests).

This is a huge, live knowledge graph: an interconnected network Knowledge Items of Entities, Contexts and Facts. It is populated from the incoming XBRL reports that are also stored during processing, so you have both the read-only source documents (legal responsibility) and the continuously changing knowledge graph that is the best, transferrable knowledge representation of the US economy.

Companies regularly send XBRL reports to the SEC. You already know, this means lots of Contexts related to a company and a period or instant, filled with comparable values of standard Attributes defined in shared taxonomies. The reports are checked during processing, so they are valid by the processing rules. However, subsequent reports may contain Contexts pointing to the same company and time, in conflict with previous data – depending on the conflict resolution policy this can be a way to correct any error in the knowledge base while maintaining the integrity and read-only nature of the individual reports.

As a result, the SEC “knows as much as possible” from the US economy. They can improve their understanding by refining the taxonomies and asking for more details for some report types. They could even export any segment or their complete knowledge to a single XBRL report as the syntax allows everything that this would need. Why is it interesting? The SEC collects data for the US economy. The ESMA does the same for the EU; other countries and regions have their own local organizations, mostly using XBRL for its flexibility and tooling. There is no technical limitation to use XBRL as a bulk query and transport platform among these organizations, creating a global oversight on the economy. Of course, each organization my decide to keep parts of their knowledge private. In a transferrable knowledge environment, this is not just a dream but a simple technical task. They can identify the Attributes, Entities, Contexts, Dimension granularity, etc. that they want to leave out of the export.

However, the homogeneity of the technical layer is only a necessary foundation for a global knowledge share platform; these organizations use different terminologies in their custom, locally shared taxonomies. To overcome this limitation, you should understand the key features of the primary taxonomies (Layer 2), then see how to create homogeneous knowledge from reports using different terminologies (Layer 3).

This level shows the true potential of XBRL: this global platform connects to a specific environment through the Concept and Unit definitions. XBRL is already the official platform for collecting ESG knowledge globally, but practically any scenario that wants to reliably collect and share knowledge can use an XBRL Information System.

Technical Appendix: XBRL Format Differences

XBRL itself is an abstract standard and defines multiple formats to publish reports, optimized for a specific use case. This is a short summary of these use cases and the consequent structural differences, for the actual technical details please consult an XBRL reference.

- XML (XBRL): content-oriented and optimized structure. This xml file first defines the contexts and units as separate identified tags, followed by the list of facts that refer to the contexts and units, thus minimizing the file size and making report management easier. The taxonomy creators, the report publishers and those who analyze them work with the same, standard XML tools; with proper XML background knowledge the whole environment is self-explanatory. However, this is also a compromise: extracting the data from source systems and transforming them into the report xml is complicated; and regardless of this effort, the result is not suitable for human consumption like a properly formatted business summary. The latter must be created from the XML or even from yet another intermediary data collection.
- JSON: data and process-oriented structure. The facts are independent JSON objects that contain all context and unit information. As there is no reference resolution, this format allows very simple fact processing, filtering on context data, and any fact subset remains self-contained. However, this results in larger file size, does not support context data modification because context data is embedded in all facts, and it requires different (JSON) tooling. It is unlikely that reporters would use JSON as their primary format; and even though report processors would benefit from it, they can produce it automatically from other formats.
- XHTML: presentation-oriented format. In this case, the report is an XML-compatible HTML file that any HTML editor tool can create, the XBRL compatibility is provided by extra inline XBRL tags. The report publisher uses these tags to create the required XBRL report environment: include taxonomies, define contexts and units. Then, wrap all XBRL facts of the XHTML report (numeric values and non-numeric fragments) and connect to the context and units using the proper tags. In this format, all facts are user-friendly text, including numbers and dates which must be converted back to data values, this conversion is controlled by the ‘format’ fact attribute. The XHTML format is arguably the hardest to produce or process, however, it is perfect for human consumption both printed and online. From the publishers’ point of view, for a little extra effort (they execute the same data extraction and transformation, only to a slightly different format) they get one result that is suitable for both the authorities and the public. The report processors extract XBRL information from XML and XHTML. The latter is slightly more complicated, the files are significantly larger with the human-only content, but they can use the same tools and algorithms. The only critical difference is converting text values back to numeric or date data that appears only with this format and must be flawless.

