CACO: The Common Approach Core Ontology for Modeling Theories of Change

Mark Fox[1], Anshula Chowdhury[2], Jane Zhang[3], Bart Gajderowicz[1], Tawfiq Abdulai[2], Kate Ruff[4], Daniela Rosu[1]

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

[2]Sametrica
sametrica.com

[3]Centre for Social Innovation
socialinnovation.org

[4]Carleton Centre for Community Innovation
Carleton University

Updated: 15 November 2019

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

Ontology URL: http://ontology.eil.utoronto.ca/CAO/caco.owl

Namespace: http://ontology.eil.utoronto.ca/CAO/caco#

Suggested Prefix: caco

Table of Contents

1. Introduction

This document defines the extensions to the Common Approach Foundation Ontology (CAFO) to enable the representation and interoperability of various Theories of Change (ToC). The Common Approach Core Ontology (CACO) provides a core set of concepts that span all variations of the ToC, making it possible for SEs to share data amongst them regardless of the version of the ToC they use. While the core concepts are adequate, for each ToC, specializations of the core concept may be required to model the different variations of the ToC.

2. Common Approach Core Ontology (CACO)

The top-level classes of the Common Approach Core Ontology are depicted in Figure 3. The main classes in CACO are grouped according to the five dimensions described by the Impact Management Project (IMP, 2019):

- **What**: what outcomes the enterprise is contributing to and how important the outcomes are to stakeholders.
    - *Outcome* Class (Section 3.1)
- **Who**: which stakeholders are experiencing the outcome and how underserved they were prior to the enterprise's effect.
    - *Stakeholder* Class (Section 3.3)
- **How Much**: how many stakeholders experienced the outcome, what degree of change they experienced, and how long they experienced the outcome for.
    - *Indicator* and *IndicatorReport* Classes (Section 3.4)
- **Contribution**: whether an enterprise's and/or investor's efforts resulted in outcomes that were likely better than what would have occurred otherwise.
    - *Indicator* and *IndicatorReports* Classes (Section 3.4)
- **Risk**: the likelihood that impact will be different than expected.
    - *Risk* Class (Section 3.5)

In order to represent the processes by which a social purpose organization delivers Outcomes to its Stakeholders, we add a sixth dimension:

- **How**
    - *Program* Class (Section 3.6)
    - *Service* Class (Section 3.7)
    - *Activity* Class (Section 3.8)
    - *Input* Class (Section 3.9)
    - *Output* Class (Section 3.10)

