International Public Sector Accounting Standards (IPSAS) Profile

Filing Manual (PROTOTYPE)

The International Public Sector Accounting Standards (IPSAS) XBRL Application Profile is an application profile of XBRL which is 100% compliant with the XBRL 2.1, XBRL Dimensions 1.0, XBRL Formula 1.0, Inline XBRL, and Generic Linkbase 1.0 specifications. The profile follows the spirit of the XBRL Abstract Model 2.0 Public Working Draft. The profile leverages the best ideas of XBRL architectures used for financial reporting. This profile can be used (should be used) by economic entities creating XBRL-based financial reports using IPSAS. This application profile uses XBRL's extensibility features. This document provides non-normative explanation and a formal specification for normative guidance.

2019-01-15 (DRAFT)

Co-editors[1]:
Charles Hoffman, CPA (Charles.Hoffman@me.com)
Raynier van Egmond (raynier@xbrlcp.com)

[1] Accounting Process Automation Google Group, https://groups.google.com/forum/#!forum/accounting-process-automation

Introduction

This document is a normative explanation of the International Public Sector Accounting Standards (IPSAS) XBRL Application Profile. The profile is compliant with the XBRL 2.1[2], XBRL Dimensions 1.0[3], XBRL Formula 1.0[4], Inline XBRL[5], and Generic Links 1.0[6] specifications. The profile follows the spirit of the XBRL Abstract Model 2.0[7] public working draft and the Open Information Model 1.0[8] candidate recommendation.

This application profile leverages the *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*[9].

This profile leverages **best practices**. A best practice is defined as a method or technique that has been generally accepted as superior to any alternatives because it produces results that are superior to those achieved by other means or because it has become a standard way of doing things.

The profile takes the best ideas from financial reporting taxonomy architectures[10] and combines those ideas to create a safe, reliable, robust, thoroughly tested, implementation of an XBRL-based business report. This profile uses a well understood multidimensional model.

1.1. Intended audience of this document

The intended audience of this document is business professionals creating XBRL taxonomies and software developers implementing software intended to be used by business professionals in the creation of XBRL taxonomies and related XBRL instances.

---

2 XBRL International, *Extensible Business Reporting Language (XBRL)*, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html

3 XBRL International, *XBRL Dimensions 1.0*, http://www.xbrl.org/specification/dimensions/rec-2012-01-25/dimensions-rec-2006-09-18+corrected-errata-2012-01-25-clean.html

4 XBRL International, *XBRL Formula 1.0*, https://specifications.xbrl.org/work-product-index-formula-formula-1.0.html

5 XBRL International, *Inline XBRL*, https://specifications.xbrl.org/spec-group-index-inline-xbrl.html

6 XBRL International, *Generic Links*, https://specifications.xbrl.org/spec-group-index-generic-links.html

7 XBRL International, *XBRL Abstract Model 2.0*, Public Working Draft 06 June 2012, http://www.xbrl.org/specification/abstractmodel-primary/pwd-2012-06-06/abstractmodel-primary-pwd-2012-06-06.html

8 XBRL International, *Open Information Model 1.0*, Candidate Recommendation 02 May 2017, http://www.xbrl.org/Specification/oim/CR-2017-05-02/oim-CR-2017-05-02.html

9 Charles Hoffman, CPA and Rene van Egmond, *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*, http://xbrlsite.azurewebsites.net/2019/Library/MethodForImplementingStandardFinancialReportUsingXBRL.pdf

10 Charles Hoffman, CPA, *XBRL-based Digital Financial Reporting Profiles and General Business Reporting Profile*, http://xbrlsite.azurewebsites.net/2018/Library/Profiles-2018-10-22.pdf

The average business professional should not need to read or understand this document. Software vendors should embed the information specified within this document within software applications such that the average business professional may only comply with these rules.

1.2. Organization of this document

This document is organized to be read linearly, start to finish.

1.3. Terminology

Throughout this document, several words are used to signify the requirements of this specification. These words are capitalized when they should be interpreted as having a strict meaning. The following definitions are taken from RFC2119[11] and modified so that they are more appropriately worded for use within this standard.

Term	Meaning
MUST	This word means that the definition is an absolute requirement of this specification.
MUST NOT	This phrase, or the phrase "MUST NEVER," means that the definition is an absolute prohibition of this specification.
SHOULD	This word means that valid reasons may exist in particular circumstances to ignore a particular item, but the full implications must be understood and be carefully considered before choosing a different course.
SHOULD NOT	This phrase means that valid reasons may exist in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully considered before implementing any behavior described with this phrase.