- CSV: generation-optimized format. This consists of a JSON 'catalog' and multiple CSV files that contain data for the facts. A report publisher with a complex IT infrastructure and many source systems can automate the data extraction and conversion into the CSV files and create a JSON catalog to create a valid XBRL report. For a bit more initial investment, this environment allows automatic updates to the report data content while the structure or the target report taxonomy is the same, and they can follow changes by modifying the CSV generation scripts and the catalog. The report processors need a tool to import the CSV format but with that, it's just another XBRL input.

Layer 2: Primary XBRL Taxonomies

As seen earlier, transferring knowledge is a two-step process: you first create and publish the transfer rules (these are the "primary" taxonomies), then use them to create a "snapshot" of the actual system in the form of XBRL reports. These rules often belong to separate domains. Like in the SEC XBRL filings, in every financial report you

- Identify and register some data about your company. You do this using the DEI taxonomy, maintained by the SEC – probably used not only for financial reports.
- Disclose the required financial data. You use the US-GAAP taxonomy, maintained by the FASB – probably not only used when reporting to the SEC.
- Add some data specific to your industry, company, etc. For this, you use additional taxonomies maintained by various authorities and your own company.

You can refer to theoretically unlimited amount of taxonomies in an XBRL report, as long as you provide individual local IDs to them (this is what you will use as the first part of all external IDs like dei: DocumentPeriodEndDate or us-gapp:Assets). However, the usability of the collected reports depends on the granularity and "completeness" of the commonly used taxonomies. It does not really help if common terms like Assets are defined in a hundred custom company-level taxonomies and reported using them. Similarly, the free text values are also not unambiguous, comparable and machine processable. The real benefit is in the numeric, date, reference and category attributes, these are that you can reliably use for filtering and running algorithms.

For this reason, the primary taxonomies are generally

- Disjuncts in an actual environment (like SEC or ESEF filings), so you know which to use to disclose any data by its domain.
- Managed by a single organization that wants to keep total control over it because it is responsible for the coherence and required transparency of the reported data to a higher authority (like the FASB towards the federal government).
- Tend to grow in size and complexity as they gradually reach out towards more details following the requirements of the authorities that want to use the collected data. They often "standardize" segments from industrial or company-level taxonomies to provide a common reporting method; also favor extracting transferrable data from textual explanations for processing.

The second goal of primary taxonomies is to enforce report creators submit valid reports. The definition of the Concepts assigns a meaning to these Attributes and that allows creating validation rules among them. For example, if you both sell goods and provides services, your profit is the sum of both, without taxes – this is a mathematical expression among the Concepts. You report these Facts individually for a period, the expression among them can serve as a validation – or even a way to calculate one recommended value if all others are provided.

A complex taxonomy like US-GAAP contains thousands of Concepts, for a large Entity you must report thousands of Facts for many Contexts – it is easy to make mistakes. However, the report editor software loads all the used taxonomies including these validation expressions and notifies you of any validation error that it finds. Note the benefit of this separation: you can use the same software to work with any taxonomies, keep using the same software over years and decades, because it reads the current concept list and validation expressions to evaluate from the Taxonomy definition.

The third goal of taxonomies (not only primary ones) is to help their users (reporting companies, consultants, authorities and independent analysts, researchers) get familiar with their content, goals, offer a standard way to look at similar disclosures from various Entities. For this reason, Taxonomies

- Organize Concepts into hierarchies and other structures; such organizations allow automatic validation as well.
- Provide textual information (various labels, detailed description, links to more formal definitions like paragraphs in the code of law, etc.) even in different languages. For example, the European ESEF taxonomy is an extension of the IFRS taxonomy created by the IFSB; a key content is the official translation of all original English texts to all the official languages of the member countries.
- Include common presentation templates that you can use to display the content of any report, with the official labels even in your own language, regardless of the original language of that report. (Of course, this does not affect the Data of Text type, but even in that case, you may display the value in the target language if present in the report or even apply automatic translation with clear marking). With these common templates, taxonomy experts can efficiently analyze and compare the data coming from different Entities and share their findings with each other as they refer to the same concepts, hierarchies, tables, etc. even if they work in different business segments or speak different languages.

The following sections contain short summaries of the abstract content of a Taxonomy; for more details and exact definitions please consult the XBRL reference.