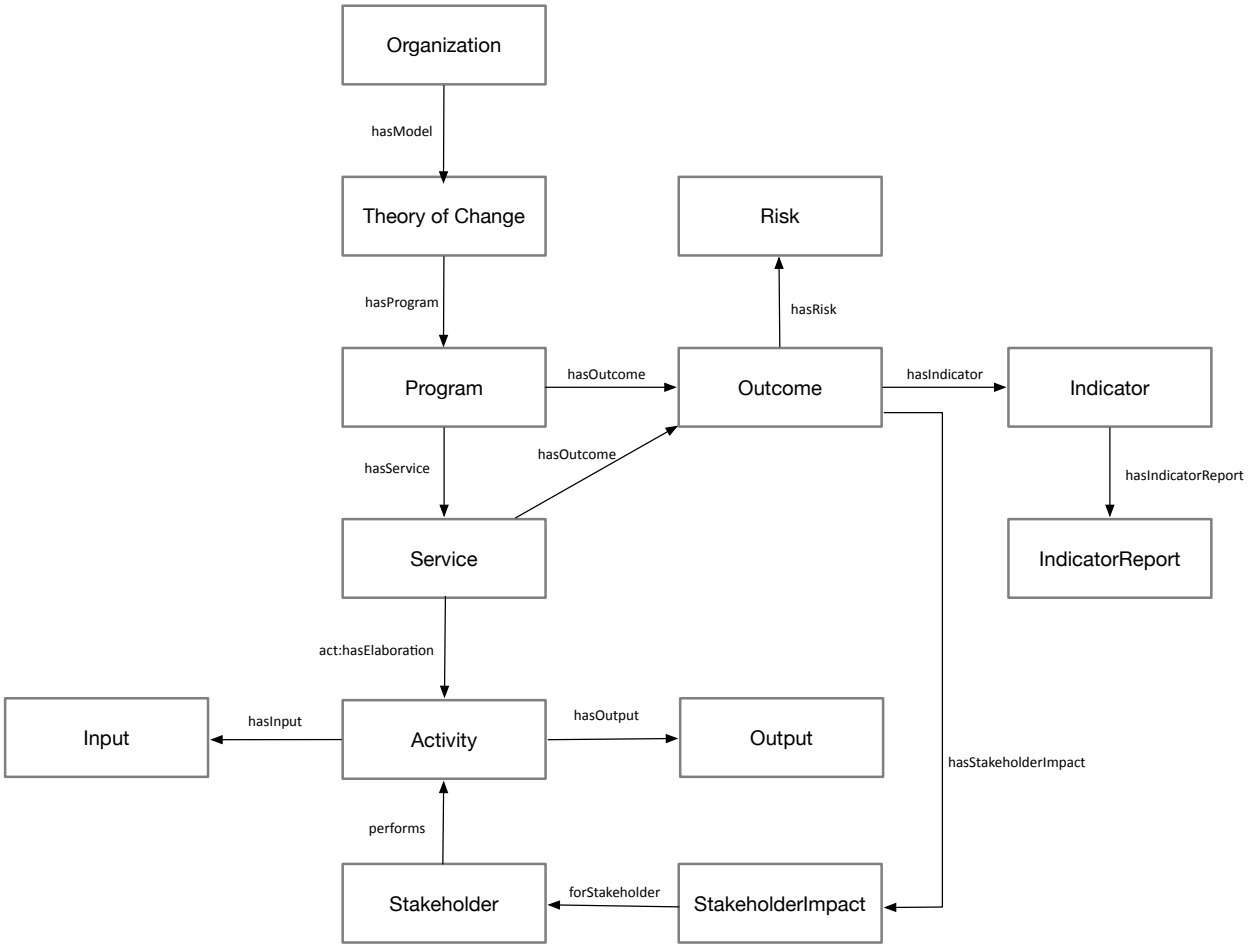

Figure 1: Top Level CACO

The classes defined in CACO can be used to map concepts from one version of the Theory of Change to another. For example, the Logic Model version of the theory of change contains inputs, activities, and outcomes. Each activity changes one state (where outcome is false) to another state (where outcome is true). In the Logic Model, there is no explicit process or sequence of events. The Outcomes Chains version of the theory of change begins with the final goal (outcome). An activity is added when it is required to make that goal come true. When an added activity requires some input, another activity that produces that input is added before the activity requiring it. Using this process, an explicit sequence of activities is created, where the completion of one activity enables the next activity to be executed. An activity's required inputs are the enabling states, while the activity's outcomes are the caused states. When the caused state makes the original goal true, it is the last activity in the sequence. In the CACO representation of this interpretation of the logic model , Indicators are used to measure Outcomes.

This document defines core classes required to model various theories of change. Specializations of these concepts are defined in the following documents:

- CACO-ACT: The Common Approach Core Ontology Activity Extension
- CACO-FIN: The Common Approach Core Ontology Finance Extension
- CACO-ORG: The Common Approach Core Ontology Organization Extension
- CACO-OUT: The Common Approach Core Ontology Outcome Extension
- CACO-PER: The Common Approach Core Ontology Person Extension

An example of how to represent a Logic Model and an Impact Chain can be found in:

- Representing Theories of Change Using the Common Approach Ontologies

The CACO classes included in this document do not represent a list of required classes that must be provided by a social purpose organization, nor is it an extensive list of all classes that may exists. A social purpose organization should only provide classes that exist in their specific organization or overall domain.

3. CACO Classes

In this section we defined the classes that comprise the Common Approach Core Ontology (CACO) (Figure 3). CACO classes are defined in terms of the Common Approach Foundation Ontology (CAFO) and the Common Approach Repository Vocabulary (CARV). Properties that are defined in the CARV are denoted in blue. All of these properties have string values and some of them have an equivalent version that uses classes as values, allowing for the maintenance of both the text and classes. The ontology is defined in Description Logic and published using the Semantic Web ontology language OWL.

