

# Compass Event, Client, and Service Ontology: A Design Pattern for Social Services

Bart Gajderowicz<sup>1</sup>, Daniela Rosu<sup>1</sup> and Mark S. Fox<sup>1</sup>

<sup>1</sup>*Department of Mechanical & Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Ontario, M5S 3G8, Canada*

## Abstract

This paper introduces the Compass ontology and proposes a design pattern aimed at supporting the formal representation of information about clients, services, and events in a social service setting. The framework captures entities required to represent the interaction between clients and services and events that track progress through changes exhibited by these stakeholders. The Compass ontology is an extension of the Common Impact Data Standard. It introduces classes and properties that allow it to answer a set of competency questions posed by subject matter experts in the area of social service provisioning. We evaluate the ontology using sample competency questions implemented in SPARQL.

## Keywords

ontology, semantics and reasoning, data model, decision support systems, health services, social services,

## 1. Introduction

As cities grow in size and complexity, it becomes increasingly difficult to identify whether the right city services are being provided to the right citizens and at the right time. This paper introduces the Compass ontology and its design pattern aimed at supporting a formal representation of information about clients, services, and events in a social service setting. Our methodology behind design decisions is application agnostic while allowing the representation of a variety of service providers and their clients, and accommodating different levels of information, from simple text descriptions and enumerated vocabularies to a rich representation of concepts and their relationships. Conceptual analysis is provided in Fox et al. [1].

Organizations delivering social services are referred to as social purpose organizations (SPO) and are made up of charities, non-profits, and other social-purpose businesses [2]. Each organization is responsible for provisioning a myriad of services to clients across a spectrum of demographics and levels of need. For example, in Canada, there are over 170,000 such organizations, while the United States has over 1.54 million. In such activity networks, complementary stakeholders are related by a shared outcome, playing the role of a beneficiary or a contributor. In complex societies, services are categorized along many dimensions, such as housing and education, as well as their contributions toward shared outcomes. Clients are differentiated across dimensions such as age, gender, and medical conditions.

*The Eighth Joint Ontology Workshops (JOWO'22), August 15-19, 2022, Jönköping University, Sweden*

✉ bartg@mie.utoronto.ca (B. Gajderowicz); drosu@mie.utoronto.ca (D. Rosu); msf@mie.utoronto.ca (M. S. Fox)

🌐 <http://bartg.org> (B. Gajderowicz); <http://eil.utoronto.ca> (M. S. Fox)

🆔 0000-0001-6201-8781 (B. Gajderowicz); 0000-0002-5877-9681 (D. Rosu); 0000-0001-7444-6310 (M. S. Fox)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

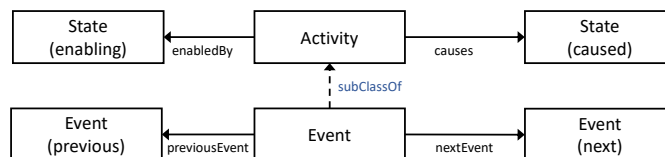
 CEUR Workshop Proceedings (CEUR-WS.org)

Recently, there have been efforts to define how to measure the impact of SPOs [3], including the Impact Management Project<sup>1</sup> that defines impact across five dimensions: *what*, *who*, *how much*, *contributions*, and *risk*. The Common Impact Data Standard (CIDS) extends this work with the sixth dimension *how*, to capture the process by which services deliver outcomes to stakeholders [2]. However, the broad categories used for social service provisioning are no longer sufficient [4]. A deeper representation of stakeholder characteristics, their needs, and the impact services have on stakeholders is required. This paper defines such a representation as an extension to CIDS, namely the Compass ontology<sup>2</sup>.

An ontological representation of service impact has long been a topic of interest for service providers, clients, and contributors like funding organizations. Over the years, various impact models have been developed for this purpose, including Logic Models, Theory of Change, Outcome Map, Outcome Chains, and Impact Map, each one having a different representation of how to model SPO impact [2]. Many ontological artifacts focus on the needs of clients [5, 6], social services [7, 8], and other city-level services such as transportation and hydro [9]. In addition, a foundational evaluation of related artifacts exists, including organizations [10, 11], their services [12, 13], as well as abstractions like business processes [14] and activities [15, 16].

