COMMON APPROACH
TO IMPACT MEASUREMENT

The Common Form Ontology:

Mark Fox[1], Kate Ruff[2], Alicia Richins[3]

[1]Centre for Social Services Engineering
University of Toronto
csse.utoronto.ca

[2]Sprott School of Business
Carleton University

[3]Common Approach to Impact Measurement
commonapproach.org

Version 0.3, DRAFT: 5 April, 2022

Technical Report
Centre for Social Services Engineering
University of Toronto
csse.utoronto.ca

Ontology URL: http://ontology.eil.utoronto.ca/cids/cfo.owl

Namespace: http://ontology.eil.utoronto.ca/cids/cfo#

Suggested Prefix: cfo

For further information contact: Mark S. Fox, msf@eil.utoronto.ca

Recommended Attribution:
Fox et al. (2021) “The Common Form Ontology”, Centre for Social Services Engineering, University of Toronto.

Revision	Date	Changes
0.2	1apr2022	Added requestedFrom to FundingRequest
0.3	5 apr 2022	Changed ServedPopulation to BeneficiaryStakeholder

Table of Contents

1. Introduction

The Common Form is a standardized way to represent the organizational information of a social purpose organization (SPO).

The organizational information collected includes both financial measures and descriptive measures. The financial information collected includes financial statements and key financial indicators, such as current assets, total assets, total liabilities, total revenues and total expenses. The descriptive information collected includes data such as charity/business number, legal form, mission statement, team and board bios. It provides space to record information about the organization as well as programs and projects within that organization.

These measures were developed with extensive stakeholder engagement and are based on existing and common organizational profiles—and in the case of the financial measures, drawn from scholarly and practitioner literature.

It is the information ”used in most exchanges most of the time”

Created to contextualize the impact data collected using the Common Impact Data Standard, the Common Form facilitates the review, analysis and aggregation of data. This enables assessors to “compare fair,” so that geography, size and age could be considered when comparing impact data.

The Common Form allows software, databases and websites to:

- enable the representation of precise definitions of the organization
- foster data interoperability of information between organizations
- make the details of an organization interpretable by a computer
- empower your clients to apply for grants, loans or investments directly from their impact data’s home base

As we expand the roster of aligned impact management and grant/fund management softwares, the Common Form ontology will allow social purpose organizations who want to share their data, and the linked details, to do so at very little cost.

1.1. How to Read the Ontology Tables

The ontology specifies the “classes”, “properties” and possible restrictions on the “values” of a property within the context of a class. While it utilizes a simplified version of the Manchester syntax (Horridge et al., 2016) for Description Logic, formal definitions of classes are not provided, but can be found in the OWL files at http://ontology.eil.utoronto.ca/cids/cids.owl. The following table is an example. The class column specifies the name of the class being defined. This table describes the Organization class – it is a partial list for explanatory purposes. The class is defined to be a subclass of the **conjunction** of the properties in the property column. An organization is a subclass of owl:Thing. The property column specifies the properties used to define the class. Organization has properties including name, legal name, and address. The Value Restriction column specifies the constraints on the values of each property for the class. The value of the “name” property must be a string. The value of the “address” property must be at most one instance of the class ic:Address.

Class	Property (partial list for example)	Value Restriction
Organization	rdfs:subClassOf	owl:Thing
	hasName	only xsd:string
	sch:legalName	exactly 1 xsd:string
	ic:address	max 1 ic:Address

Table 1: Partial representation of the class Organization

The following value restrictions are used in this document:

- "min n": Specifies that the property has to have a minimum n values.
- "max n": Specifies that the property has to have a maximum n values.
- "exactly n": Specifies that the property has to have exactly n values.
- "only": Specifies that the values of the property can only be an instance/type of the class specified, e.g., a string, integer, or another class such as Organization.

We use camelCase for specifying classes, properties and instances. For example, "legalName" instead of "legal_name". The first letter of a class name is capitalized. The first letter of a property and instance name are not capitalized.

An instance of a class must satisfy the class's definition. The instance's properties and values must satisfy the value restrictions of the class it is an instance of. Table 2 defines an instance of Organization. acmeSocialServices is of rdfs:type Organization. rdfs:type signifies that it is an instance. The remaining properties provide additional information about acmeSocialServices. Note that if we left out the property sch:legalName, it would be an error as the definition of Organization above has a value restriction for the property sch:legalName of having exactly 1 xsd:string. If zero or more than one was specified, it would be an error. The value of ic:address is NOT a string, but an instance of ic:Address class. In this example, properties have prefixes which identify the complete URI (i.e., namespace or library where the property name originates from).

