COBACORE

Community Based Comprehensive Recovery

Grant Agreement № 313308

D2.3: Development of the COBACORE Data Framework

Version:	2.0
Due Date:	31 March 2016
Delivery Date:	30 September 2016 (v2)
Nature:	Report
Dissemination Level:	Public
Lead partner:	UU
Authors:	Martin Haran, Leo Galway, Conor Woods, James Connolly and Ivan Palomares Carrascosa (UU)
Internal reviewers:	David McMullan (Spatialest), Jose Manuel Sanchez (IntegraSys)

www.COBACORE.eu

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 313308.

Version Control:

Version	Date	Author	Author's Organization	Changes
0.1	30-06-2015	Martin Haran, Leo Galway	UU	Base draft
0.2	22-07-2015	Martin Haran, Leo Galway	UU	Review draft
0.3	18-01-2016	Martin Haran, Leo Galway	UU	Final version as submitted.
2.0	30-09-2016	Martin Haran, Leo Galway, Conor Woods	UU	Revised version.

Deliverable Title: Report on the Development of the COBACORE Data Framework

Deliverable Number 2.3

Keywords: Framework Design and Architectural Composition, Functional Specification; Feature Compliance; Framework-Platform Integration.

Executive Summary

This revised version of D2.3 provides a comprehensive overview of the design process and development stages undertaken in producing the operational COBACORE data framework. The deliverable has been updated to afford a much more detailed technical account of the science underpinning the data framework. D2.3 serves to contextualise and communicate the 'under the bonnet' operational capacities which underpinned the function and feature sets of the COBACORE platform. This report serves as a narrative to the operational COBACORE data framework and allows an appreciation of the iterative learning curve that was the development process for a data framework within the disaster recovery and reconstruction domain. D2.3 brings transparency and affords a greater appreciation of the underpinning architecture and 'back-office' operations which serve as facilitators to the functional capacities of the COBACORE platform.

This updated version of the deliverable affords enhanced understanding of the 'bespoke' nature and attributes of the COBACORE data framework, the learning pathway, the refinement measures and streamlining implemented following the integration process (between the data framework and the platform developed in WP4) and the robustness of the 'load' testing undertaken to ensure operational capacity as well as the efficiency and 'reliability' of system generated 'outputs'.

Table of Contents

List of Figures

List of Tables

1 Introduction

Building upon the data identification (D2.1) and modelling hierarchy (D2.2), WP2 has worked in partnership with consortium partners in Work Packages 3, 4 and 5 to create a data framework architecture. This encompasses a series of design considerations necessary to facilitate the agreed functional specification of the COBACORE platform and to ensure effective alignment with end-user expectations (professional responders, affected community and volunteer responders). Moreover, the operational capacity and functional specifications of the framework have been developed to support the management, visualisation, mining and model development undertaken in WP3 and WP4 and in conformance with the needs and expectations of the evaluation exercises designed and developed in WP5.

The design and development of the data framework (Task 2.2) and its subsequent integration with the platform developed in WP4 constitutes the principal technical activities undertaken in WP2. This revised version of D2.3 serves as an narrative to the framework development and processes by depicting the critical thinking applied in the refinement and customisation of the framework architecture as well as communicating the rationale and premise underpinning the final framework design and the process of integration between the data framework (WP2) and the platform (WP4).

D2.3 is a public (PU) deliverable, as such; concerted effort had been made in the original submission to ensure that the development process was conveyed utilising terminology and language which was comprehensible to prospective end-users (given the multi-disciplinary nature of the domain). In retrospect it was felt that this perhaps served to 'over-simplify' the development process and to 'under-represent' the science and mathematical composition underpinning the data framework. As such, this updated report includes a detailed and comprehensive overview of the iterative nature of the development process applied, encompassing the exploration of different design pathways, the alternative 'matching' techniques and models applied as well as the in-house and stakeholder testing undertaken over the course of the WP2 development timeline. Specifically, the report affords a more detailed appreciation of the semantic capacities and sense-making functionalities which may not be immediately apparent to prospective end-users of the platform (in essence provide insight to the '*under the bonnet'* operations developed and implemented within the confines of the data framework which serve to underpin the platform functions and features).

1.1 Structure of the document

The remainder of the report is presented in a series of sections. Section 2 details the positioning of the data framework within the COBACORE project, including how the development process served to advance technical capacity and innovation within the recovery and reconstruction domain. Section 3 details the constituent components of the data framework architecture and the relationship with the main platform. The semantic architecture underpinning the data framework and the sense making features including the key attributes encompassed within the matching algorithm are detailed in section 4. Key

learning outcomes including revisions to the data framework design pre and post testing are detailed in section 5. Conclusions are presented in section 6.

The Annex contains three publications that are connected to this report. The reader is invited to read these publication to obtain a comprehensive view of the work done under the umbrella of task 2.2.

1.2 Changes in the revised version of this report

This updated version of the deliverable affords enhanced understanding of the 'bespoke' nature and attributes of the COBACORE data framework, the learning pathway, the refinement measures and streamlining implemented following the integration process (between the data framework and the platform developed in WP4) and the robustness of the 'load' testing undertaken to ensure operational capacity as well as the efficiency and 'reliability' of system generated 'outputs'.

Specific additions to the updated version of the deliverable include a detailed technical overview of the semantic architecture, algorithm development and sense making technologies employed and tested as part of development process (including provision of the underlying scientific and mathematical dimensions of the framework) and how this has served to advance the current state of the art within the disaster recovery and reconstruction domain.