The Compass ontology, presented here, extends the Common Impact Data Standard (CIDS) [17, 2], which provides classes and properties for representing impact models. For example, a “poverty reduction program” can be made up of services such as “mobile services” that provide food and clothing to those experiencing homelessness or a training service that provides employment skills. In previous work [1], we introduced a set of competency questions and use cases compiled by subject matter experts<sup>3</sup> and defined the scope for the Compass ontology presented here. Sections 2, 3, and 4 define three of the main Compass classes that extend the CIDS ontology, namely event, client, and service. The complete need pattern referenced here is provided in [18]. In section 5 we evaluate the extension by showing sample SPARQL queries and results generated from a knowledge graph built using the Compass ontology<sup>4</sup>. In section 6 we conclude by evaluating the results and discussing the state of the ontology and future work.

## 2. Event Pattern



**Figure 1:** Event class extension of the *5087-1:Activity* class.

An Event describes something that occurs to a stakeholder as an “event” in their life, identifying the location and time at which it occurred. It can describe the client as being the subject of an action or a change of state. The Compass *Event* class extends the *5078-1:Activity* class

<sup>1</sup>Impact Management Project: <https://impactmanagementproject.com>

<sup>2</sup>The Compass ontology was developed with, and is being used by HelpSeeker: <https://helpseeker.co>.

<sup>3</sup>Subject matter expert requirements and use-cases were collected by HelpSeeker: <https://helpseeker.org>.

<sup>4</sup>Code available at <https://github.com/csse-uoft/jowo-2022>.

defined in the 5078-1 ISO Standard [19]. The Activity class describes something that occurs in the domain and changes one state to another. It identifies an “enabling state,” defined as what state must be true of the “world” for the Activity to be performed and the “caused state,” defined as what must be true of the “world” once the Activity is completed. The Activity class also defines properties for preconditions and post-conditions to allow for automatically inferring a sequence of activities. The basic Compass *Event* class extends the activity class with the following properties:

- **hasName:** a unique name/identifier for the event
- **hasDescription:** descriptions of the event
- **occursAt:** time interval during which the event occurs
- **hasLocation:** placename where the event occurred
- **previousEvent:** link to previous related event, if any
- **nextEvent:** link to next related event, if any

**DL 1 (Compass Event Class).**

$$\begin{aligned}
 Event &\sqsubseteq 5087-1:Activity \\
 &\sqcap =1 \text{ hasName.xsd:string} \quad \sqcap \forall \text{ hasDescription.xsd:string} \\
 &\sqcap =1 \text{ occursAt.time:Interval} \quad \sqcap =1 \text{ hasLocation.i72:Feature} \\
 &\sqcap =1 \text{ previousEvent.Event} \quad \sqcap =1 \text{ nextEvent.Event}
 \end{aligned}
 \tag{DL-1}$$

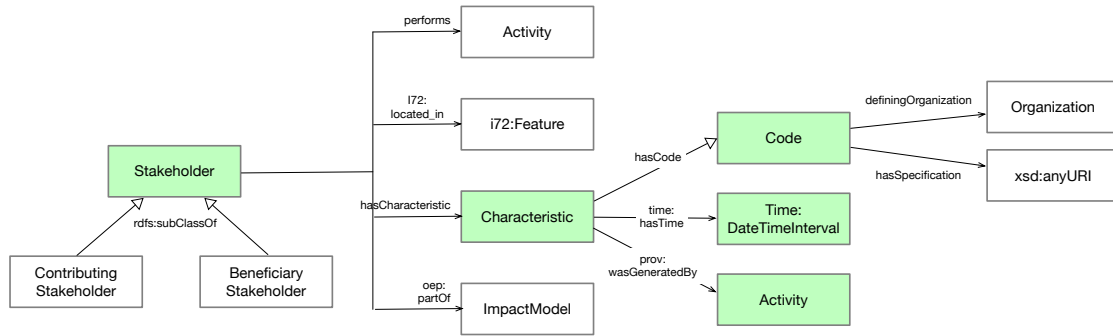
### 3. Client Pattern

The *Client* class represents a person who uses services and, potentially, benefits from their outcomes. The class inherits all properties of *5087-2:Person* and satisfies the needs related to *cids:Stakeholder* class, categorized by the stakeholder’s characteristics, as shown in Figure 2. A stakeholder, where *Stakeholder*  $\sqsubseteq$  (*Organization* or *Person*), can be related to a number of concepts, such as outcomes and locations, which characteristics cannot. At the same time, a client can be characterized along multiple dimensions, such as age and gender and a specific set of needs, such as addiction and homelessness. A specific service can be interested in a subset of those characteristics, such as a minimum age requirement for a homeless shelter. Hence, a client can be matched with services at the stakeholder rather than characteristic level. This allows for more flexibility in associating clients and services with stakeholders, and in turn, outcomes and location. Please refer to the CIDS manual for further details [17].