Instance	Property	Value
acmeSocialServices	rdfs:type	cids:Organization
	hasName	"Acme Social Services"
	sch:legalName	"Ontario 12345 Ltd."
	ic:address	acmeAddress
acmeAddress	rdfs:type	ic:Address
	ic:hasStreet	"Bloor"
	ic:hasStreetType	ic:street
	ic:hasDirection	ic:west
	ic:hasStreetNumber	0

Table 2: An example of an Instance of an Organization

Table 3 lists the prefixes used in this document.

Prefix	URI
act	http://ontology.eil.utoronto.ca/tove/activity#
cids	http://ontology.eil.utoronto.ca/CIDS/cids#
ic	http://ontology.eil.utoronto.ca/tove/icontact#
oep	http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/part.owl#
org	http://ontology.eil.utoronto.ca/tove/organization#
owl	http://www.w3.org/2002/07/owl#
rdfs	http://www.w3.org/2000/01/rdf-schema#
sch	http://schema.org/

Table 3: Prefixes used in this document and the connected URI

2. Core Ontology: Common Form

The cfo:Organization class extends the cids:Organization with Common Form information. The cids:Organization class is defined as a rfds:subClassOf org:Organization which provides the basic properties related to organization structure and behavior. cids:Organization extends org:Organization with the following properties:

- org:hasID: Limits an organization to one org:OrganizationID. Ties to a unique recognized/public identifier for the Organization i.e. a business number, charity number, etc.
- hasImpactModel: Identifies one or more ImpactModels for the organization.
- hasIndicator: List of indicators associated with the Organization.
- hasOutcome: List of outcomes associated with the Organization.
- hasCharacteristic: List of stakeholder characteristics associated with the Organization.
- hasContact: Identifies one or more people who are the contact for the Organization.
- hasDescription: a string that describes the Organization.
- hasWebAddress: a URI specifying the website for the organization.
- hasMission: text defining the mission or vision of the organization.
- hasNumberFullTimeEmployee: A non-negative integer that specifies the Organization's number of full-time employees
- hasNumberPartTimeEmployee: A non-negative integer that specifies the Organization's number of part-time employees
- hasNumberVolunteer: A non-negative integer that specifies the Organization's number of volunteers
- sch:dateCreated: A xsd:date the organization was created, in the format of yyyy-mm-dd.

The Charity Number or Business Number of the cids:Organization is specified using the org:hasID property as defined in org:Organization below.

Class	Property	Value Restriction
cids:Organization	rdfs:subClassOf	org:Organization
	org:hasID	exactly 1 org:OrganizationID
	hasImpactModel	only ImpactModel
	hasIndicator	only Indicator
	hasOutcome	only Outcome
	hasCharacteristic	only Characteristic
	hasContact	only cids:Person
	sch:description	exactly 1 xsd:string
	hasWebAddress	only xsd:anyURI
	hasMission	exactly 1 xsd:string
	hasNumberFullTimeEmployee	exactly 1 xsd:nonnegativeinteger
	hasNumberPartTimeEmployee	exactly 1 xsd:nonnegativeinteger
	hasNumberVolunteer	exactly 1 xsd:nonnegativeinteger

	sch:dateCreated	exactly 1 xsd:dateTime

cids:Organization reuses and extends the TOVE Organization Ontology (Fox et al., 1998). The TOVE Organization Ontology is accessible at https://ontology.eil.utoronto.ca/tove/organization.owl. The following properties of org:Organization are inherited by cids:Organization:

- org:hasID: specifies identifiers for the Organization. They may be a business number, charity number, etc.
- ic:has Address: The main address of the Organization.
- org:hasLegalName: A string that specifies the legal name of the Organization
- org:hasLegalStatus: An instance of a class that specifies the legal status of the Organization. The legal status will differ based on country.
- ic:hasTelephone: Main phone numbers of the Organization
- hasContact: identifies one or more people who are the contact for the Organization.
- org:consistsOf: An org:Organization may be divided into org:Divisions.

Class	Property	Value Restriction
org:Organization	org:hasID	only org:OrganizationID
	ic:hasAddress	only ic:Address
	org:hasLegalName	exactly 1 xsd:string
	org:hasLegalForm	only cids:Code
	ic:hasTelephone	only ic:PhoneNumber
	org:consistsOf	only org:Division
	org:hasContact	only Person
org:OrganizationID	org:issuedBy	org:Organization
	hasIdentifier	exactly 1 xsd:string
	sch:dateCreated	exactly 1 xsd:dateTime

Figure 1 depicts its primary object properties.