11 IETF, *Key words for use in RFCs to Indicate Requirement Levels*, https://www.ietf.org/rfc/rfc2119.txt

Term	Meaning
MAY	This word means that an item is truly optional. One business unit may choose to include the item because a particular marketplace requires it or because the business unit feels that it enhances the product while another business may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. Conversely, an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

Again, keep in mind that the primary purpose of this document is to create reliable, safe, predictable, interoperable, high-quality, high-function software applications in support of the automation of accounting, reporting, auditing, and analysis workflows and tasks.

1.4. System narrative

A general purpose financial report is a high-fidelity, high-resolution, high-quality information exchange mechanism. That mechanism has historically used the media of "paper". Over the past 50 years or so, that paper-based mechanism has given way to a new mechanism, "e-paper". By "e-paper" I mean PDF documents, HTML documents, Word documents and such.

XBRL is a new media, a new mechanism for creating a general purpose financial report. XBRL is a high-fidelity, high-resolution information exchange media that allows for the creation of high-quality financial reports.

A **report** is simply some set of information. A report could be a ledger, the entries within a journal, the contents of a spreadsheet, the results of a query from a database, etc. What all these reports have in common is the notion that some set of facts is grouped together usually for some specific reason. Another term for this is the notion of a fact ledger[12]. Reports are readable by machines but the information in a report can also be readable to humans.

A **report fragment** is a set of facts which go together (tend to be cohesive and share a certain common nature) for some specific purpose within a business report. For example, a "balance sheet" is

[12] Charles Hoffman, CPA and Andrew Noble, *Introducing the Fact Ledger*, http://xbrlsite.azurewebsites.net/2018/Library/IntroductionToTheFactLedger.pdf

a component of a financial business report. "Maturities of long-term debt" is a component of a financial business report.

A **fact** describes a single, observable, piece of information contained within a report which is contextualized for unambiguous interpretation or analysis by distinguishing characteristics of the fact. Every fact has exactly one value. Every fact must have one characteristic but may have many characteristics.

A **characteristic or aspect** provides information necessary to describe a fact or unambiguously distinguish one fact from another fact. A fact has a set of one or many characteristics or aspects, the set being a property of the fact, which describes the fact.

A **document** is an organized set of report fragments. Report fragments are sequenced or organized in an appropriate flow. A report can be sensibly and logically represented as an electronic document (such as a Word or PDF document), a web document (such as an HTML file or Wiki page), an OLAP-type cube, a multidimensional hypercube, a spreadsheet, or any other visual form including something provided by a dynamic viewing application (such as a pivot table, or drill-down information viewer).

The presentation or view of a report within a document is created by one or more digital report viewing tools (commonly known as rendering engines), which are specifically capable of reading the structured digital format (in this case XBRL technical syntax) and then creating a structured presentation. It is important to know that different rendering engines may present the same digital business report in different ways. This does not mean that the underlying representation or meaning being conveyed is different, only that the translation of the semantic representation to a visual presentation is different.

Rendering engines are expected to understand the semantics of a business report which will help them in creating understandable human readable renderings from the raw XBRL. Alternatively, pixel-perfect renderings can be created manually using Inline XBRL.

1.5. Methodology

The methodology which will be used to create and maintain this specification and related artifacts is the build, test, deploy, and maintain methodology[13]. Quality matters and testing and proof of concepts contribute to achieving the high quality that is necessary.

[13] EDM Council, *FIBO™ Build, Test, Deploy and Maintain Methodology*, https://spec.edmcouncil.org/fibo/doc/20170930_FIBO_BTDM.pdf

The method for creating XBRL taxonomies and XBRL instances is the *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*[14].

1.6. General high-level requirements

The following is a summary of general high-level requirements of this specification:

- **Straightforwardly usable over the Internet**: The business reports are intended to be used over the Internet.
- **Shall support a wide variety of common business use cases**: A wide variety of business use cases should be handled, considering the 80/20 rule is appropriate. It is not a requirement to meet all business use cases.
- **Minimal options**: The number of optional features is to be kept to the absolute minimum, ideally zero.
- **Formal and concise**: The design shall be formal and concise.
- **Readable by both humans and machines**: A report should be readable by both humans and machines. Information provide within a report should be more a representation of information than presentation oriented.
- **Global standard format with high level of semantics:** The format of the report should be a global standard which can provide a high level of semantic clarity.
- **The "model" and the "view" should not be intermingled**: The information and the model should be separate.
- **Business rules separate**: Business rules should be separated from the information. Business rules which are external to the report should be allowed for.
- **Managed global standard**: The report should ultimately be a global standard under the control of someone like OMG, XBRL International, ISO, Apache OpenOffice, or some other such organization.
- **Provide technical syntax, structural interoperability, but be domain neutral**: The XBRL technical syntax will be used to represent a report and the metadata/rules of the report providing a formal "shape" of a report. But the semantics of the information within a report is determined by