Concepts

A Concept is simply an Attribute definition that you will need to transfer your knowledge into an XBRL report. This section focuses on some aspects that appear in financial context and environments you will meet them. We use the 2024 version of US-GAAP and the CoreFIling web interface as the source of examples, but you will meet similar scenarios in any mature XBRL primary taxonomy.

A Concept tries to provide a precise, objective definition to a common idea, and it is often harder than seems. For example, the positive change of your bank account is your Income, a negative change is your Loss – and you want to give a name to the Concept that refers to this change regardless of the direction and is also familiar to you. The result is IncomeLoss, somewhat resembling to the joke, “the sign of a good compromise is when both parties are equally disappointed.” In many cases the actual Concept identifier is the result of such compromise with the goal that if you search for a term that you know, you should find the identifier. You as a user may look for “Income” or “Loss” depending on your situation – you will find this Concept either way.

In many cases like IncomeLoss (or in proper terminology, Concepts related to monetaryItemType) the value has a “direction” that is essential during calculations and validation: you should add to the ones with the same and subtract from the ones with the opposite direction. This “direction” is the “Balance” attribute of the Concept that can either be “credit” or “debit”; a positive “credit” or a negative “debit” value is generally good for you. Other attributes of a Concept:

- Period Type gives you if this Concept can appear in an “instant” or a “duration” Context.
- Nillable indicates if you can leave it blank, false for mandatory Concepts.
- An Abstract Concept is a “container node” in a hierarchy without actual value.

As explained above, there is a need to dig deeper for details, that generally means a simple Concept like IncomeLoss becomes a function of many other, smaller values that the supervisor authority wants to see in the general taxonomy (to avoid the need to dig down into any custom taxonomies). This results a high granularity over time and a naming scheme that reflects it. So, you will soon find inconvenient long identifiers like “IncomeLossFromContinuingOperationsIncludingPortionAttributableToNoncontrollingInterest”. This is just an inevitable consequence of the role of a primary taxonomy: it needs to capture a lot of details to keep them in the scope of a general analysis.

However, you can use software tools to browse, search and select these identifiers instead of typing them – as explained, they can work with any taxonomy and provide you the same interface that you will find familiar after some time.

Context Dimensions

The definition structure is simple: the taxonomy defines a “Domain” node, and “Member” items under it. In most cases you see a list of Members under the Domain, but the standard allows placing Members under another Member and form a tree structure.

You can define one “default” Member to each Dimension to which Facts are collected from Contexts that do not carry this Dimension. If not specified otherwise, the default Member of any Dimension is the Dimension itself. This is a convenient technical trick that allows displaying “all items” along each segment; and its value should be the sum of all other values (assigned to a “real” Dimension Member).

Just like Concepts, the primary taxonomy wants to integrate any “additional granularity” from all areas to make them visible for a central analysis, to let the Entity use them instead of a local definition. Consequently, you will see many Dimensions in the US-GAAP Taxonomy, from “Stock Classes” to Fitch or S&P ratings. The small advantage here is that the member count is relatively low, following the practical limitations of the external sources.

Calculations

Calculations use the Formula tool of the XBRL standard to create mathematical expressions among the Concepts.

In one Report, you can create multiple Contexts of either instant or duration type, and link various Facts to them. Each Fact is a Context – Data pair. Now, if you examine one Context, you see a “virtual property table” containing values to many Concepts. If you have created a calculation formula among these Concepts, now you can use the values from the property table to see if they fit. If not, that is an error related to that calculation in that Context. If you evaluate all given Calculations for all Contexts, you have a validation result for that report.

If an Information System like SEC only accepts a report if the whole validation process succeeds, you can rely on the imported knowledge according to the current definition of all taxonomies.

Presentation Trees (Statements, Disclosures, etc.)

This far, we focused on the technical details of how the Entity’s knowledge is transferred to a valid XBRL report, a bottom-up perspective. However, the Taxonomy follows a top-down approach. When you browse the US-GAAP taxonomy, you see a list of trees like “Statement of Income” and “Disclosure of Inventory”, they are the Presentation Trees.

When you create an XBRL report of a given type, you must fill the required set of statements and disclosures. To accomplish that, you define Contexts of the required periods and provide Facts for the Concepts listed in them. When you browse the Taxonomy, you see that the same Concepts appear multiple times in different statements or disclosures. Of course, you should provide one Fact for each Concept in the Context and that will appear at all referred locations.