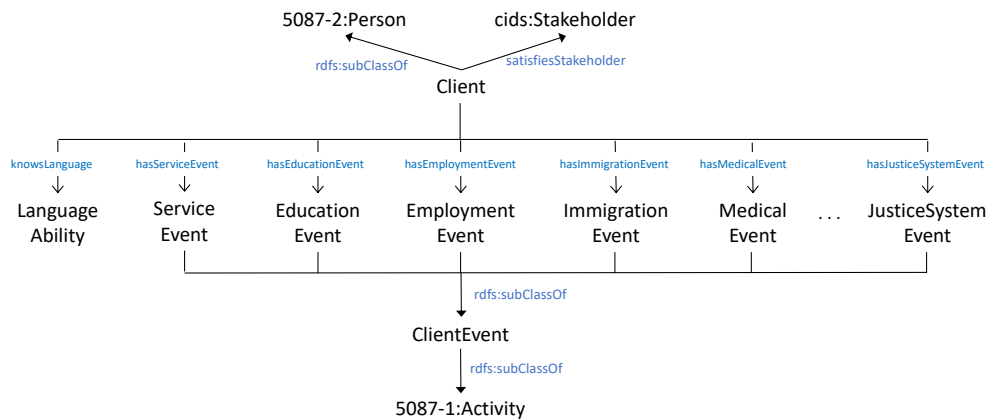
**DL 2 (Compass Client Class).**

$$\begin{aligned}
 Client &\sqsubseteq 5087-2:Person \\
 &\sqcap \forall \text{ satisfiesStakeholder.cids:Stakeholder} \quad \sqcap =1 \text{ hasGender.CL-Gender} \\
 &\sqcap \forall \text{ schema:knowsLanguage.LanguageAbility} \quad \sqcap \forall \text{ hasEthnicity.CL-Ethnicity} \\
 &\sqcap \forall \text{ hasDependent.5087-2:Person} \quad \sqcap \forall \text{ hasReligion.CL-Religion} \\
 &\sqcap \forall \text{ memberOfAboriginalGroup.CL-CA-AboriginalGroup}
 \end{aligned}
 \tag{DL-2}$$

The *Client* class extends the *5087-2:Person* class with the following properties:



**Figure 2:** Common Impact Data Standard *Stakeholder* class and its core classes.



**Figure 3:** Client class extension and related Event classes.

- **satisfiesStakeholder:** Identifies the stakeholder specifications that the client satisfies.
- **hasGender:** A coded property specifying the gender.
- **hasEthnicity:** A coded property specifying known ethnicities of the client.
- **memberOfAboriginalGroup:** A coded property identifying client’s aboriginal group.
- **hasReligion:** A coded property specifying known religions of the client.
- **hasDependent:** A set of Person instances, such as children or parents.
- **schema:knowsLanguage:** *LanguageAbility* instances, known and proficiency.

Some properties of the *Client* classes are “coded properties” where the range of values is constrained by client taxonomy codes, instances of *ClientCode*, where *ClientCode*  $\sqsubseteq$  *cids:Code*. Such codes are subclasses of the *ClientCode* class. For example, the class *CL-Age* can have instances of age categories, including *INST-Young* and *INST-Adult*. The subcategories for basic demographics include Age, Ethnicity, Family status, Gender, Religion, and Sexuality.

Central to the representation of information about a client are events that have occurred in their life. These events can span medical (in and out of institutions like hospitals), educational, housing, etc. Client events are used to derive characteristics of a client at some point in time. For example, one or more housing events may be used to determine that a client is homeless. Events can be viewed as records of a client’s life and viewed retrospectively to understand

their pathways. The *Client* class extends the definition in DL-2 with the following additional event-related properties, as shown in Figure 3:

- **hasServiceEvent**: instances of *ServiceEvent* for a service the client has received.
- **hasEducationEvent**: instances of *EducationEvent* for education the person has received.
- **hasEmploymentEvent**: instances of *EmploymentEvent*.
- **hasImmigrationEvent**: instances of *ImmigrationEvent*, each defining stages of immigration, if relevant.
- **hasMedicalEvent**: instances of *MedicalEvent* defining separate medical events, including births and deaths.
- **hasHousingEvent**: instances of *HousingEvent* that track changes in housing status including homelessness.
- **hasNameEvent**: instances of *NameEvent*.
- **hasGenderEvent**: instances of *GenderEvent*.
- **hasBirthEvent**: instances of *BirthEvent*.
- **hasDeathEvent**: instances of *DeathEvent*.
- **hasMaritalEvent**: instances of *MaritalEvent*.
- **hasHomelessEvent**: instances of *HomelessEvent*.
- **hasJusticeSystemEvent**: instances of *JusticeSystemEvent*.