[14] Charles Hoffman, CPA and Rene van Egmond, *Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax*, http://xbrlsite.azurewebsites.net/2019/Library/MethodForImplementingStandardFinancialReportUsingXBRL.pdf

the creators of the report. Business domain user information would always fit into the required report "shape". Reports are expected to interoperate semantically with other semantic standards such as FIBO[15] and proprietary approaches to representing semantics such as SSIM[16] created by software vendors.

- **Format should allow for versioning, collaboration, etc.**: The syntax format should allow for ease of versioning, constructing systems which are collaborative in nature (multi-user).

1.7. System high-level requirements

The following is a summary of the high-level system requirements for reports:

- Minimize the probability of ambiguity between what a reporting entity may say and what a user of the report may interpret.
- Maximize safe reuse of information contained within a report.
- Minimize the possibility of errors within the report.
- Maximize the probability of detecting errors using automated processes assisted by software applications.
- Maximize the probability that any software which supports XBRL will be able to make use of a report with no need for adjusting the software.

1.8. Principles

Principles help you think about something thoroughly and consistently. Overcoming disagreements between stakeholders and even within groups of stakeholders is important. Agreement between stakeholder groups and within stakeholder groups contributes to harmony. Lack of agreement contributes to dissonance. Principles help in the communications process.

A first step of arriving at harmony is outlining the interests, perceptions, positions, and risks of each constituency/stakeholder group.

A "stakeholder" is anyone that has a vested interest. Another term for stakeholder is "constituent". A "constituent" is a component part of something.

---

[15] EDMcouncil, *FIBO*, https://spec.edmcouncil.org/fibo/

[16] Pacio, *Standardised Semantic Information Model (SSIM)*, https://www.pacio.io/wp-content/uploads/2018/08/stack-grid.pdf

Foundational to arriving at harmony is having a common conceptual framework including a set of consistent principles or assumptions or world view for thinking about the system. For example, accounting and financial reporting have such a conceptual framework including principles/assumptions such as "materiality" and "going concern" and "conservatism".

This "framework for agreeing" helps the communications process which increases harmony and decreases dissonance. This is about bringing the system into balance, consciously creating the appropriate equilibrium/balance.

The following is a set of principles which those creating this specification agree to use to understand their perceptions, positions, and risks when it comes to creating this specification.

1. Prudence dictates that using information from an XBRL-based report should not be a guessing game.
2. A near zero defect report is useful, a defective financial report is not useful.
3. Rules prevent anarchy.
4. The only way to achieve a meaningful exchange of information without disputes is with the prior existence of and agreement as to a standard set of technical syntax rules, business logic rules, and workflow rules.
5. Explicitly stated information or reliably derived information is preferable to implicit information. Derived and implied are not the same things.
6. Reports can be guaranteed to be defect free using automated processes to the extent that machine-readable rules exist.
7. When possible to effectively create, machine-based automated processes tend to be more desirable than human-based manual processes because machine processes are more reliable and cost less.
8. Business logic rules should be created by knowledgeable business professionals, not information technology professionals.
9. Complexity cannot be removed from a system, but complexity can be moved.
10. Simplicity and simplistic are not the same thing. Simplistic entails dumbing down a problem in order to make the problem easier to solve. Simplistic ignores complexity in order to solve a problem which can get you into trouble.

Simple is something that is not complicated, that is easy to understand or do. Simple means "without complications".

2. Restrictions on XBRL Technical Syntax

The following section summarizes parts which exist within the XBRL 2.1 Specification, XBRL Dimensions 1.0 specification, and XBRL Formula 1.0 specification which MUST NOT exist within XBRL taxonomy schemas, XBRL linkbases, and XBRL instances which comply with this application profile.

All other aspects of XBRL 2.1, XBRL Dimensions 1.0, XBRL Formula 1.0, and Generic Linkbase 1.0 are allowed other than those items specifically prohibited within this section.

2.1. Tuples MUST NOT exist.

Tuples can always be detected because elements which define tuples have a substitutionGroup attribute value of "xbrli:tuple". No such elements are allowed under this profile.

Reasoning: Tuples are not allowed by the US GAAP Taxonomy Architecture or the IFRS XBRL Taxonomy architecture and are therefore not allowed within SEC XBRL financial filings. Tuples and XBRL Dimensions tend to provide the same functionality so both are not necessary. XBRL Dimensions provides better functionality than tuples.