2 Positioning the Data Framework within COBACORE

Within the domain of crisis response and recovery, there is a need for improved situational awareness in order to fully harness the capabilities and capacities within the affected community. Furthermore, to achieve such improvements, platforms that facilitate enhanced sense making, with a view to improving communication and organization of the affected community, will potentially benefit the overall response and recovery efforts. The Community Based Comprehensive Recovery (COBACORE) project aims to support common needs assessment and recovery planning efforts in complex multi-sectorial, multi-stakeholder crisis environments, by building upon the community as an important source of information and capabilities. Moreover, COBACORE aims to help bridge the so-called collaboration gap: failure of collaboration through insufficient information sharing among partners, incompatible work practices and misaligned decision making processes, in order to improve situational awareness and to serve as a robust and credible evidence base to inform decision making and guide resource deployment and allocation.

The COBACORE platform will utilise a wide range of data from heterogeneous sources (D2.1 provides a detailed overview of the forms and sources of such data) for the purposes of representation and analysis of situation, community and needs models during disaster response and recovery. Through analysis of the data encapsulated within these models, improved situational awareness may be achieved, which leads to the provision of sense-making capabilities as an aspect of the functionality inherent within the platform. Subsequently, a **‘suitable data framework’** and **‘apposite data flow process’** is vital to achieving a useful and timely response to the interactions of disparate data sources.

From the perspective of the underlying data flow, the COBACORE platform comprises two primary data processing components. The first key component is a web-, or mobile-, based platform, which is responsible for dynamic data acquisition, data storage and initiating queries on the situation, community and needs models on behalf of a user. In addition, this component provides interfaces to the various users of the platform and performs visualization of various facets of the underlying baseline data and data acquired by the platform, from both the affected and responding communities, along with visualization of the responses to information requests by users. In essence, data *flows* into the platform and information *flows* out of the platform via this component. Although the component provides a degree of data processing and rudimentary sense making, subsequent *intelligent* sense making will utilize a secondary semantic framework component, which primarily provides a mapping between the disparate data sources onto the situation, community and needs models, through the use of an ontology that encapsulates the core domain concepts utilized by COBACORE.

Consequently, the ontology facilitates interoperability between the data sources by providing a holistic representation of their inherent knowledge, and may be indirectly utilized by users to infer information beyond the scope of an individual’s input data. this endows the overall platform with the capacity to provide information-informed decision making. Accordingly, data input into the system by individual users will be classified according to the information model

represented by the ontology. Subsequently, associations between a user's data and any existing data that has been captured by the COBACORE platform may be determined and visualized by the web-, or mobile- based platform. This helps users to gain a more useful understanding of their data by framing the data within the wider context of the current situation.

2.1 Contribution to advancing the current SOTA

In terms of innovation and advancement of the current state of the art (SOTA), the COBACORE data framework differs from existing disaster decision support systems in that the focus is exclusively on the 'recovery and reconstruction' phase of the disaster cycle. The Unique Selling Point (USP) from an end-user perspective is the fact that the COBACORE data framework has been designed to be inter-operable (with existing Decision Support Systems (DSS). Added to this, the data framework permits disaster recovery and reconstruction professionals to 'plug-in' or upload their own data, which can then be overlaid with the dynamic data being generated during the recovery and reconstruction event. The ability to utilise both existing and dynamic datasets was a feature repeatedly highlighted by recovery and reconstruction professionals participating in the evaluation events as one of the most desirable features of the COBACORE data framework. A further dimension through which the COBACORE data framework architecture advances the current SOTA is 'transferability' and 'scalability' of the design as stipulated in D2.2. These features were originally built in to accommodate different forms, types and scale of disaster, but such is the flexibility of the design that the underpinning DSS architecture has now been deployed for urban regeneration and is also being adapted/modified to be deployed as an events planning DSS – facilitating co-ordination across organisers, police and associated emergency service personnel.

Whilst the data framework development process may not have served to advance semantic 'science' – the application of the semantic architecture and associated sense-making technologies to the disaster recovery and reconstruction domain is innovative in itself. Although a number of crisis-orientated ontologies exist, not all of these are explicitly focused on the response and recovery stage of the disaster management cycle. Pertinently, COBACORE baseline concepts such as Need and Capacity had, up until this project, not been encompassed within existing crisis-orientated ontologies. Building upon the work of Liu et al (2013)[1] the COBACORE data framework served to expand the ontology capacity within the disaster recovery and reconstruction domain, developing a series of re-usable ontological concepts. These are publically available outputs from the project and will serve as a platform for future development activity within this domain (*see* Palomares et al, 2015: Galway et al, 2015 and Palomares et al, 2016).

Moreover, the semantic architecture developed within the COBACORE project has served to advance this dimension of computer science within the confines of disaster recovery and reconstruction. The COBACORE project represents the first research project to apply semantic

[1] Liu et al (2013) identified 26 existing ontologies specific to the disaster recovery and reconstruction domain.

reasoning to disaster recovery and reconstruction (previous research has focussed on disaster response). The sematic architecture developed for the project is detailed extensively in section 4. A series of in-house tests validated the ‘sense-making’ capacity of the architecture developed and but for the operational inefficiencies encountered upon integration (data framework with the platform), this would have constituted a major dynamic of the technical output. The fact that the semantic architecture was removed from the final framework design was predicated on the need to prioritise operational efficiency (load carrying capacity) during the final evaluation. The semantic architecture developed for COBACORE is nonetheless deployable, particularly in a more longitudinal evaluation (akin to a real-life disaster recovery and reconstruction timeline). Further development work could be undertaken in order to improve the ‘efficiency’ and ‘expedience’ of the semantic reasoning. Within these confines the base architecture affords a robust and credible platform upon which future refinement could be initiated.