### DL 3 (Compass Client Class).

*Client*  $\sqsubseteq$  5087-2:Person

$\sqcap \forall \text{hasServiceEvent. ServiceEvent}$	$\sqcap \forall \text{hasMedicalEvent. MedicalEvent}$
$\sqcap \forall \text{hasMaritalEvent. MaritalEvent}$	$\sqcap \forall \text{hasHousingEvent. HousingEvent}$
$\sqcap \forall \text{hasNameEvent. NameEvent}$	$\sqcap \forall \text{hasGenderEvent. GenderEvent}$
$\sqcap \forall \text{hasBirthEvent. BirthEvent}$	$\sqcap \forall \text{hasDeathEvent. DeathEvent}$
$\sqcap \forall \text{hasEducationEvent. EducationEvent}$	
$\sqcap \forall \text{hasEmploymentEvent. EmploymentEvent}$	
$\sqcap \forall \text{hasImmigrationEvent. ImmigrationEvent}$	
$\sqcap \forall \text{hasJusticeSystemEvent. JusticeSystemEvent}$	

(DL-3)

The definition for the *EducationEvent* class is provided below, with remaining client-related event definitions omitted. An *EducationEvent* is used to describe a client's educational or training status and experience. It is a subclass of *ClientEvent* and introduces the following properties:

- **hasStatus**: the status of the education event, complete, inProgress or incomplete.
- **atOrganization**: identifies the education organization the event took place.
- **hasCertification**: identifies the type of certification the education event has.
- **hasType**: identifies the type of education, including BA, BSC, etc.

#### DL 4 (Compass EducationEvent Class).

$EducationEvent \sqsubseteq ClientEvent$   
 $\sqcap =1 atOrganization.Organization$   
 $\sqcap \forall hasType.CL-EducationType \quad \sqcap \exists hasCertification.Certification \quad (DL-4)$   
 $\sqcap hasStatus.(\{complete\} \sqcup \{inProgress\} \sqcup \{incomplete\})$

**StakeholderEvent:** The *StakeholderEvent* captures events performed by stakeholders. The *cids:Stakeholder* class is a subclass of an *Organization* or *5087-2:Person* and identifies a person or organization required to answer competency questions such as: “How many users interacted with service?”, “How long did #6 stay in counselling?” and “How are people looking for help?”. The *StakeholderEvent* extends the *Event* class with the following property.

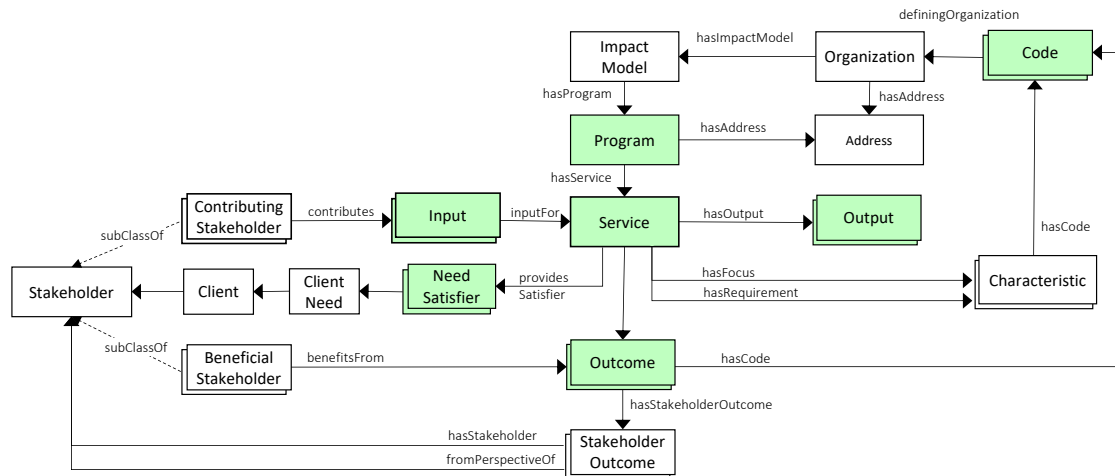
- **forStakeholder:** the stakeholder related to this Event instance.

#### DL 5 (Compass StakeholderEvent Class).

$StakeholderEvent \sqsubseteq Event \sqcap =1 forStakeholder.cids:Stakeholder \quad (DL-5)$

## 4. Service Patterns

The service is comprised of activities that focus on a shared set of inputs, outputs, and outcomes. It also defines a set of stakeholders that contribute to or benefit from the service. The Compass *Service* class defines a service provider and inherits all properties from the *cids:Service* class, which in turn is a subclass of *5087:Activity*. As such, a Service can be thought of as performing activities that cause some events to occur.