2.2. Complex typed members MUST NOT exist.

Typed members can always be detected as they contain the xbrldt:typedDomainRef attribute which defines the typed member. No such attribute should ever be detected within a discoverable taxonomy set (DTS) which makes use of this profile where the typed member is complex.

Simple typed members are allowed, but not required and generally explicit members are preferable to typed members.

Reasoning: Complex typed members are extremely difficult to implement within software and therefore should be avoided. Explicit members can provide all the functionality of a typed member. Typed members do not have labels.

2.3. Context scenario elements MUST NOT exist.

Context elements which contain a <scenario> element can always be detected. No such element should ever be detected within a discoverable taxonomy set (DTS) which makes use of this profile.

Reasoning: Scenario elements within contexts are not allowed by the US GAAP Taxonomy Architecture and are therefore not allowed

within SEC XBRL financial filings. XBRL makes no distinction between dimensions and members which are contained within a <scenario> element and those contained within a element. Therefore, use of both elements is unnecessary.

[CSH: ESMA allows <scenario> rather than ; therefore this could be changed to allow EITHER or <scenario> but not both in the same document.]

2.4. Precision attributes MUST NOT be provided on any fact within an XBRL instance.

Precision attributes can always be detected on facts. No such attribute should ever be detected within an XBRL instance which makes use of this profile.

Reasoning: Precision attributes are not allowed by the US GAAP Taxonomy Architecture and are therefore not allowed within SEC XBRL financial filings. The decimals attribute, which is allowed, serves exactly the same purpose as the precision attribute. If both attributes are allowed then it make it necessary to convert from decimals to precision and precision to decimals which could cause interoperability issues.

2.5. Prohibition of relations MUST NOT be used.

XBRL's prohibition of relations features MUST NOT be used.

Reasoning: Prohibition features are unsafe.

3. Restrictions on Semantics

The following is a summary of additional restrictions explicitly placed on the semantics of reports articulated using the XBRL technical syntax which adhere to this application profile. This section basically makes things which are legal in XBRL illegal. The reason for imposing these restrictions is they cause irrational, illogical or nonsensical representations when expressed in XBRL.

3.1. Report elements contained within or defined by an XBRL taxonomy MUST clearly be defined such that they can be categorized into one of the following groups of report elements:

- **Hypercube:** A hypercube can always be detected by the value of the substitutionGroup attribute value of xbrldt:hypercube. Other common terms used for hypercube include Table, Cube, Matrix, Array.

- **Dimension:** A dimension can always be detected by the value of the substitutionGroup attribute value of xbrldt:dimension. Other terms used for dimension include Axis or Aspect.
- **Member:** A member can always be detected by the value of the dataType attribute value of nonnum:domainItemType from the namespace identifier http://www.xbrl.org/dtr/type/non-numeric.
- **Primary Items:** A primary items report element can always be detected by the fact that it is the last child of a hypercube within the presentation relations and that it has an abstract attribute value of "true". Other terms used for Primary Items include Line Items.
- **Abstract:** An abstract can always be detected by the fact that it is not identifiable as a hypercube, dimension, or member and that does have an abstract attribute with the value of "true".
- **Concept:** A concept can always be detected by the fact that it is not a hypercube, dimension, member, primary item, or abstract.

This rule implies that every XML Schema element defined in an XBRL taxonomy schema can be categorized into one of these groups and that the term "report element", or "XML Schema element" or "element" or "XBRL element" should never be used. Rather, the terms above should be used to refer to report elements.

3.2. Report element categories MUST be related in specific ways.

One report element category can only be related to another report element category in very specific ways when represented in XBRL presentation relations. Note that XBRL definition relations are more restrictive than XBRL presentation relations. The same is true with XBRL calculation relations. The intent of this rule is to minimize ambiguity and maximize consistency with XBRL definition relations, particularly XBRL Dimensions relations expressed using XBRL definition relations. These restrictions can be implemented within an XBRL definition linkbase[17].

This is a more restrictive relations model, this model is encouraged.

[17] Model structure relations represented using XBRL definition relations, http://xbrlsite.azurewebsites.net/2019/Prototype/ipsas/ModelStructure/ModelStructure-rules-ipsas-def.xml

		IPSAS STRICT Model Structure Relations						
		Parent						
		Network	Table	Axis	Member	Line Items	Abstract	Concept
Child	Network	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL
	Table	OK	Disallowed	Disallowed	Disallowed	Disallowed	OK	Disallowed
	Axis	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Member	Disallowed	Disallowed	OK	OK	Disallowed	Disallowed	Disallowed
	Line Items	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Abstract	OK	Disallowed	Disallowed	Disallowed	OK	OK	Disallowed
	Concept	Disallowed	Disallowed	Disallowed	Disallowed	OK	OK	Disallowed

3.3. Hypercubes, dimensions, members, primary items, and abstracts MUST have a periodType attribute value of XBRL equal to "duration".