An XBRL report only contains the Contexts and Facts as described in Layer 1. The processor software loads the required Taxonomies and the list of Presentation Trees, and the Calculations related to their content. Therefore, you can see and navigate through the data content through these presentation trees and validation messages. Just like a digital image: the file only contains a long list of numbers; the viewer application knows how to organize and set up the pixels from them.

Templates (Pivot tables, Data Point Model)

There is yet another way to view the data content of an XBRL report. At the bottom, that is a very long list of values that are “tagged” with additional properties: an Entity, a time instant or duration, a Concept, a Unit or language and perhaps yet other custom Dimensions. Our goal now is to select some of these values by some tags and organize them into a table by others. The fundamentally new aspect of this approach is that it handles the “fixed properties” (Concept, Entity, Period, …) just like the other, custom dimensions. This allows extending the data source of the table from one Context to many, even include multiple Reports from different Entities.

In the most common table definition (“Template” in XBRL terminology), the first column is the Concept, so in the table each row displays the value of the same Concept for different source. It can even borrow the tree structure of the concepts from a Presentation Tree and organize the rows accordingly (adding header and summary rows). As columns, it can use the time of the Contexts as a simple example, thus you can see all the values for an examined period in the required granularity, like some selected Concepts of your Income Statement from 2013 to 2023 in the table body and the subsequent years in the first header row.

For a more complex example, imagine that you collected your travelDistance for the same period – but you have increased the granularity of your data by adding the TravelMode dimension, so you have multiple values (Facts) for the same Concept and Period, whenever you registered the TravelMode in a Context. Now, you can use the TravelMode as a “filter” of your table: if you select one of them (like, TravelMode = Rail), you got rid of all duplicates and can fill the table with all known values for Rail travels (zero when there is no registered Fact for a period). You can also use TravelMode as a secondary column definition, so you have a second header row listing all values of TravelMode under each year in the first header row. This is practically the pivot table functionality that you have in Excel, but here the data comes from your XBRL Information System with all the knowledge it had collected.

Similarly to the Presentation Trees, the Taxonomy contains various Templates that you can use to review or compare the content of compliant Reports in a standard way.

Technical Appendix: Taxonomy file structure

As explained before, the XBRL standard contains 4 different file formats to store the report data. They are equivalent in their capabilities (any report can be stored in any of the formats without data loss) but optimized for different use cases (manual creation, processing, presentation, automated generation).

However, Taxonomies are always created and used the same way. Some special organizations manage a few Taxonomies; they very rarely create new ones but regularly publish new versions of the existing ones. Taxonomies are loaded by software to help create, (process, display, etc.) XBRL reports.

Therefore, the optimal solution is one, final, reliable, flexible, etc. syntax rule set for creating taxonomies. Consequently, Taxonomies are stored in XML files using standard extensions: Namespaces, XML Schema Definition (XSD), XMl Linking Language (XLink). Each Taxonomy is stored in a complex folder structure and (quoting the FASB site) “requires XBRL enabled software to view” – but here are some introductory details.

Generally, you can download the taxonomy as a .zip file (this is how an XBRL enabled software works). You will find a META-INF folder with a taxonomyPackage.xml and a catalog.xml in it. The first one lists the “entryPoints”: tailored versions of the same taxonomy for different audiences, like an SME (Small or Medium Enterprise) category company does not need all the complex content of a multinational one. Each entryPoint then refers to a root schema definition (.xsd) file with a complete URL. The catalog.xml contains URL prefix to folder mappings, so you can consult that to find any referred files in the Taxonomy in the local folder structure. Unresolved URLs should be downloaded.

The internal structure of each entryPoint depends on the Taxonomy creator. In general, they organize the definitions to various folders and include the required ones into each entryPoint. You can follow the namespace import references.