**Figure 4:** Compass extension of the *Service* class and its core classes.

A program is comprised of one or more services. For example, a “poverty reduction” program can have many services, with each service consisting of different activities, such as “find housing”, “assess housing readiness”, and “rent subsidy”, and complementary inputs, outputs and

outcomes. The *Compass Program* class extends the *cids:Program* class by redefining *hasService* to include the *Compass Service* class and adds an *ic:hasAddress* property. This extension allows an organization to provide services at different locations. For example, a “health” service can set up multiple offices closer to its clients.

- **hasService:** Identifies the Services that comprise this Program.
- **ic:hasAddress:** Identifies the address for this Program.

#### DL 6 (Compass Program Class).

$$\begin{aligned} Program &\sqsubseteq cids:Program && (DL-6) \\ &\sqcap \forall cids:hasServiceService \sqcap =1 hasAddress.ic:Address \end{aligned}$$

The *Compass Service* class is an extension of the *cids:Service* class. As described in [18], this pattern differentiates between a service (e.g. counselling services) and the resources it provides (e.g. counsellors, counseling sessions, counseling referrals), which is defined as *NeedSatisfier* instances. The *Service* extension introduces the following properties.

- **hasRequirement:** Identifies characteristics that limit who can use the service, listed in service taxonomy code list.
- **hasFocus:** Identifies client characteristics that the service focuses on, listed in service taxonomy code list.
- **hasMode:** The mode with which the service is delivered.
- **providesSatisfier:** The need satisfier this service provides.

#### DL 7 (Compass Service Class).

$$\begin{aligned} Service &\sqsubseteq cids:Service \\ &\sqcap \forall hasRequirement.cids:Characteristic \\ &\sqcap hasMode.(\{inperson\} \sqcup \{phone\} \sqcup \{online\} \sqcup \{offline\}) \\ &\sqcap \forall hasFocus.cids:Characteristic \sqcap \forall providesSatisfier.NeedSatisfier \end{aligned} \quad (DL-7)$$

Similar to the *Client* class, some properties of the *Service* classes are “coded properties”, meaning their range of values is constrained by service taxonomy codes, instances of *ServiceCode* where *ServiceCode*  $\sqsubseteq$  *cids:Code*.

**ServiceEvent:** A *ServiceEvent* is an event performed by a service that changes a client’s state in some way. It has the following properties:

- **hasStatus:** Status of the service.
- **atOrganization:** The organization providing the service.
- **forReferral:** The referral that led to the service event, if any.

#### DL 8 (Compass ServiceEvent Class).

$$\begin{aligned} ServiceEvent &\sqsubseteq ClientEvent \\ &\sqcap hasStatus.(\{scheduled\} \sqcup \{inProgress\} \sqcup \{completed\}) \\ &\sqcap =1 atOrganization.Organization \sqcap \forall forReferral.Referral \end{aligned} \quad (DL-8)$$

**ApplicationEvent:** When clients interact with service providers through a computer application, interaction-logging capabilities capture information related to service access. Relevant use cases include competency questions such as: “Who are the highest systems users currently?” and “What are the system interactions patterns for these clients over past 3 months?”. An *ApplicationEvent* is a subclass of *Event*. It defines an event created by an *Application* instance and captures any information relevant to that event.

- **hasApplication:** the application this event was created in
- **hasUserStakeholder:** the stakeholder using the application when the event was created.
- **dateCreated:** timestamp when the event was created
- **hasSource:** the URL or unique address where this event originated
- **hasMetaData:** the information stored with the event

**DL 9 (Compass ApplicationEvent Class).**

$$\begin{aligned}
 & \text{ApplicationEvent} \sqsubseteq \text{Event} \\
 & \sqcap =1 \text{ forStakeholder.Stakeholder} \quad \sqcap =1 \text{ hasApplication.Application} \\
 & \sqcap =1 \text{ hasUserStakeholder.Stakeholder} \quad \sqcap =1 \text{ sch:dateCreated.xsd:dateTime} \\
 & \sqcap =1 \text{ hasSource.xsd:string} \quad \sqcap \forall \text{ hasMetaData.xsd:string}
 \end{aligned}
 \tag{DL-9}$$