The CACO incorporates several ontologies beyond CAFO. We define the following prefixes that are used in the remainder of this report.

Prefix	URI
act	http://ontology.eil.utoronto.ca/tove/activity#
cafo	http://ontology.eil.utoronto.ca/CAO/cafo#
cav	http://ontology.eil.utoronto.ca/CAO/cav#
do	https://raw.githubusercontent.com/AGLDWG/dataset-ont/master/dataset#
gcif	http://ontology.eil.utoronto.ca/GCI/Finance/GCI-Finance#
gn	http://www.geonames.org/ontology/ontology_v3.1#
i72	http://ontology.eil.utoronto.ca/ISO21972/iso21972#
oep	http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/part#
org	http://ontology.eil.utoronto.ca/tove/organization#

sur	http://ontology.eil.utoronto.ca/tove/survey#
time	https://www.w3.org/2006/time#

3.1. TheoryOfChange

CACO is designed to represent the different versions of theories of change in use by Social Impact Organizations. The TheoryOfChange class is the root of a taxonomy of theories of change, two of which are included here: Logic Model and Impact Chain. The properties of each reflect the differences in focus and level of detail each theory focuses on.

Class	Property	Value Restriction
TheoryOfChange	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	sch:dateCreated	exactly 1 yyyy-mm-dd
	forOrganization	exactly 1 Organization

LogicModel is a subclass of TheoryOfChange. The top level definition of a LogicModel contains the following properties:

- hasProgram identifies all of the Programs being modeled.
- hasStakeholder identifies key stakeholders participating in the model
- hasOutcome identifies key Outcomes for the model
- hasInput identifies key Inputs for the model
- hasOutput identifies key Outputs for the model
- hasActivity identifies key Activities of the model
- hasResource identifies key Resources of the model

By “key” we mean a subset of each that are important to depict at this level of abstraction.

Class	Property	Value Restriction
LogicModel	rdfs:subClassOf	TheoryOfChange
	hasProgram	only Program
	hasOutcome	only Outcome
	hasStakeholder	only Stakeholder
	hasInput	only Input
	hasOutput	only Output
	hasActivity	only Activity

	hasResource	only Resource

ImpactChain is a subclass of TheoryOfChange. The top-level definition of an ImpactChain contains the following properties:

- hasOutcome: identifies key Outcomes for the model
- hasActivity: identifies key Activities of the model
- hasIndicator: identifies the key Indicators of the model
- By "key" we mean a subset of each that are important to depict at this level of abstraction. Note that it uses a small subset of the CACO ontology as its focus is on an abstraction of the theory of change.

Class	Property	Value Restriction
ImpactChain	rdfs:subClassOf	TheoryOfChange
	hasOutcome	only Outcome
	hasActivity	only Activity
	hasIndictor	only Indicator

3.2. Outcome, Impact, StakeholderImpact

As defined by the Impact Measurement Project, Outcomes "are what stakeholders experience as a result of an enterprise's activities. They can be positive or negative, intended or unintended." CACO captures the key properties of an outcome by representing:

- hasStakeholderImpact: identifies the Stakeholder and the impact it has on them
- forDomain: identifies the Domains that the Outcome aligns with
- hasIndicator: identifies the set of Indicators the organization assigns to the Outcome
- canEnable: links an Outcome to a Service or Activity that is made possible due to the result of the Outcome. It abstracts a more detailed specification of Activities producing States that enable other activities. It is used primarily for representing Impact Chains.
- canProduce: Links an Outcome to another Outcome. It abstracts the underlying activity chain that usually links one Outcome to another. It is used primarily for representing Impact Chains.
- oep:partOf: identifies the Theory of Change it is a component of
-

Classes in blue are carried over from the specification of an Outcome in "Common Approach Repository and Vocabulary" (CARV).