Since the COBACORE project was initiated, the role and contribution of ‘unbound volunteers’ in disaster recovery has grown in prominence. There are numerous examples across the world of spontaneous social media initiated volunteer groups being formed in expectation of contributing to the recovery and reconstruction effort. These ‘spontaneous’ groups (although well-intentioned) have exhibited varying degrees of success in terms of the outputs delivered and the extent to which they have aligned with the overall recovery plan. It is noteworthy that the COBACORE project timeline has coincided with the initiation and launch of Recovers.org – a subscription based web and mobile platform for coordinating disaster relief efforts. Initially deployed in the US, Recovers.org has commenced a process of global expansion with communities paying a subscription to receive a personalised web site, training, and access to technology that lets them match volunteers and donations to people and places in need. The Recovers.org concept is very much akin to that proffered by COBACORE – indeed it utilises the concept of ‘need’ as developed within the COBACORE ontology framework. The Recovers.org platform presently has a narrower scope and does not have the levels of interoperability (with other DSS’) afforded by the COBACORE data framework. Recovers.org does nonetheless serve to underpin the demand for improved data sharing, coordination capacity and integration between volunteer responders and professionals. To date more than 180 communities have ‘subscribed’ to Recovers.org – mostly in the US.

With concerted efforts being presently advanced on data standardisation by the ISO committee 299 Security and resilience including the production of ‘Guidelines for planning of involvement of spontaneous volunteers’[2] the operational capacity and inter-operability of DSS within the disaster recovery and reconstruction domain will advance considerably over the next 2-3 years. In preparation for that evolution the COBACORE data framework in its current configuration constitutes a free-to-use, publically available, deployable resource for communities across Europe facilitating collaboration and data sharing between professionals

[2] https://www.iso.org/obp/ui/#iso:std:66951:en

and communities. Customisation of the data framework can be facilitated via the COBACORE team - if necessitated.

3 COBACORE Platform and Data Model

The overarching architecture of the COBACORE platform, as designed and developed through an iterative process involving both feature design (WP3) and prototype evaluations (WP5). It is built upon the Microsoft .NET technology stack. Consequently, the core platform and Relational Database Management System (RDBMS) are hosted using .NET Server, within the Microsoft Azure cloud platform. Central to the platform is the use of a Microsoft SQL Server database, which provides a relational database for the back-end storage of both the static base information required for disaster recovery an reconstruction within a given scenario (typically collated prior to platform deployment) and the dynamic information added to the platform (typically acquired over time through usage by the tripartite stakeholders: Affected Community, Responding Community, Professional Responders). Correspondingly, the back-end database forms the COBACORE Data Model.

3.1 The COBACORE Data Model

Currently, the relational database that constitutes the COBACORE Data Model is represented using a traditional flat table format, which contains over 50 tables that utilise a large number of interlinked dependencies, as illustrated in Figure 1.

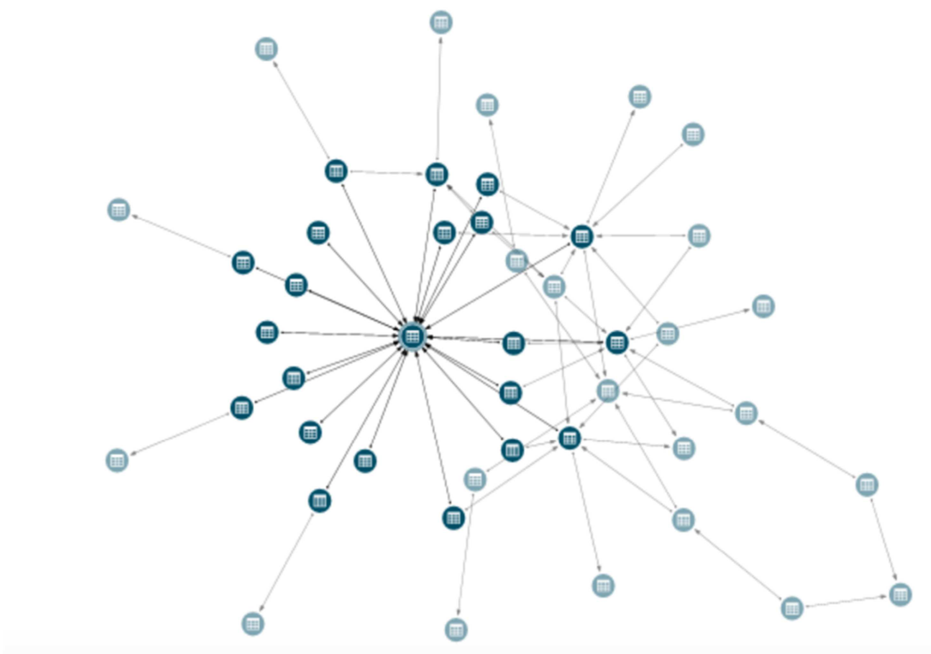


Figure 1: Interlinked Dependencies of the COBACORE Data Model

As may be observed in Figure 1, a number of tables are employed to provide interlinking between other (key) tables within the relational database, which subsequently provide an explicit representation of the core concepts defined by WP3, whereby an instance of a concept

is stored as a single row in the corresponding table. In addition, a number of other tables are solely utilised to maintain the features and visualisations of the platform. Table 1 provides an example of the primary tables within the relational database that are used to store instances reflecting the core concepts depicted in Deliverable2.2 namely; Need (*needs*), Capacity (*resources*), Activity (*activities*) and Actor (*actors*).