The FinancialInformation class has two subclasses: AuditedFinancialInformation and UnauditedFinancialInformation. The information required by each subclass differs and is defined below.

The FundingRequest class has two subclasses: ProjectFunding and OrganizationFunding. ProjectFunding defines information required in an application for funding the organization. ProjectFunding contains information required to fund a project being run by the organization. FundingRequest also has a property servesPopulation that identifies the population served by the funding request.

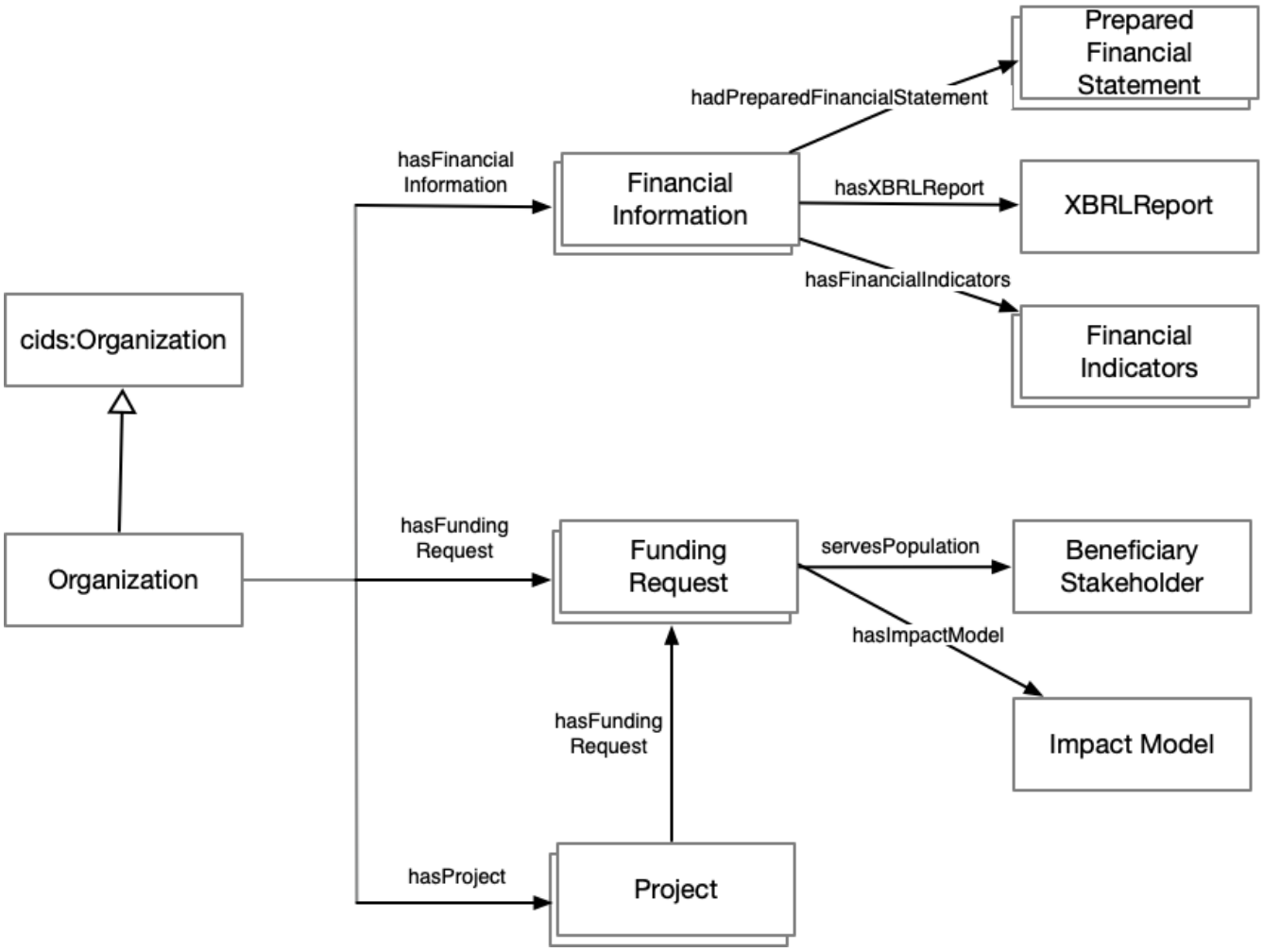

Figure 1: Organization Pattern

The cfo:Organization class is defined as a rfds:subClassOf cids:Organization which provides the basic properties related to organization structure and behavior and impact models. cfo:Organization extends cids:Organization with the following properties:

- hasFinancialInformation: Links to instances of FinancialInformation.
- hasFundingRequest: Links to instance of FundingRequest which specifies the funds requested.
- hasProject: Links organization to its projects.
- hasBios Links to a file containing the organization's executives' and Board's, or project/program biographies.
- servesPopulation: Links to one or more BeneficiaryStakeholder which define the stakeholders served.
- hasBylaws: Specifies the Organization's bylaws as text.
- hasIndigenousHeritage: Specifies the Organization's Indigenous heritages as text.
- hasPartners: Links to organizations it partners with.
- hasBusinessPlanFile: Links to a file containing the organization's business plan.
- imagineCanadaProgram: Specifies whether the organization participates in the Imagine Canada Standards program.
- hasBCorpScore: B-Corp score for the organization.
- hasBCorpProfile: Links to a file containing the organization's B-Corp profile.

- hasCharityDataProfile: Links to a file containing the organization's charity data profile.
- hasFundingRequest: Links to instances of FundingRequest.

The Charity Number or Business Number of the cids:Organization is specified using the org:hasID property as defined in org:Organization.

Class	Property	Value Restriction
Organization	rdfs:subClassOf	cids:Organization
	hasFinancialInformation	only FinancialInformation
	hasFundingRequest	only FundingRequest
	hasBios	only xsd:anyURI
	servesPopulation	only BeneficialStakeholder
	hasBylaws	only xsd:string
	hasIndigenousHeritage	only xsd:string
	hasPartners	only Organization
	hasProject	only Project
	hasBusinessPlanFile	only xsd:anyURI
	imagineCanadaProgram	exactly 1 {"yes", "no", "inProgress"}
	hasBCorpScore	exactly 1 xsd:decimal
	hasBCorpProfile	exactly 1 xsd:anyURI
	hasCharityDataProfile	exactly 1 xsd:anyURI

Table 4: Properties of Organization Class

The FinancialInformation class is used to specify the organization's financial information. It has the following properties:

- forYear: The year of the financial information.
- hasPreparedFinancialStatement: Links to the type of prepared statement.
- hasXBRLReport: Links to the instance of Organization's XBRL report.
- hasFinancialIndicators: Links to the instance of the Organization's financial indicators.

The FinancialStatement class is a subClassOf FinancialInformation, and captures the type of prepared financial statements for the organization.

The PreparededFinancialStatement class is a subClassOf FinancialInformation, and captures prepared financial statements for the organization. It has the following properties:

- hasType: Defines the type of prepared financial statements that the organization will share.
- forYear1: URI linking to the prepared statement for Year 1.
- forYear2: URI linking to the prepared statement for Year 2.
- forYear3: URI linking to the prepared statement for Year 3.

The XBRLReport class is a subClassOf FinancialInformation, and captures prepared XBRL reports for the organization. It has the following properties:

- forYear1: URI linking to the prepared report for Year 1.

- forYear2: URI linking to the prepared report for Year 2.
- forYear3: URI linking to the prepared report for Year 3.

The FinancialIndicators class is a subClassOf FinancialInformation, and captures financial indicators for the organization with no prepared financial statements or XBRL reports. It has the following properties:

- hasCurrentAssets: Current assets of the organization for the specified year in dollars.
- hasTotalAssets: Total assets of the organization for the specified year in dollars.
- hasTotalLiabilities: Total liabilities of the organization for the specified year in dollars.
- hasTotalRevenues: Total revenues of the organization for the specified year in dollars.
- hasTotalExpenses: Total expenses of the organization for the specified year in dollars.
- hasEndowment: Specifies the amount of the organization's endowment.
- hasInventory: Specifies the amount of the organization's inventory.
- hasDeferredRevenue: Specifies the amount of the organization's deferred revenue.
- hasSalesRevenue: Specifies the amount of the organization's sales revenue.
- hasGovernmentGrants: Specifies the amount of the organization's government grants.
- hasFundRaising: Specifies the amount of the organization's fundraising.
- hasTotalOverhead: Specifies the amount of the organization's total overhead.
- hasDepreciationExpense: Specifies the amount of the organization's depreciation expenses.
- hasInvestmentExpense: Specifies the amount of the organization's investment expenses.