Class	Property	Value Restriction
Outcome	rdfs:subClassOf	cav:Outcome

	hasStakeholderImpact	only StakeholderImpact
	hasIndicator	only Indicator
	canEnable	only (Service or Activity)
	canProduce	only Outcome
	oep:partOf	exactly 1 TheoryOfChange
	sch:identifier	exactly 1 xsd:string (begins with OUT)
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	forDomain	only Domain
	definedBy	exactly 1 xsd:string (begins with ORG)
	sch:dateCreated	exactly 1 yyyy-mm-dd
Domain	owl:equivalentClass	{sdg1, sdg2, sdg3}

The StakeholderImpact class specifies for a specific Stakeholder whether the Outcome will have a positive, negative or neutral affect on the Stakeholder, and the nature of the impact:

- forStakeholder: identifies the Stakeholder affected
- intendedImpact: identifies the intended direction of the change – note that Indicators capture the actual direction.
- produces: identifies the Impact on the Stakeholder.
- hasIndicator: identifies the set of Indicators the Organization assigns to the Outcome but are specific to this Stakeholder

The Impact class specifies:

- forStakeholder: identifies the Stakeholder affected
- fromPerspectiveOf: identifies the Stakeholder who is determining the importance of the Impact
- hasImportance: specifies the nature of the importance as a om:Quantity. Users can define or select whatever quantification they prefer

Class	**Property**	**Value Restriction**
StakeholderImpact	forStakeholder	exactly 1 Stakeholder
	intendedImpact	exactly 1 of {positive, negative, neutral}
	producesImpact	some Impact
	hasIndicator	only Indicator

Impact	forStakeholder	exactly 1 Stakeholder
	fromPerspectiveOf	exactly 1 Stakeholder
	hasImportance	exactly 1 om:Quantity

3.3. Stakeholder

The Stakeholder class is a subclass of an Organization or Person. It identifies what activities they perform using the *perform* property, where they are located geographically using the *locatedIn* property, and the specific Theory of Change the stakeholder is a part of. Some methods for measuring impact, such as a Social Return on Investment, require that social purpose organizations distinguish between beneficiary and contributing stakeholders. A Beneficiary_Stakeholder is a stakeholder that benefits from a logic model's outcome. A Contributor_Stakeholder is a stakeholder that contributes input to ensure a service can produce outcomes. These are explicitly defined as separated classes to highlight the fact that a contributor stakeholder is not always the same instance as the beneficiary stakeholder. The distinction is available in the ontology, but is not required.

Class	**Property**	**Value Restriction**
Stakeholder	rdfs:subClassOf	(cafo:Organization or cafo:Person)
	performs	some Activity
	i72:located_in	only i72:Feature
	oep:partOf	only TheoryOfChange
BeneficialStakeholder	rdfs:subClassOf	Stakeholder
	beneiftsFrom	min 1 Outcome
	org:hasRole	only (Client or Employee)
ContributingStakeholder	rdfs:subClassOf	Stakeholder
	benefitsFrom	some Outcome
	contributes	min 1 Input
Client	rdfs:subClassOf	org:Role
Employee	rdfs:subClassOf	org:Role

Additional properties for Stakeholders can be inherited from either Organization (specified in the "Common Approach Core Ontology Organization Extension" (CAFO-O) or Person (specified in the "Common Approach Core Ontology Person Extension" (CAFO-P).

3.4. Indicator and IndicatorReport

In this section we reproduce the Indicator class from the Common Approach Indicator Vocabulary and Repository (CARV) specification. We extend it with properties that integrate it with CACO. In particular we the following properties:

- forOutcome: Identifies the Outcome's the Indicator measures. Note this links to the individuals of the Outcome class, and not the string name (sch:identifier) of an Outcome
- hasIndicatorDefinition: provides the machine readable definition of the Indicator, which includes data type, numerator/denominator if a ratio, etc.
- hasMethod: specifies the method by which the Indicator was derived (not a string)
- hasDataSource: specifies the Datasets used to derive the value if the method is Computation
- hasThreshold: specifies a threshold Quantity that the Indicator should be above or below

The IndicatorDefinition class explicitly defines what "things" it measures. They may include Stakeholders, Organizations, activities, etc. Properties in blue are inherited from cav:Indicator. Properties in green are inherited from cafo:Indicator