Table Name	Purpose	Dependencies (Tables)
activities	Storage of data relating to activities registered on the platform	activities_groups activities_linkedneeds activities_linkedresources activities_notes activity_status activities_watch actors categories conversations
actors	Storage of data relating to actors registered on the platform	activities activities_groups activities_watch actor_addresses actor_contactdetails actor_gcmdevices actor_pins actor_ratings groups groups_actors messages needs needs_affectedactors needs_watch notifications org_board_items persons

		resources resources_availableactors resources_watch socialmessages users
needs	Storage of data representing a Need registered on the platform	activities_linkedneeds actors conversations needs_affectedactors needs_watch needs_categories status urgencies
capacity	Storage of data representing a Capacity registered on the platform	activities_linkedresources actors available_times conversations resources_availableactors resources_watch resources_categories status

Table 1: Key Tables Utilised by the Data Model

For each of the primary database tables listed in Table 1, the corresponding set of dependencies is also given, indicating the set of database tables that are interlinked with each of the primary tables. Consequently, as Table 1 shows, there are a large number of linked dependencies pertaining to tables that are used to represent the core concepts of the platform, along with tables that retain supporting information in terms of both platform user data and platform operation. While the use of the relational data model facilitates primary information analysis within the platform, advanced sense making requires a degree of functionality based on semantic analysis of the relationships inherent in the data model. To this end, an information model has been defined and developed in conjunction with a supporting semantic framework.

3.2 The COBACORE Information Model

Similar to the data model, the creation of the information model and corresponding ontology has been premised on the set of conceptual data models and subsequent set of core concepts previously identified and defined within deliverable D2.2. Through an iterative process, a refined version of the information model has been developed, in parallel to the design and implementation of the COBACORE platform. Accordingly, the primary user of the model was stipulated as the COBACORE platform, which will have direct access to the information model. This therefore provides indirect access, through utilization of the platform to users from both the affected and responding community, along with platform users that belong to the group of professional responders. In terms of the corresponding use case, the information model will provide the platform with advanced sense making functionality, by supporting advanced matching during information analysis conducted by the platform.

In order to develop the information model a knowledge engineering approach was utilized. This was based upon commonly used concepts related to the disaster recovery and reconstruction domain. A number of corresponding ontologies and vocabularies were initially identified and investigated for possible use as a basis for the COBACORE information model. However, as discussed extensively in D2.2 the bespoke nature of the COBACORE platform necessitated development of a unique, extensible ontology that was initially premised on the four core concepts. Consequently, the COBACORE ontology is primarily focused on the core concepts, thereby providing a foundation for the future expansion of the ontology for the disaster recovery domain. Protégé has been used to generate the ontology, utilizing the OWL2 DL profile and Pellet reasoner, in order to test for consistency and inference during development of the information model.

Figure 2 provides an illustration of the current revision of the ontology, featuring a set of core classes and subsidiary classes, which are partitioned into domain concepts, and values that are utilized for the representation of enumerated classes. The base class for the information model is defined as the *DomainConcept* class, which comprises the primitive classes (*Need*, *Capacity*, *Activity*, *Actor*) that represent the four core concepts, along with a number of supporting classes, including *Category*, *Type* and *Location*, which are utilized by relationships within the ontology. For example, the *Category* class contains a number of subclasses that represent the recovery domain associated with a *Need*, *Capacity* or *Activity*, e.g. *Amenities*, *Environment*, *Healthcare*, etc. Likewise, the *Type* class contains three subclasses, each with a set of additional subclasses, e.g. *Assets (Physical, Financial)*, *Information (Instruction, Expertise)*, and *Skills (Service, Labour)*. These are utilized within the ontology to create a set of subclasses based on the *Category* classes, which may be used to implicitly categorize the type of *Need* or *Capacity* to which an instance belongs, as part of the pre-filtering phase of the matching algorithm (discussed later in Section 4).

In conjunction with the *DomainConcept* base class of the information model, a number of classes have also been defined that represent enumerations that are utilized in relationships by some of the *DomainConcept* subclasses. Subsequently, as illustrated in Figure 2, the *ValuePartition* class acts as the root class for the enumerated classes, which provide a set of

defined classes, including *GroupRole*, *Urgency* and *Status*, that contain disjointed subclasses with corresponding instances. This is for the purpose of representing a fixed set of predefined values within the information model. For example, the *GroupRole* class contains two subclasses with the corresponding literal values *participant* and *lead contact*, which are used to specify an aspect of the relationship between an *Actor* and a *Group*. Similarly, the *Urgency* class contains a set of mutually exclusive subclasses, with corresponding literal values, that permit the relationship between the *Need* class and a set of predefined timeframes to be represented within the information model. By building up sets of primitive and defined classes around the core concepts, the resulting ontology facilitates inference-based analysis of instances contained within the underlying data model. This provides the COBACORE platform with a contextualized view of the requirements and capabilities of individuals and groups within the affected and responding communities.