PeriodType has no semantics for these types of report elements.

3.4. Hypercubes, dimensions, members, primary items, and abstracts MUST have an abstract attribute value of XBRL equal to "true".

Abstract has no semantics for these types of report elements. XBRL requires hypercubes and dimensions to be abstract. It seems reasonable to therefore require members and primary items to likewise be abstract.

3.5. Hypercubes, dimensions, members, primary items, and abstracts MUST NOT have a balance attribute.

Balance is not an appropriate property for these report elements.

3.6. Relations expressed using XBRL definition relations related to XBRL Dimensions and relations expressed using XBRL presentation relations MUST be consistent.

This rule implies that XBRL definition relations and XBRL presentation relations must never be inconsistent and therefore ambiguous.

3.7. The extended link roles of XBRL presentation relations, XBRL calculation relations, XBRL definition relations, and XBRL formula MUST be consistent for the same report fragment.

This rule explicitly states that extended link roles identify report fragments and that the extended link role of a report fragment MUST be consistent in all linkbase structures.

Reasoning: In XBRL, an extended link is used to define parts that go together. For example, the balance sheet XBRL presentation relations use an extended link such as "http://my.com/role/balancesheet" to identify the pieces of a balance sheet. That same extended line would be used on the XBRL definition relations that related to the set of XBRL presentation relations. This, likewise, must be the case for XBRL calculation relations. This is not the approach the SEC takes. The SEC basically says that a software application should search the entire set of XBRL calculations for a set that works for the network you are working with and that XBRL calculation relations should not be duplicated.

[CSH: This likely needs to be explained. This is a tradeoff between reliability, logic, and the duplication of some XBRL calculation relations.]

3.8. Each Concept which could be used in an XBRL instance to report a Fact MUST exist within at least one Hypercube.

Facts are never "free floating" in space. As such, each Concept which might be used to define a fact MUST always exist within a Hypercube.

4. Explicit Semantics

While the previous section restricts certain specific uses of the XBRL technical format for the purpose of minimizing the chances of ambiguity and otherwise eliminating irrational, illogical, nonsensical representations of information; this section articulates specific report logic or semantics.

4.1. Networks and Hypercubes MUST articulate clear, unique business meaning.

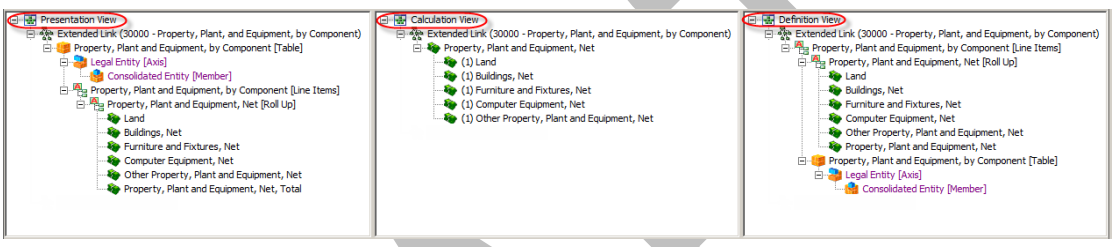
As such all Hypercubes MUST be isomorphic (carry one meaning) and all Networks MUST be isomorphic (carry one meaning). Each Hypercube and Network should identify specific, unique fragments of a Report.

Reasoning: Using the same Hypercube "Statement [Table]" for both the balance sheet and income statement (and other statements or any other report fragment) makes it impossible to

use the Hypercube to identify a report fragment. Uniquely identifying each fragment with a unique Hypercube makes querying information from a report trivial.

4.2. Report fragments MUST be represented using the same network URI across all XBRL presentation, XBRL calculation, XBRL definition, and XBRL Formula related relation networks.

In essence this means that if a report component is expressed, then the network identifier of that report component must be the same for each set of presentation, calculation, definition, and XBRL Formula networks which express information for that report component. Any URI that is the general URI of XBRL applies to the entire document.



4.3. Networks MUST be identified using numbers to sequence and sort codes to categorize.

XBRL has no specific mechanism for sequencing Networks or categorizing Networks. As such, the IPSAS XBRL Taxonomy defines the following scheme for sequencing and categorizing Networks which follows the spirit of the U.S. Securities and Exchange Commission Edgar Filer Manual (section 6.7.12).