**ServiceFailureEvent:** The Compass *ServiceFailureEvent* class represents an event triggered when there is a barrier preventing clients from using a service they are otherwise eligible for. It also provides a solution, if one exists. This event occurs when there is a discrepancy between the service’s requirements and the client’s characteristics. For example, if a client is experiencing substance abuse (*INST-Addiction*), and a shelter (*cp:shelter-A*) requires its participants to be sober (*INST-Sober*), a *ServiceFailureEvent* is triggered. The triggered event’s properties will be: *forService* = *cp:shelter-A*, *hasCharacteristic* = *INST-Sober*, and *hasFailureType* equals any service or activity that, as its outcome, removes *INST-Sober* or adds *INST-Addiction* as the client’s characteristics. The *ServiceFailureEvent* class has the following properties:

- **forService:** *Service* or *Activity* this failure event indicates cannot be used by a client.
- **hasCharacteristic:** *Characteristic* causing the failure, i.e. service requirements not matching a client’s characteristics.
- **hasFailureType:** *Service* or *Activity* preventing the stakeholder from using the service.
- **hasDescription:** description of the failure type.

**DL 10 (Compass ServiceFailureEvent Class).**

$$\begin{aligned}
 & \text{ServiceFailureEvent} \sqsubseteq \text{Event} \\
 & \sqcap \forall \text{ forService.(Service or Activity)} \sqcap =1 \text{ hasCharacteristic.Characteristic} \\
 & \sqcap =1 \text{ hasFailureType.(Service or Activity)} \sqcap \forall \text{ hasDescription.xsd:string}
 \end{aligned}
 \tag{DL-10}$$

Clients can interact with service providers through a online software system, represented by the *Application* class, a subclass of *Service*, extending it with the following property:

- **hasSource:** The URL or unique address where the application can be referenced.



#### DL 11 (Compass Application Class).

$$Application \sqsubseteq Service \sqcap \forall hasSource.xsd:string \quad (DL-11)$$

#### 4.1. Client Community Pattern

A community is made up of two or more individuals that share some characteristics, such as age, gender, or set of goals, like finding housing or sobriety. These characteristics are represented as characteristics of the community. We also associate the characteristics of the clients with the location they reside in, to understand the service's catchment area. *5078-2:CityAdministrativeArea* class is imported from the 5078-2 ISO standard [9] to capture this functionality. Hence, a Compass *Community* class has the following property:

- **hasCommunityCharacteristic**: characteristics of the community this class represents

#### DL 12 (Compass Community Class).

$$Community \sqsubseteq 5087-2:CityAdministrativeArea \sqcap \forall hasCommunityCharacteristic.CommunityCharacteristic \quad (DL-12)$$

**CommunityCharacteristic**: The *CommunityCharacteristic* references the identifying characteristics that several individuals have in common that in turn makes them a community.

- **hasCommunityCharacteristic**: the characteristic that defines this *Community*
- **hasNumber**: the number of people in the community (not all individuals in a spatial area, only those that fit the *CommunityCharacteristic* profile).

#### DL 13 (Compass CommunityCharacteristic Class).

$$CommunityCharacteristic \sqsubseteq owl:Thing \sqcap \forall hasCharacteristic.cids:Characteristic \sqcap =1 hasNumber.xsd:NonNegativeInteger \quad (DL-13)$$

## 5. Evaluation

In previous work [1], we provided competency questions related to service coverage. The current paper focuses on client and service events evaluated with SPARQL queries.

**[ServiceEvent]** *How long did client #2 stay in counselling?* This question queries *cp:ServiceEvent* instances for entries related to client *cp:Client2* accessing counseling services, identified by the service taxonomy code *cp:INST-Counseling*. For each record found, it calculates the number of weeks between the data properties *time:hasBeginning* and *time:hasEnd*, indicating when the counseling services began and ended<sup>5</sup>. Weeks are summed in the **weeks** column of Table 1.

```
1 SELECT DISTINCT ?client ?weeks WHERE {
2   BIND(cp:Client2 AS ?client).
3   ?serviceEvent rdf:type cp:ServiceEvent ;
4   cp:forClient ?client ;
5   time:hasBeginning ?beg ; time:hasEnd ?end;
```

<sup>5</sup>Date arithmetic functions for *ofn:* and *spif:* defined at <https://graphdb.ontotext.com/documentation/9.10/free/sparql-functions-reference.html>.