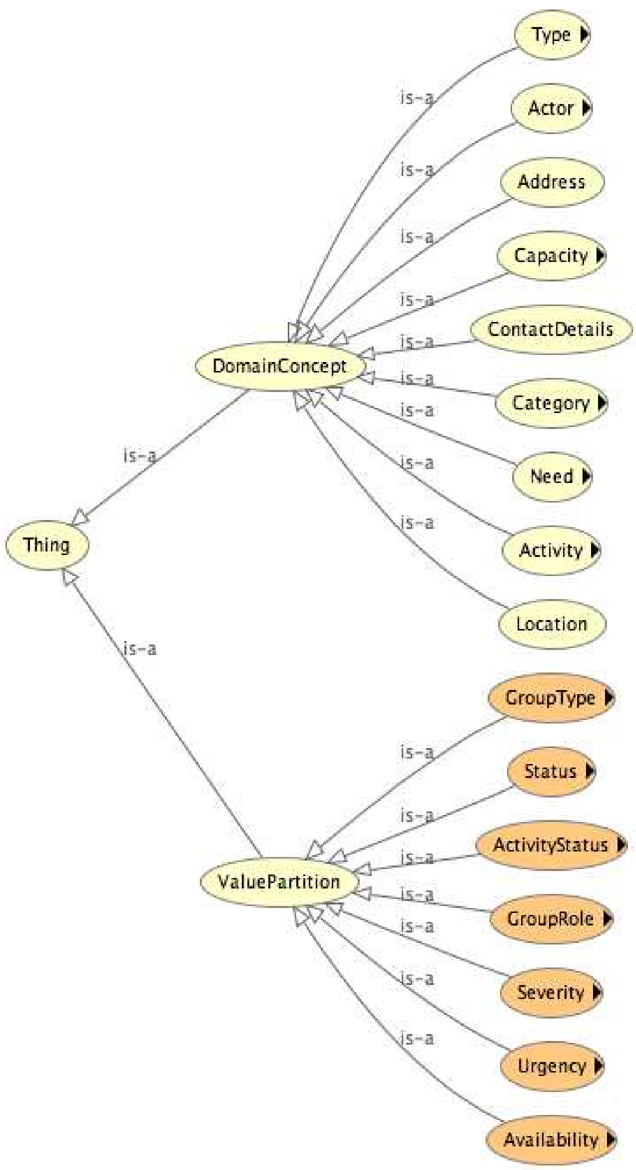


Figure 2: High-Level View of the COBACORE Information Model

3.2.1. The Information Model *Need* Class

In the COBACORE platform the definition of a *Need* represents both essential and non-essential requirements by an *Actor* (or *Group*) that are predominantly members of the affected or responding community. Although considered as a core concept for the COBACORE platform, the concept of a *Need* is not a commonly described concept within existing crisis-orientated ontologies. Consequently, one aspect of the added value of the COBACORE platform is in the facility to capture and analyse the unmet needs of the affected and responding community post disaster response. In terms of the information model, the Resource Description Framework Schema (RDFS) for the concept of a *Need* is given in Figure 3.