The Network definition is broken up into three parts with each part separated by " – " (space dash space):

"{SortCode} – {Type} – {Title}"

Where:

- {SortCode} is an alphanumeric set of characters that is used only to sort the Networks sequentially for display purposes with software applications. Thus, the SortCode "10" would appear between the SortCode "2". Reporting entities are free to define their own sort code.
- {Type} must be one of the words 'Disclosure', 'Document', 'Schedule' or 'Statement'.
- {Title} is text that should allow a human reader to distinguish what each separate Network contains.

4.4. Report element names and IDs MUST NOT carry semantics.

The meaning of report elements is provided by the report element documentation, labels, references, relations, and business rules expressed.

4.5. The XBRL context element entity identifier and scheme MUST identify the entity issuing the report.

Use dimensions to provide any additional information deemed necessary to further characterize a reported fact.

4.6. The XBRL context period MUST indicate the calendar period of a reported fact.

Use dimensions to provide any other information deemed necessary to characterize an period related characteristic of a reported fact.

4.7. Members of a dimension can be arranged within one of the following member arrangement patterns:

[CSH: This needs work.]

The relations between the members of a dimension can be organized into member arrangement patterns: composition, aggregation, wholeness[18].

- **Whole-part**
- **Is-A**
- **Aggregation**

4.8. Primary items MUST be arranged within one identifiable and distinguishable concept arrangement pattern.

The following is a summary of mathematical and non-mathematical concept arrangement patterns:

- **Roll up**: Fact A + Fact B + Fact C = Fact D (a total)
- **Roll forward**: Beginning balance + changes = Ending balance (this is sometimes called a "movement analysis" or BASE pattern; beginning balance + additions – subtractions = ending
- **Adjustment**: An adjustment reconciles an originally stated balance to a restated balance between two report dates; Originally stated balance + adjustments = restated balance

[18] Here is information related to whole-part relations; *Towards Understanding Whole-part Relations*, http://xbrl.squarespace.com/journal/2015/1/20/toward-understanding-whole-part-relations.html

- **Variance**: A variance is a change across a reporting scenario. Actual amount - Budgeted amount = variance
- **Complex computation**: A complex computation is a type of concept arrangement where facts are related by some computation other than a roll up, roll forward, adjustment, or variance. For example, Net income / Weighted average shares = earnings per share. These can always be detected by the existence of XBRL Formulas.
- **Set (or Hierarchy)**: A set (or Hierarchy) is a type of concept arrangement where facts are related in some way, but not mathematically.
- **Text Block**: A text block is text or prose that may contain many numeric or nonnumeric facts. Essentially, a text block is escaped HTML.

Each report fragment MUST be organized into identifiable and discrete concept arrangement patterns.

NOTE: As new concept arrangement patterns are identified the list of supported patterns will be expanded. If information is arranged in a manner that is not consistent with one of the supported concept arrangement patterns, then the Set (or Hierarchy) pattern should be used to represent that pattern.

4.9. All computations or business rules which the creator of the report desires a user of the report to understand MUST be expressed using XBRL calculations (roll up) or XBRL Formula (all other computations).

All computations, which are part of the concept arrangement patterns, can be automatically generated by software as XBRL Formula based business rules which enforce the concept arrangement patterns. One exception to this is the complex computation pattern which could literally be any computation and therefore this is impossible to automate.

XBRL Formulas are preferred to XBRL calculations in most situations.

4.10. A Block is a report fragment which shares the same Concept Arrangement Pattern.

A Block is defined as an identifiable fragment of a report that shares the same concept arrangement pattern and member arrangement pattern within a Network. For example, the following is a Block:

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

5. Expressing or Extending Domain Semantics

This section provides a mechanism for controlling extensibility of an XBRL taxonomy. This section describes rules which must be followed when creating high-fidelity digital reports to effectively manage and control extensibility such that the creator of information and the consumer of information interpret reported information the same.

5.1. Reporting entities are free to create new Topics, however any new Topic created MUST be connected into the IPSAS Topic hierarchy.

The IPSAS XBRL Taxonomy provides all anticipated Topics that an economic entity might need to create a general purpose financial report. Any Topic not provided within the IPSAS XBRL Taxonomy but a reporting entity might feel that it needs can be created by a reporting entity; but such Topic MUST be connected to the existing set of IPSAS Topics.

5.2. Reporting entities are free to create new Disclosures; however any new Disclosure created MUST be connected into the IPSAS Topic and Disclosure hierarchy.