Class	Property	Value Restriction
Indicator	rdfs:subClassOf	cafo:Indicator
	forOutcome	only Outcome
	hasThreshold	exactly 1 i72:Quantity
	hasMethod	exactly 1 (quest:Survey or Interview or Estimate or Computation)
	hasDataSource	only Dataset
	hasIndicatorReport (inverse forIndicator)	only IndicatorReport
	forOrganization	exactly 1 Organization
	i72:unit_of_measure	exactly 1 i72:Unit_of_measure
	i72:value	exactly 1 i72:Measure
	i72:for_time_interval	exactly 1 time:DateTImeInterval
	sch:identifier	exactly 1 xsd:string (begins with IND)
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	definedBy	exactly 1 xsd:string (begins with ORG)
	forOutcomeS	only xsd:string (begins with OUT)
	hasSimilarIndicator	only xsd:string (begins with SI)
	sch:dateCreated	exactly 1 yyyy-mm-dd

We include the IndicatorReport from CAV which is used to report measures of an Indicator by an Organization. IndicatorReport is extended with object properties:

- forOrganization: identifies the Organization that submits the report (not a string)
- forIndicator: identifies the Indicator that is being reported (not a string)
- hasValue: specifies the value as a om:Quantity (not a string)
- hasMethod: specifies the method by which the Indicator was derived (not a string)
- hasDataSource: specifies the Datasets used to derive the value if the method is Computation
- time:hasTime: specifies the time interval that the Indicator Report covers

Class	Property	Value Restriction
IndicatorReport	rdfs:subClassOf	cav:IndicatorReport
	forOrganization	exactly 1 cafo:Organization
	forIndicator	exactly 1 Indicator
	hasValue	exactly 1 i72:Quantity
	hasMethod	exactly 1 (quest:Survey or Interview or Estimate or Computation)
	hasDataSource	only Dataset
	time:hasTime	only time:DateTimeInterval
	sch:identifier	exactly 1 xsd:string (begins with IR)
	sch:name	exactly 1 xsd:string
	forSIndicator	exactly 1 xsd:string (begins with IND)
	forSOrganization	exactly 1 xsd:string (begins with ORG)
	forYear	exactly 1 yyyy
	hasSValue	exactly 1 xsd:string
	sch:dateCreated	exactly 1 yyyy-mm-dd

An IndicatorReport can designate the source of the data used in generating the report. Dataset provides basic information about the datasets used. It has the following properties:

- dc:spatial: specifies the spatial area that the dataset covers
- dc:temporal: specifies the time interval that the dataset covers
- sch:identifier: specifies the unique ID of the dataset, if any
- sch:name: specifies the name of the dataset
- sch:description: provides a description of the dataset
- sch:dateCreated: specified the date the dataset was created

Class	Property	Value Restriction
Dataset	rdfs:subClassOf	dcat:Dataset
	dc:spatial	only (dc:Location or i72:Feature)
	dc:temporal	only (dc:PeriodOfTime or time:DateTimeInterval)
	sch:identifier	exactly 1 xsd:string

	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	sch:dateCreated	exactly 1 yyyy-mm-dd

3.5. Risk

Risk "assesses the likelihood that impact will be different than expected, and that the difference will be material from the perspective of people or the planet who experience impact." Stating the riskiness of the impact is important for interpreting the subsequent results. The Impact Management Project recommends that as part of any impact assessment, the risk of the impact be considered as one of the five dimensions of performance.

The following defines the taxonomy of risk and key properties:

- hasLikelihood: identifies the likelihood that the risk will occur
- hasConsequence: identifies the degree of impact the risk could have
- hasMitigation: a string that specifies a mitigation plan or references a document

Note that the subclasses of risk do not have properties that distinguish one from another. These would be provided in later versions, as needed.