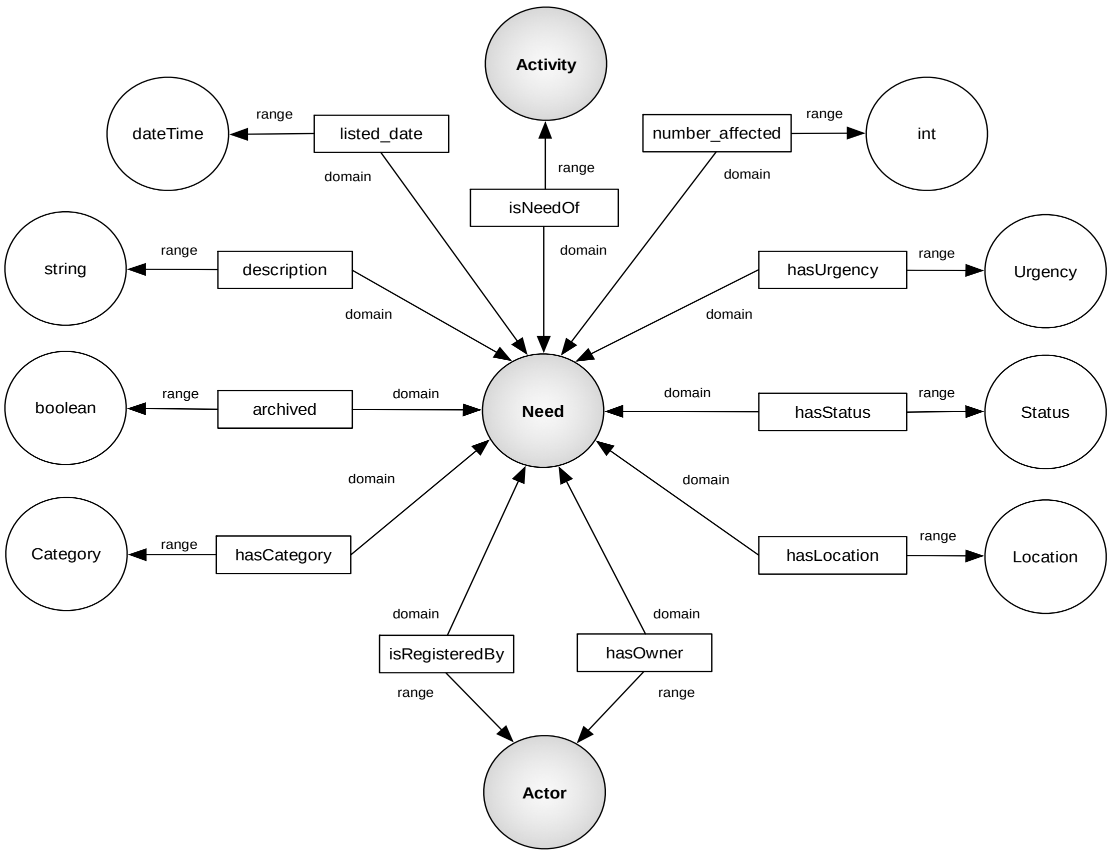


Figure 3: RDF Schema Diagram for *NEED* Concept

As can be seen in Figure 3, the *Need* class is intrinsically linked to the core *Actor* and *Activity* classes through a number of relationships that characterize the intrinsic associations between the classes. An instance of a *Need* is both registered by and owned by an instance of an *Actor*, and may be associated with an instance of an *Activity*. The *Need* class also contains a number of relationships to other classes within the ontology, including *Category*, *Urgency*, *Status* and *Location* and contains a number of explicit attributes, such as *archived*, *description*, *listed date*, and *number affected*. These provide further information utilized by the COBACORE platform for both data analysis and visualization. Through the use of the *Need* class within the information model and corresponding data model, the COBACORE platform can maintain a

'real-time' overview of the actual needs, thereby providing an outlook on how such needs will be met.

3.2.2. The Information Model *Capacity* Class

In contrast to a *Need*, within the COBACORE platform a *Capacity* represents a potential resource that could be used to influence the situation and help fulfil a *Need* or participate in an '*Activity*'. This not only includes a physical resource associated with an *Actor* but also resources in terms of skills, information and financial capabilities. The corresponding Resource Description Framework Schema for the concept of *Capacity* is given in Figure 4.

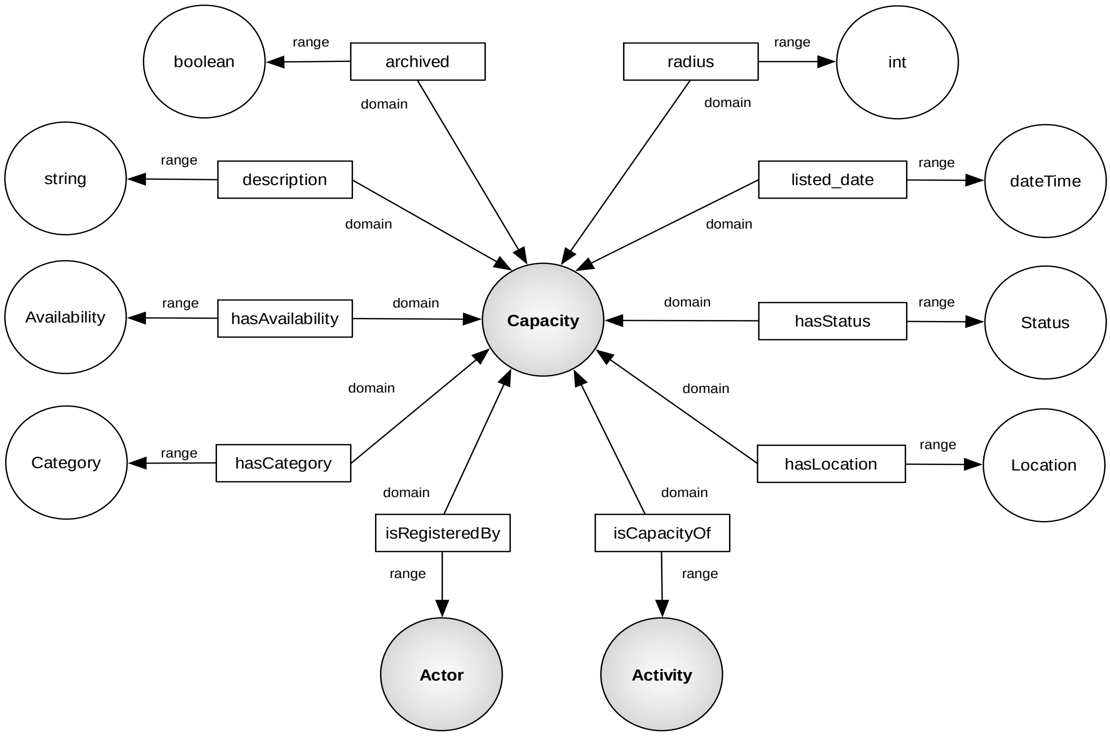


Figure 4: RDF Schema Diagram for the *Capacity* Concept

Similar to a *Need*, the *Capacity* class, as shown in Figure 4, is intrinsically linked to the core *Actor* and *Activity* classes through relationships, whereby an instance of a *Capacity* is registered on the COBACORE platform by an *Actor* (or *Group*) and may be utilized by an instance of an *Activity*. In addition, the *Capacity* class contains relationships to a number of subsidiary classes within the ontology, such as *Availability*, *Category*, *Status* and *Location*, along with an explicit set of attributes, including *archived*, *description*, *listed date*, and *radius*, which permit the duration and sphere of influence of an instance of a *Capacity* to be represented within the information model. Subsequently, through the set of relationships and attributes, capacities registered on the COBACORE platform may be analysed for use within the situational overview provided by the platform. Moreover, both relationships and attributes may be potentially utilized in order to conduct advanced matching of registered needs and capacities.