The IPSAS XBRL Taxonomy provides all anticipated Disclosures that an economic entity might need to create a general purpose financial report. Any Disclosure not provided within the IPSAS XBRL Taxonomy but a reporting entity might feel that it needs can be created by a reporting entity; but such Disclosure MUST be connected to the existing IPSAS Topic and Disclosure hierarchy.

NOTE: Care must be taken to distinguish between when a new Disclosure should be created and when an existing Disclosure should be extended.

5.3. Reporting entities are free to create new Tables, Axes, Members, Line Items, Abstracts, or Concepts; however any new report element created MUST be connected into an existing Disclosure or into a new Disclosure created by the reporting entity.

The IPSAS XBRL Taxonomy provides all anticipated Tables, Axes, Members, Line Items, Abstracts, and Concepts. Any such report element not provided within the IPSAS XBRL Taxonomy can be created by a reporting entity; however, all such new report elements created MUST be connected to an existing Disclosure or a new Disclosure created by the reporting entity. Further, each Concept created MUST be connected to the IPSAS class hierarchy.

6. Reporting Checklist Semantics

A reporting checklist relates to the logic of when a report fragment is required to exist within a specific report.

For example, the disclosures "Nature of Entity", "Basis of Presentation" and "Significant Accounting Policies" are always required in a financial report.

If a specific Line Item such as "Inventories" is reported, then the Disclosure "Inventory Subclassifications" is required to be reported.

This information is leverageable by software applications used to construct or consume financial reports. This information is articulated as machine-readable rules as XBRL definition relations[19].

A reporting checklist defines relations using the following arcroles which are used to define relations within a XBRL definition relations.

- report-requiresDisclosure
- report-reportsConcept
- concept-allowedAlternativeConcept
- reportingLineItem-requiresDisclosure
- disclosure-allowedAlternativeDisclosure
- report-possibleDisclosure

6.1. All Disclosures that are always required MUST be represented in a reporting checklist using the report-requiresDisclosure *arcrole.*

If a Disclosure is always required to be provided within a report provided by a reporting entity; then report-requiresDisclosure is used to identify that logic within the reporting checklist.

6.2. All Concepts that, if reported, require the existence of a specific Disclosure; that logic MUST be represented in the reporting checklist using the reportingLineItem-requiresDisclosure *arcrole.*

If a report provides a concept and the existence of that reported fact calls for the existence of a specific disclosure that logic MUST be expressed using the reportingLineItem-requiresDisclsoure arcrole.

[19] Reporting checklist example, http://xbrlsite.azurewebsites.net/2016/conceptual-model/drules-arcroles.xsd

6.3. All Disclosures that, if reported, require the existence of some additional Disclosure; that logic MUST be represented in the reporting checklist using the reportingDisclosure-requiresDisclosure *arcrole.*

If a report provides a Disclosure and the existence of that reported Disclosure calls for the existence of some other specific Disclosure that logic MUST be expressed in the reporting checklist using the reportingDisclosure-requiresDisclsoure arcrole.

6.4. All Disclosures that, if reported, could be reported using an alternative disclosure; that logic MUST be represented within the reporting checklist using the disclosure-allowsAlternativeDisclosure *arcrole.*

If a report MUST provide a disclosure and that disclosure could be provided using one or more alternative disclosures that logic MUST be expressed using the disclosure-allowsAlternativeDisclsoure arcrole.

7. Disclosure Mechanics Semantics

A Disclosure or a fragment of information provided within a financial report follows certain specific structural, mechanical, mathematical, logical, and accounting related rules. Disclosure mechanics semantics rules can be used to express this logic within an XBRL taxonomy.

For example, the Disclosure "Inventory Subclassifications" is a common financial report disclosure. That disclosure is always a roll up. That disclosure always uses the concept "ipsas:Inventories" to report that roll up of subclassifications. That disclosure commonly includes the line items "Raw material", "Work in progress", and "Finished good". If the disclosure "Inventory Subclassifications" exists; then the disclosure "Inventories Policy" is also required to exist.

This information is leverageable by software applications used to construct or consume financial reports. This information is articulated as machine-readable rules as XBRL definition relations.

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

This disclosure mechanics information can be represented in machine-readably form using XBRL definition relations using the following arcroles[20]:

- disclosure-hasConceptArrangementPattern
- disclosure-equivalentTextblock
- disclosure-requiresConcept
- concept-allowedAlternativeConcept
- disclosure-oftenContainsConcept
- disclosure-requiresAxis
- conceptArrangementPattern-requiresConcept
- disclosure-relatedLevel1NoteTextBlock
- concept-allowedAlternativeConcept
- disclosure-relatedPolicy
- disclosure-requiresMember
- reportedDisclosure-requiresDisclosure
- disclosure-oftenContainsConcept
- disclosure-hasMemberArrangementPattern

The following rules describe these disclosure mechanics.