**Table 1**  
Service CQ-2 SPARQL Query Results

client	weeks
cp:Client2	43

```

6  cids:hasCode cp:INST-Counseling.
7  BIND((ofn:weeksBetween(
8    spif:parseDate(?end, "yyyy-MM-dd'T'HH:mm:ss.SSS"),
9    spif:parseDate(?beg, "yyyy-MM-dd'T'HH:mm:ss.SSS")))
10 AS ?weeks). }

```

**[EducationEvent]** *What is our clients' education history and status?* This question queries *cp:EducationEvent* instances. It identifies two clients. *cp:Client400* who is currently attending a “trade school”, as indicated by **status=cp:InProgress**, after having completed “gradeschool” and “high school.” *cp:Client401* has also graduated “gradeschool” and “high school”, and is *scheduled* to start attending “higher education”. The results are listed in Table 2.

```

1  SELECT DISTINCT ?client ?eduEvent ?startF ?endF ?status
2  WHERE {
3    ?eduEvent rdf:type cp:EducationEvent ;
4      cp:forClient ?client ;
5      cp:hasStatus ?status ;
6      time:hasBeginning ?start.
7    BIND(spif:dateFormat(?start, "d-M-Y") as ?startF).
8    OPTIONAL{?eduEvent time:hasEnd ?end}.
9    BIND(spif:dateFormat(?end, "d-M-Y") as ?endF).
10 } ORDER BY ?client ?start

```

**Table 2**  
Education Event SPARQL Query Results

client	eduEvent	startF	endF	status
cp:Client400	cp:graduate-gradeschool-client400	1-9-2003	1-5-2007	cp:completed
cp:Client400	cp:graduate-highschool-client400	1-9-2006	1-5-2010	cp:completed
cp:Client400	cp:graduate-trade_school-client400	1-9-2010		cp:InProgress
cp:Client401	cp:graduate-gradeschool-client401	1-9-2003	1-5-2007	cp:completed
cp:Client401	cp:graduate-highschool-client401	1-9-2007	1-5-2011	cp:completed
cp:Client401	cp:graduate-higher_education-client400	1-9-2011		cp:scheduled

**[ApplicationEvent]** *What stakeholders access our application most frequently?* This question queries *cp:ApplicationEvent* instances, identifying two types of Application accessed (**app**) and three types of stakeholders (**sh**), as listed in Table 3. **appEvents** sums the number of instances by **app** and **sh**. The top event *cp:App\_CounselingSessions* is used by the *cp:FemaleHousedYouth-in\_Area0* stakeholders, totalling 12 instances. This stakeholder’s characteristics are linked to Code values identifying female youths that live in the location “Area0.”

```

1  SELECT DISTINCT ?app ?sh (count(?appEvent) as ?appEvents)
2  WHERE {
3    ?appEvent rdf:type cp:ApplicationEvent ;
4      cp:hasApplication ?app;
5      cp:hasUserStakeholder ?sh;
6      cp:occursAt ?occurs.
7    BIND(spif:dateFormat(?occurs, "d-M-Y hh:mm") as ?occursF).
8  } GROUP BY ?app ?sh ORDER BY ?app DESC(?appEvents)

```

**Table 3**

Application Event SPARQL Query Results

app	sh	appEvents
cp:App_CounselingSessions	cp:sh-Female-Housed-Youth-in_Area0	12
cp:App_CounselingSessions	cp:sh-Adult-Female-Homeless-in_Area0	4
cp:App_ServiceSearch	cp:sh-Female-Homeless-Youth-in_Area0	9
cp:App_ServiceSearch	cp:sh-Adult-Female-Homeless-in_Area0	5
cp:App_ServiceSearch	cp:sh-Homeless-Male-Youth-in-in_Area0	3

## 6. Discussion and Conclusion

This paper introduces the Compass ontology and evaluates its ability to answer a set of competency questions posed by various stakeholders in social services. The answers to the selected queries were extracted from a knowledge graph based on the Compass ontology and populated with client, service, service usage, and events data.

Our evaluation demonstrates how competency questions can be answered using the Compass ontology. **Service queries** identify service events aggregated to answer temporal questions related to the length of client usage of a service category, namely counselling services. By associating services with a specific code, like *INST-Counseling*, the query can aggregate over all service instances a client used. We can also track a client's progress and infer their latest state by identifying the last "complete" client event, such as *EducationEvent* to infer their education status. Finally, we can evaluate software system usage by analyzing application logs associated with the characteristics of system users. We can then answer questions about the most frequent usage by client categories defined by the *ClientCode* instances, such as age, gender, housing status, and location. The event, client, service definitions and code taxonomies continue to be extended to ensure ontological concepts defined by Compass are shared and compared between organizations working towards similar or overlapping outcomes.