3.2.3. The Information Model *Activity* Class

Within the COBACORE platform an *Activity* represents an action, or set of actions that may be carried out, which will have an effect on the overall situation. The overall concept of *Activity* aims to link an action with a stated *Need*, which permits interaction between actors (or groups) and the outcome of a match between one or more *Need* and one or more *Capacity* to be monitored. Correspondingly, an *Activity* can describe any type of action within the domain that may be independent of a registered *Need* but which belongs to an *Actor* (or *Group*). Moreover, the concept of *Activity* is fundamental to the COBACORE platform as it can potentially provide a direct relationship to all three of the other core concepts, as illustrated in Figure 5.

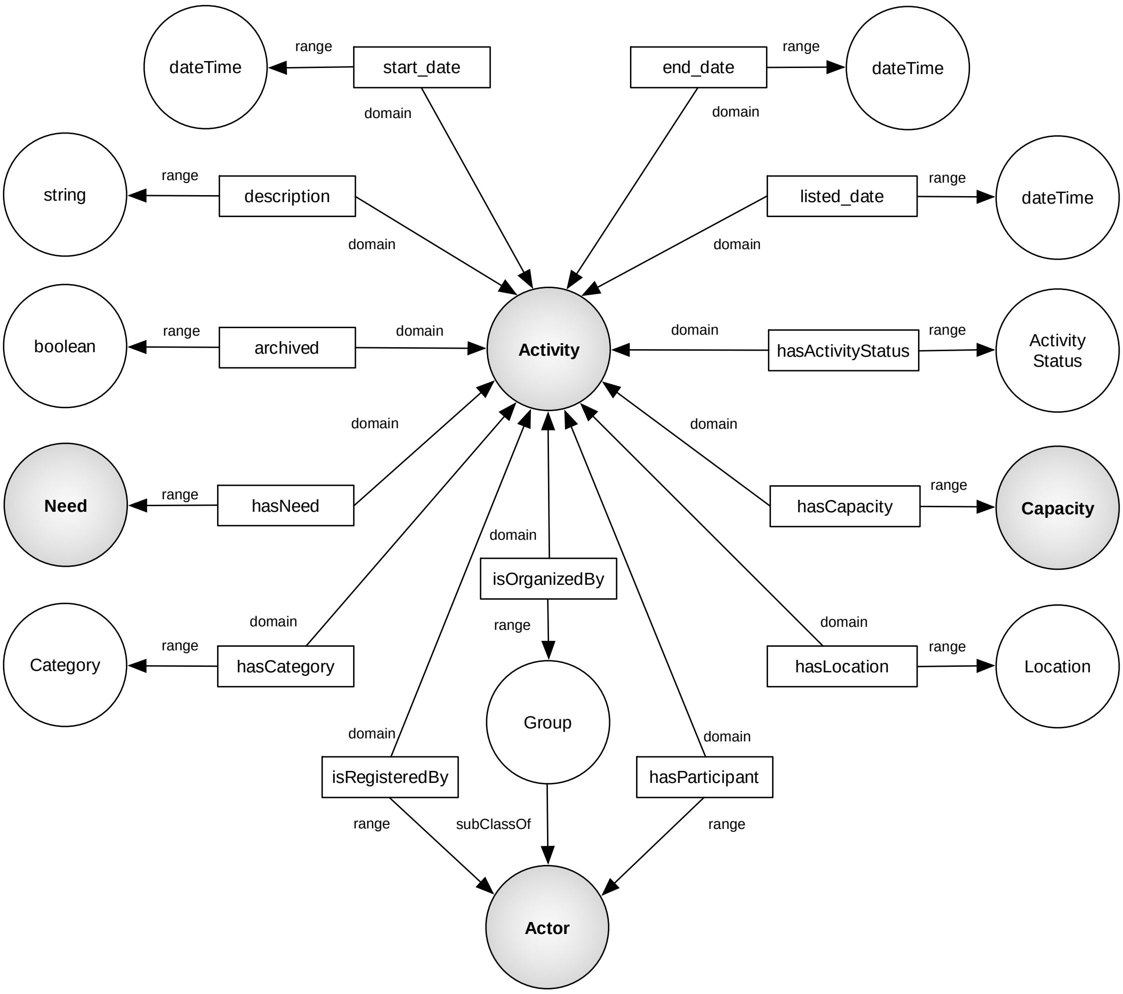


Figure 5: RDF Schema Diagram for *Activity* Concept

As suggested, the *Activity* class is intrinsically linked to the *Need*, *Capacity* and *Actor* classes, through various relationships that are dependent on both the registration of the *Activity* on the COBACORE platform and the current state of progression of the *Activity*. An *Actor* (or *Group*) will perform registration, or be considered as the organizer, of an instance of an *Activity*. Likewise, an *Actor* may also be a participant in an instance of an *Activity*. In addition, an *Activity* is associated with one or more *Need* and, if available, one or more corresponding

Capacity. However, if no *Capacity* is suited as a match for a *Need*, an instance of an *Activity* may still be registered in order to identify and further assign a suitable *Capacity* once available. Consequently, the relationships between an instance of an *Activity* and instances of an *Actor*, *Need* and *Capacity* must eventually be specified in order to establish the underlying rationale and context for the corresponding action within a given situational scenario. Furthermore, the information model for the *Activity* class contains relationships to a number of subsidiary classes, including *Category*, *Activity Status* and *Location*, along with a number of explicit attributes such as *archived*, *description*, *start date*, *end date*, *listed date*, which permit the COBACORE platform to further monitor, analyse and visualize the status of any registered activities.