Class	Property	Value Restriction
Risk	hasLikelihood	exactly 1 {veryUnlikely , unlikely, likely, veryLikely}
	hasConsequence	exactly 1 {minimal, average, severe}
	hasMitigation	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	sch:identifier	exactly 1 xsd:string
EvidenceRisk	rdfs:subClassOf	Risk
ExternalRisk	rdfs:subClassOf	Risk
StakeholderParticipationRisk	rdfs:subClassOf	Risk
DropOffRisk	rdfs:subClassOf	Risk
EfficiencyRisk	rdfs:subClassOf	Risk
ExecutionRisk	rdfs:subClassOf	Risk
AlignmentRisk	rdfs:subClassOf	Risk
EnduranceRisk	rdfs:subClassOf	Risk

UnexpectedImpactRisk	rdfs:subClassOf	Risk

3.6. Program

A program defines a set of services that focus on a shared set of Outcomes. For example, a "poverty reduction program" can be made up of a set of Services such as mobiles services that provides food and clothing to those that live on the street, and a training service that provides basic skills for those living on the street. A Program has a set of Stakeholders that my contribute or benefit:

- hasService: identifies the Services that make up the Program
- hasOutcome: identifies the Outcomes that the program is trying to achieve
- hasContributingStakeholder: identifies the stakeholders that contribute to the Program
- hasBeneficialStakeholder: identifies the stakeholders that benefit from the Program
- hasInput: identifies the Inputs to the Program
- hasOutput: identifies the Outputs of the Program

Class	Property	Value Restriction
Program	rdfs:subClassOf	Activity
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	hasService	only Service
	hasOutcome	only Outcome
	hasContributingStakeholder	only ContributingStakeholder
	hasBeneficialStakeholder	only BeneficialStakeholder
	hasInput	Input
	hasOutput	Output

3.7. Service

A Program is composed of one or more Services. As described in the Program description, a poverty reduction program can have many services with each service comprised of different activities, Inputs, Outputs and Outcomes.

- act:hasSubActivity: identifies the Activities that make that comprise the Service
- hasInput: identifies the Inputs to the Service
- hasOutput: identifies the Outputs of the Service

- hasOutcome: identifies the Outcomes that are specific to the Service
- hasContributingStakeholder: identifies the stakeholders that contribute to the Service
- hasBeneficialStakeholder: identifies the stakeholders that benefit from the Service

Class	Property	Value Restriction
Service	rdfs:subClassOf	Activity
	eop:partOf	only TheoryOfChange
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
	act:hasElaboration	only Activity
	hasInput	only Input
	hasOutput	only Output
	hasOutcome	only Outcome
	hasContributingStakeholder	only ContributingStakeholder
	hasBeneficialStakeholder	only BeneficialStakeholder

3.8. Activity

Activity defines the actual "actions" performed by an organization to implement a Service. CACO's Activity class is defined to be a subclass of the TOVE Activity, and is extended by including properties for Input, Output and what Service or Activity it is a subActivityOf. An activity's type is based on its outcome rather than service or input. This allows activities to be classified by the type of change they produce rather by what resources they use (input) or who performs the activity (service). Its properties are:

- canProduce: specifies the Outcome that results from performance of the Activity. It is used primarily for representing Impact Chains.
- hasInput: specifies the Input to the Activity
- hasOutput: specifies the Output of the Activity
- hasCode: specifies zero or more codes, created by various organizations, to identify a type of Activity, e.g., ICHI – International Classification of Health Interventions activities
- act:subActivityOf: specifies the Service or Activity that this Activity is part of

An instance of an ActivityCode is used to specify a particular code created by a standards organization. Its properties are:

- forOrganization: specifies the Organization that created the code, e.g., ICHI, ISO, …

- sch:identifier: specifies the code number
- sch:name: specifies the name or title of the code
- sch:description: specifies the description of the code

Class	Property	Value Restriction
Activity	rdfs:subClassOf	act:Activity
	canProduce	only Outcome
	eop:partOf	some TheoryOfChange
	hasInput	only Input
	hasOutput	only Output
	hasCode	only ActivityCode
	act:subActivityOf	only (Service or Activity)
ActivityCode	forOrganization	exactly 1 cafo:Organization
	sch:identifier	exactly 1 xsd:string
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string

3.9. Input

A key component of theories of change are Inputs. Inputs specify the resources required by a social purpose organizationsocial purpose organization to produce results (Ralser, 2008). An Input is provided by a contributing stakeholder and may come in many forms. We identify three broad categories of Input:

- FinancialInput represents a monetary resource, with a monetary unit of measure, such as donating cash or paying off debt.
- SkillInput is any type of skills-based expertise such as legal, translation, carpentry, etc.
- PhysicalInput is any type of physical item, such as food, clothing, furniture, etc.