You will find element definitions, this is how Concepts, Dimensions, etc. are created:

```
<xs:element id='us-gaap_Revenues' name='Revenues' nillable='true'
substitutionGroup='xbrli:item' type='xbrli:monetaryItemType'
xbrli:balance='credit' xbrli:periodType='duration' />
```

... and links that connect elements to each other or extend them with additional information, like this is how we assign a default and a “total” label to the same item.

```
<link:loc xlink:href="us-gaap-2024.xsd#us-
gaap_Revenues" xlink:label="loc_Revenues" xlink:type="locator"/>

<link:label id="lab_Revenues_label_en-
US" xlink:label="lab_Revenues" xlink:role="http://www.xbrl.org/2003/role/label" xl
ink:type="resource" xml:lang="en-US">Revenues</link:label>

<link:label id="lab_Revenues_totalLabel_en-
US" xlink:label="lab_Revenues" xlink:role="http://www.xbrl.org/2003/role/totalLabe
l" xlink:type="resource" xml:lang="en-US">Revenues, Total</link:label>

<link:labelArc order="1.0" xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept
-label" xlink:from="loc_Revenues" xlink:to="lab_Revenues" xlink:type="arc"/>
```

For easier management, the content of the files is indicated in their name, like “lab” (labels), “cal” (calculations), “ref’ (references), “def” (definitions), “pre” (presentations).

As apparent from the examples, the Taxonomy creator has great freedom to organize the definitions according to their specific needs, and it may take a while to get an oversight of a major one like US-GAAP. On the other hand, they always use the same overall structure, simple and standard tools (xsd, xlink) supported in any programming platform, there is no “hidden trick” or proprietary / binary formats, just a huge amount of data in standard XML. You can use any general XML tool to peek into them if you really need – otherwise, you should rely on an “XBRL enabled software” as recommended.

Layer 3: Secondary Taxonomies

The primary taxonomies exist to allow transferring knowledge into an Information System through many XBRL reports. They are optimized for the best achievable granularity, not processing. In many cases, some taxonomies are overlapping like US-GAAP and IFRS, defining conflicting Concepts and Dimensions for the same “meaning”. This is like a hospital where come doctors speak English but others French. The solution is similar: in Anatomy, they decided not to convince all the others to speak their language but use Latin, a language that nobody speaks in real life.

Instead of moving backward and change anything in the source reports, accept that they all speak their own languages. SEC uses its own DEI taxonomy to identify and register knowledge about the Entitles identified by CIK (SEC’s Central Index Key). SEC will keep updating DEI and assign new CIKs. The ESEF uses its own taxonomies, and LEI (Legal Entity Identifier). Apart from the fact that no “super-organization” can unify these differences; the knowledge collected in the different environments will not change.

Accept that these Information Systems use their own calculations and presentations. They ensure that the transferred knowledge is valid; organizations collected experience and experts to keep transferring knowledge into these ecosystems. They will not change, just like patients will keep talking to their doctors in English or French, not Latin.

However, the fundamental XBRL layer is the same: Entities, Reports, Contexts, Facts – the objective, unambiguous knowledge is there. The missing element is another Taxonomy, defined and handled by the same infrastructure as the primary Taxonomies. For financial reporting, Charles Hoffman, the author of XBRL created the Fundamental Accounting Concepts (FAC).

FAC does not focus on capturing all details of an American company; it defines the fundamental Concepts, validation rules, hierarchical and tabular data presentations optimized for accountants, financial audits, analysts, researchers. This layer covers the fine details and specialties of the primary taxonomies, provides a uniform, globally meaningful view on the underlying knowledge. FAC, just like US-GAAP, is a proper, live, continuously improving Taxonomy that follows the needs of its users: auditors, accountants, financial experts.

Populating an FAC XBRL Report

The FAC Concepts have just as precise, formal definitions as those in the primary Taxonomies. This allows create matching rules between them. These rules are likely unidirectional: you can generate an FAC Fact from one or more US-GAAP fact, as FAC is simpler, less granular than the primary US-GAAP. So, in most cases you need a summation of primary Concepts, evaluated by using the values of the related Facts.

Regarding the Contexts, the custom Dimensions work the same way: whenever you have Dimensions in the primary taxonomies, you must translate them to common, FAC-level Dimensions. The good news here is that most of these Dimensions already represent global categories (like the S&P rating classes), in this case the values are only “re-packaged” to the FAC taxonomy. In other cases, the different primary taxonomies may have different Dimensions, or not equivalent members.