## References

- [1] M. S. Fox, B. Gajderowicz, D. Rosu, A. Turner, L. Lyu, An Ontological Approach to Analysing Social Service Provisioning, in: IEEE International Smart Cities Conference (ISC2), 2022, p. In press.
- [2] M. S. Fox, K. Ruff, CIDS: An Ontology for Representing Social and Environmental Impact, in: Ontologies for the Behavioural and Social Sciences Workshop (OntoBESS), at 12th International Conference on Biomedical Ontologies, volume 3073, CEUR Workshop Proceedings, Bolzano, Italy, 2021, pp. 162--173.
- [3] K. Ruff, S. Olsen, The Next Frontier in Social Impact Measurement Isn't Measurement at All, Stanford Social Innovation Review (2016) 1--8. URL: [https://ssir.org/articles/entry/next\\_{ }frontier\\_{ }in\\_{ }social\\_{ }impact\\_{ }measurement\\_{ }#](https://ssir.org/articles/entry/next_{ }frontier_{ }in_{ }social_{ }impact_{ }measurement_{ }#).
- [4] M. D. Lytras, A. Visvizi, Who uses smart city services and what to make of it: Toward interdisciplinary smart cities research, Sustainability (Switzerland) 10 (2018) 1--16. doi:10.3390/su10061998.

- [5] B. Gajderowicz, M. S. Fox, M. Grüninger, Ontology of social service needs: Perspective of a cognitive agent, in: Proceedings of the 2018 Joint Ontology Workshops, Cognition And OntologieS + Explainable AI, Cape Town, 2018, pp. 1--12.
- [6] L. Biccheri, R. Ferrario, An analysis of the notion of need for the representation of public services, CEUR Workshop Proceedings 2518 (2019).
- [7] B. Gajderowicz, M. S. Fox, M. Grüninger, Requirements for an ontological foundation for modelling social service chains, in: Y. Guan, J. Liao (Eds.), Proceedings of the 2014 Industrial and Systems Engineering Research Conference, Montréal, QC, 2014, p. 4104.
- [8] Aunt Bertha Inc, Open Eligibility Project, 2017. URL: <http://openeligibility.org>.
- [9] International Organization for Standardization, ISO/IEC CD 5087-2 Information technology - City data model - Part 2: City level concepts, 2022. URL: <https://www.iso.org/standard/80742.html>.
- [10] M. S. Fox, The TOVE Project Towards a Common-Sense Model of the Enterprise, in: Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Springer Berlin Heidelberg, 1992, pp. 25--34.
- [11] M. S. Fox, M. Barbuceanu, M. Grüninger, J. Lin, An Organization Ontology for Enterprise Modelling, in: Modeling, International Conference on Enterprise Integration Modelling Technology, Springer, 1997, pp. 1--25.
- [12] R. A. Falbo, G. K. Quirino, J. C. Nardi, M. P. Barcellos, G. Guizzardi, N. Guarino, A. Longo, B. Livieri, An Ontology Pattern Language for service modeling, Proceedings of the ACM Symposium on Applied Computing 04-08-April (2016) 321--326. doi:10.1145/2851613.2851840.
- [13] R. Schmidt, Ontology-based Modelling of Service Processes and Services, Proceedings of the IADIS International Conference (2007) 67--74. URL: <http://www.iadis.net/dl/final{ }uploads/200702L009.pdf>.
- [14] E. M. Sanfilippo, S. Borgo, C. Masolo, Events and activities: Is there an Ontology behind BPMN?, Frontiers in Artificial Intelligence and Applications 267 (2014) 147--156. doi:10.3233/978-1-61499-438-1-147.
- [15] A. Sathi, M. S. Fox, M. Greenberg, Representation of activity knowledge for project management, IEEE Transactions on pattern analysis and machine intelligence (1985) 531--552.
- [16] M. Katsumi, M. S. Fox, Defining Activity Specifications in OWL, in: Workshop on Ontology Patterns, Vienna, Austria, 2017, pp. 1--12.
- [17] M. S. Fox, K. Ruff, A. Chowdhury, B. Gajderowicz, T. Abdulai, J. Zhang, The Common Impact Data Standard: An Ontology for Representing Impact, Technical Report, University of Toronto, Toronto, Canada, 2021. URL: <https://commonapproach.org/wp-content/uploads/2020/12/Common-Impact-Data-Standard-V1.1.pdf>.
- [18] D. Rosu, M. S. Fox, B. Gajderowicz, Compass Needs Ontology : A Design Pattern for Representing Needs in Social Work, in: The Eighth Joint Ontology Workshops (JOWO'22), CEUR Workshop Proceedings, CEUR-WS.org, Jönköping, Sweden, 2022, p. 12.
- [19] International, ISO/IEC DIS 5087-1 Information technology – City data model – Part 1: Foundation level concepts, 2022. URL: <https://www.iso.org/standard/80718.html>.