Properties common across all types of Input are:

- eof:partOf: specifies the theory of change this Input is part of
- requiredBy: specifies the Program, Service or Activity that uses or consumes this Input
- contributedBy: specifies the Stakeholders provide it
- hasType: specifies the type of Resource by denoting the relevant subclass of Resource
- hasAmount: specifies the Quantity of Input (which in turn specifies the unit of measure)
- time:hasTime: specifies the time interval over which the Input is provided

Class	Property	Value Restriction
Input	eof:partOf	exactly 1 TheoryOfChange
	inputFor	only (Program or Service or Activity)
	hasContributingStakeholder	only ContributingStakeholder
	hasType	exactly 1 Resource
	hasPlannedAmount	exactly 1 i72:Quantity
	hasAmount	exactly 1 i72:Quantity
	time:hasTime	only time:DateTimeInterval
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string
FinancialInput	rdfs:subClassOf	Input
	hasType	only FinancialResource
	hasAmount	exactly 1 (i72:Quantity and i72:unit_of_measure i72:Monetary_unit)
SkillInput	rdfs:subClassOf	Input
	hasType	only SkillResource
PhysicalInput	rdfs:subClassOf	Input
	hasType	only PhysicalResource

3.10. Output

Outputs are a quantitative summary of an activity. For example, if the activity is 'we provide training' and the output is 'we trained 50 people to NVQ level 3' (CED, 2012). Or a production output could produce 100 meals for the homeless. Basic to these outputs is "what" has been produced and the quantity.

- forActivity: identifies the Activity or Service that produces the Output
- hasQuantity: identifies that amount that is produced
- produces: identifies the Resource that is produced such as a skill, or a type of Meal

Class	Property	Value Restriction

Output	eop:partOf	some TheoryOfChange
	forActivity	only (Service or Activity)
	hasAmount	only i72:Quantity
	produces	only Resource
	sch:name	exactly 1 xsd:string
	sch:description	exactly 1 xsd:string

4. Acknowledgements

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

5. References

CED, (2009), "A guide to Social Return on Investment", The Canadian CED Network.

Fox, M.S., (2013), "A Foundation Ontology for Global City Indicators", Working Paper, Enterprise Integration Laboratory, University of Toronto, Revised: 13 October 2017

Fox, M.S., "The role of ontologies in publishing and analyzing city indicators", *Computers, Environment and Urban Systems*, Vol. 54, pp. 266–279. http://eil.mie.utoronto.ca/wp-content/uploads/2015/06/fox-ceus15.pdf.

Fox, M.S. (2019). Common Approach Base Indicator *Ontology and Repository*. Toronto, Canada.

Funnell, S. C., & Rogers, P. J. (2011). Purposeful program theory: Effective use of theories of change and logic models, Vol. 31. John Wiley & Sons.

Gajderowicz, B., Fox, M. S., & Grüninger, M. (2018). Ontology of social service needs: Perspective of a cognitive agent. In *Proceedings of the 2018 Joint Ontology WOrkshops, Cognition And OntologieS + Explainable AI* (pp. 1–12). Cape Town.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In *Proceedings of the international journal of human-computer studies*. Padova, Italy.

Grüninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies. In *International Joint Conferences on Artificial Intelligence Workshop on Basic Ontological Issues in Knowledge Sharing*.

Harries, E., Hodgson, L., and Noble, J., (2014), "Creating Your Theory of Change, NPC's Practical Guide", New Philanthropy Capital, London. https://www.thinknpc.org/resource-hub/creating-your-theory-of-change-npcs-practical-guide/

IMP, "What is Impact", Impact Management Project. https://impactmanagementproject.com/impact-management/what-is-impact/

Nicholls, J., Lawlor, E., Neitzert, E., and Goodspeed, T., (2012), "A Guide to Social Return on Investment", The SROI Network. http://www.socialvalueuk.org/app/uploads/2016/03/The%20Guide%20to%20Social%20Return%20on%20Investment%202015.pdf

Ralser, T. (2008). Organizational Value/Nonprofit ROI. *ROI For Nonprofits: The New Key to Sustainability* (pp. 51–66). Hoboken, New Jersey: Wiley.