For both Concepts and Dimensions, the relevant parties (authors of the Taxonomies) must negotiate over the transfer rules. Charles Hoffman already created partial mappings to FAC from US-GAAP and IFRS. This work must be completed in agreement with the FASB and IFSB and even after done, must be regularly updated with the subsequent changes in all taxonomies. This is a complex task, but it enforces the cooperation and shared understanding among the taxonomy authorities, and enables a common, more transparent, global economic oversight.

In the case of Entities, it is very likely that a global identification scheme will be necessary. The problem is that companies may have multiple registrations, like have both a CIK and a LEI identifier; they are different as strings but point to the same Entity. The common global financial Information System either needs an internal, unique (“secondary”) entity identification scheme to which it translates all external (“primary”) identifiers, or an algorithm that can select the “master” primary identifier for any Entity, like, for a US company, the CIK is the primary identifier, and its LEI is translated to that when creating the FAC Context for it.

Important note: the original Report content remains intact during the translation process, only new FAC Facts and Contexts are created. If we need to translate the Entity identification, the new Contexts will use that identifier. This can be done as the XBRL syntax allows referring to multiple Entities; semantically it is also valid: the Report still refers to the same Entity, just uses a different identifier. On saving the Report, the original segment, the FAC only, or both can be exported to XBRL.

FAC Report validation and presentation

As a proper Taxonomy, FAC contains validation rules, presentation hierarchies and table templates. After the Fact and Context generation, the Report contains FAC Facts and Contexts, these services work on them just like with the primary taxonomies.

Recalling

- the bottom-up nature of Aspects (a Knowledge Item "has" an Aspect if it contains all required Attributes),
- that Concepts are Attributes with some XBRL-specific extensions,
- and that a Context is a Knowledge Item collecting its Attributes from the Facts referring to it

we can see that calculations and presentations are "applicable" to a Context if that contain some or all Concepts to be used by them. In this way, the newly populated FAC Report segment can be validated (and the validation may both pass or fail) and displayed (showing all required data or missing some, indicating a missing data or translation rule).

Learning from the underlying primary taxonomy and SEC Report Type, FAC can have similar requirements, like a set of FAC validations to pass and reports/tables to fill for all 10-K reports. These are understandable for the financial experts as FAC uses their terminology instead of the US-GAAP; but as the original Report content is intact, they can always look behind the secondary Taxonomy and see the original US-GAAP Facts.

The same services can be used while preparing the Report. You want to submit an SEC 10-K report, but while editing it, you can still load the FAC Taxonomy overlay and check the report from purely financial perspective. It is even possible that the report fails an FAC-level validation while it is accepted by the US-GAAP taxonomy or improve a Text Fact based on the description of the FAC Concept it is translated to.

However, you can edit your report only through the primary Taxonomy, the "natural language" of your report, the secondary Taxonomies are for view only.

FAC-based analysis

The true power of the FAC taxonomy is that it focuses on the needs of the financial experts, uses their terminology. The underlying translation layer is responsible for the proper translation of the original Report, on FAC level, all of them look the same, their data content can be compared, aggregated, analyzed regardless of the sources and different primary taxonomies. This is useful in situations like validating reports from the same Entity, some using US-GAAP and others IFRS; or creating global situation or trend analysis on sector level like automotive or coal mining. All this is doable if the reports are in XBRL and there is a translation from the primary taxonomies to FAC.

Outlook to ESG

To handle urgent Environmental, Social, and Governance (ESG) issues, we need an efficient method to capture as much local knowledge as possible in many, often overlapping environments, then extract coherent, transparent knowledge for proper, conscious resource management and prioritization.

The XBRL standard had already been selected, the SEC published its ESG Taxonomy and expects the first set of reports from the largest US companies next year. The international ESG taxonomy is under development, just like the EU version. Some countries like India already started collecting ESG data in XBRL using their taxonomy.

The experience of XBRL financial reporting shows great potential like you can peek into millions of reports and hundred million data points through the SEC EDGAR portal; but also missed opportunities like the fact that after 20+ years, we still miss the common taxonomy and knowledge extraction features above the differences between US-GAAP and IFRS, although in the works by many, including the XBRL author, Charles Hoffman since the beginning.

With the explained knowledge layer separation, we can step forward in this direction both in financial reporting (utilizing the decades of work and research including existing mappings), and in ESG (shorten the learning curve, give the international ESG taxonomy the same role as FAC).