Class	Property	Value Restriction
FinancialInformation	rdfs:subClassOf	owl:Thing
	forYear	exactly 1 xsd:date
	hasPreparedFinancialStatement	only PreparedFinancialStatement
	hasXBRLReport	only XBRLReport
	hasFinancialIndicators	exactly 1 FinancialIndicators
FinancialStatement	instances	{list of statement types}
PreparedFinancialStatement	cids:hasType	exactly 1 FinancialStatement
	forYear1	exactly 1 xsd:anyURI
	forYear2	exactly 1 xsd:anyURI
	forYear3	exactly 1 xsd:anyURI
XBRLReport	forYear1	exactly 1 xsd:anyURI
	forYear2	exactly 1 xsd:anyURI
	forYear3	exactly 1 xsd:anyURI
FinancialIndicators	hasCurrentAssets	exactly 1 xsd:decimal
	hasTotalAssets	exactly 1 xsd:decimal
	hasTotalLiabilities	exactly 1 xsd:decimal
	hasTotalRevenues	exactly 1 xsd:decimal
	hasTotalExpenses	exactly 1 xsd:decimal
	hasEndowment	exactly 1 xsd:decimal

	hasInventory	exactly 1 xsd:string
	hasDeferredRevenue	exactly 1 xsd:decimal
	hasSalesRevenue	exactly 1 xsd:decimal
	hasGovernmentGrants	exactly 1 xsd:decimal
	hasFundRaising	exactly 1 xsd:decimal
	hasTotalOverhead	exactly 1 xsd:decimal
	hasDepreciationExpense	exactly 1 xsd:decimal
	hasInvestmentExpense	exactly 1 xsd:decimal

The Project class specifies information about the organization's project when they request funding for a specific project or program. It has the following properties:

- hasProjectName: Name of the project for which funds are requested.
- hasProjectObjectives: Description of project objectives.
- hasProjectDescriptionFile: Link to file describing the project.
- hasProjectPartners: Links to partner organizations.
- hasBudgetFile: Link to file describing the project budget.
- hasFundingRequest: Links to the project;s funding request.

Class	Property	Value Restriction
Project	hasProjectName	exactly 1 xsd:string
	hasProjectObjectives	only xsd:string
	hasProjectDescriptionFile	exactly 1 xsd:anyURI
	hasProjectPartners	only cids:Organization
	hasBudgetFile	exactly 1 xsd:anyURI
	hasFundingRequest	only FundingRequest

The FundingRequest class specifies information about the organization or project's request for funding. It has the following properties:

- fundsRequested: Specifies the amount of funds requested for the organization or project.
- fundsReceived: Specifies amount awarded by funding agency.
- fundsIntension: Specifies the intended use of the funds, such as for the organization or a project/program.
- hasImpactModel: Links to a CIDS impact model instance.
- hasImpactModelFile: Links to an impact model defined in a file.
- servesGeographicRegion: A string that specifies the region.
- servesMarket: A string that specifies the market to be served.
- servesPopulation: Links to one or more BeneficiaryStakeholder which define the stakeholders served.
- requestedFrom: One or more organizations that the funds are requested from.

Class	Property	Value Restriction

FundingRequest	rdfs:subClassOf	owl:Thing
	fundsRequested	exactly 1 xsd:decimal
	fundsReceived	exactly 1 xsd:decimal
	fundsIntension	exactly 1 {“organization”, “project”}
	cids:hasImpactModel	only cids:ImpactModel
	hasImpactModelFile	only xsd:anyURI
	servesGeographicRegion	only xsd:string
	servesMarket	only xsd:string
	servesPopulation	only cids:BeneficialStakeholder
	requestedFrom	only cids:Organization

3. Acknowledgements

This work was supported, in part, by the Ontario Ministry of Economic Growth and Development and Employment and Social Development Canada.

4. References

Fox, M.S., Barbuceanu, M., Gruninger, M., and Lin, J., (1998), “An Organisation Ontology for Enterprise Modeling”, In *Simulating Organizations: Computational Models of Institutions and Groups*, M. Prietula, K. Carley & L. Gasser (Eds), Menlo Park CA: AAAI/MIT Press, pp. 131-152.

Fox et al. (2021) “The Common Impact Data Standard V2.0 An Ontology for Representing Impact”, Centre for Social Services Engineering, University of Toronto.