7.1. All disclosures MUST be identified as having a specific concept arrangement pattern.

Each Disclosure is identified as to which concept arrangement pattern it possess using the disclosure-hasConceptArrangementPattern arcrole.

[CSH: To do remaining arcroles.]

8. Continuity Cross-checks Semantics

General purpose financial reports created using IPSAS have a certain amount of variability in terms of how those reports can be created at a high level. This variability is defined using continuity cross check rules.

20 Example disclosure mechanics arcroles, http://xbrlsite.azurewebsites.net/2016/conceptual-model/drules-arcroles.xsd

The different varieties of allowed/possible arrangements of concepts and the relating continuity cross check patterns can be grouped or clustered using the notion of a reporting style.

It is always better to explicitly report high-level concepts within financial reports. However, if it is the case that concepts could be reported using alternative concepts or that reporting certain line items are optional; then continuity cross-check semantics functionality can be used to describe this logic. Impute rules can be used to derive the value of a non-reported fact by using logical rules to derived the values of such non-reported facts from reported facts.

For example, the fact "Noncurrent assets" is often not reported within a financial report. If the concept "Noncurrent assets" is not explicitly reported and if the rule "Assets = Current assets + Noncurrent assets" and if the facts "Assets" and "Current assets" is reported; THEN the concept "Noncurrent assets" can be logically derived using facts which are reported and rules available such that "Noncurrent Assets = Assets – Current Assets".

Further, if a fact is reported within a report fragment and that same fact is implied to exist within another report fragment; the fact must not contradict reported or implied facts or otherwise be inconsistent with such other facts.

[CSH: To do]

9. Type or Class Relations

Concepts defined can be related to other concepts forming a "class" or "type" of concept that make up a formal set. For example, "Cash and cash equivalents" and "Receivables" and "Inventory" are all types of "Current Assets".

Types or classes of concepts defined to be used in one way may not be used in another way that is not intended by the defining taxonomy. For example, the concept "Payments to purchase property, plant, and equipment" if defined to be PART-OF "Net cash flows from investing activities" MUST NOT then be used to represent a line item of "Net cash flows from financing activities".

[CSH: To do]

10. Style Guide

When creating an XBRL taxonomy some style guide should be used to make sure that report element names and labels are created

consistently. For example, the *XBRL US Style Guide*[21] is an example of such a style guide.

11. Report Lists

When using more than one business reports, some sort of list of reports contained within a report repository is necessary. For example, the U.S. Securities and Exchange Commission provides the following RSS feed which provides lists of reports[22].

The following section specifies how to create such a report list using RSS.

[CSH: This is similar to the SEC XBRL document RSS feed. To do]

12. Units

XBRL International maintains a comprehensive global standard units registry[23]. When representing information within an XBRL-based report, when representing units, units[24] from this global standard units registry SHOULD be used where possible.

13. XBRL Formula Extension Functions

This specification defines a number of custom arcroles using the global standard XBRL approach. In order to process those custom

21 XBRL US, *XBRL US Style Guide*, https://xbrl.us/xbrl-reference/style-guide/

22 U.S. SEC Monthly report list, https://www.sec.gov/Archives/edgar/monthly/

23 XBRL International, Units Registry 1.0, https://specifications.xbrl.org/work-product-index-registries-units-registry-1.0.html

24 XBRL International, *Units*, http://www.xbrl.org/utr/utr.xml

arcroles, XBRL processors and/or XBRL Formula processors need to be extended to support this new functionality. XBRL International provides a global standard mechanism for extending XBRL Formula functionality[25].

[CSH: Need to define these extension functions. To do]

14. References

This section contains references to information that is useful for the creation of this specification. These references will not be in the final version of this specification.

Method of Implementing a Standard Digital Financial Report Using the XBRL Syntax:

http://xbrlsite.azurewebsites.net/2019/Library/MethodForImplementingStandardFinancialReportUsingXBRL.pdf

Open Source Framework for Implementing XBRL-based Digital Financial Reporting:

http://xbrlsite.azurewebsites.net/2019/Library/MethodForImplementingStandardFinancialReportUsingXBRL.pdf

[25] XBRL International, *Functions Registry 1.0*, https://specifications.xbrl.org/work-product-index-registries-functions-registry-1.0.html