3.2.4. The Information Model *Actor* Class

In a similar manner to the *Activity* class, the *Actor* class also potentially provides direct relationships between all core concepts, as illustrated in Figure 6.

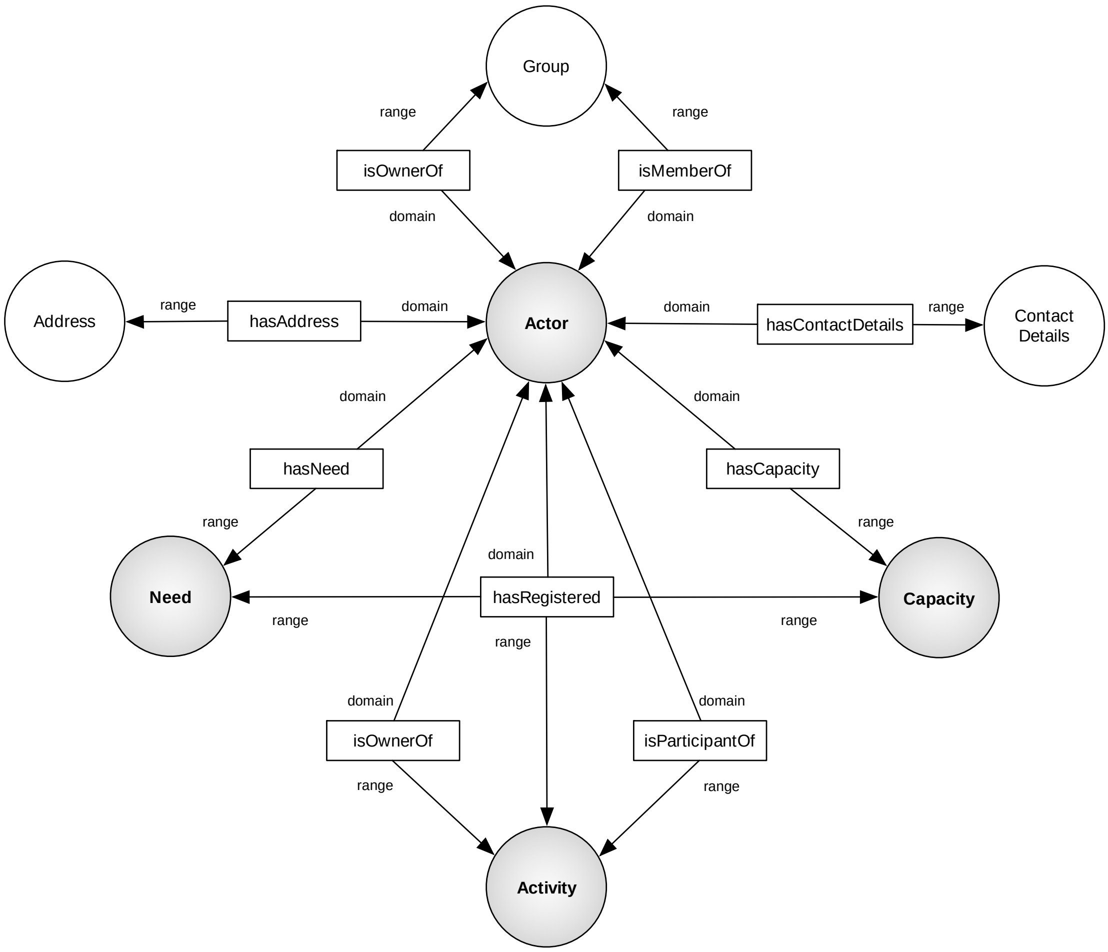


Figure 6: RDF Schema for *Actor* Concept

Within the COBACORE platform, the *Actor* concept can constitute a single person or a group. A group can contain a number of individuals or a single person with virtual members. Subsequently, an instance of an *Actor* can register, be a participant in, or be an owner of an

instance of an *Activity*. An *Actor* can also be the owner of, or member of, a *Group*, thus the ontology and underlying data model provide a number of relationship possibilities in terms of involvement between the *Actor*, *Activity* and *Group* classes. In addition to registering an instance of an *Activity*, an *Actor* may also register an instance of a *Need* or a *Capacity* on the COBACORE platform, and can also be associated with the requirement for a *Need* or provision of a *Capacity*, independent of the corresponding registration. The subsequent extensible approach to the representation of the *Actor* concept, within both the information model and data model, facilitates a degree of flexibility for the types of user (i.e. Affected Community, Responding Community and Professional Responders) that the COBACORE platform is able to accommodate. As can also be observed in Figure 6, the *Actor* class also provides relationships to the *Address* and *Contact* Details classes, rather than using explicitly specified attributes, due to the composite nature of the associated information.

3.2.5. The Information Model *Group* Class

As previously discussed, within the COBACORE platform, the *Group* concept is considered as a specialized form of the *Actor* concept, whereby the information model subclasses the *Actor* class in order to form the *Group* class. Subsequently, all relationships and attributes associated with the *Actor* class are inherent within the *Group* class, in conjunction with a number of additional relationships and attributes specific to the *Group* class, as illustrated in Figure 7.

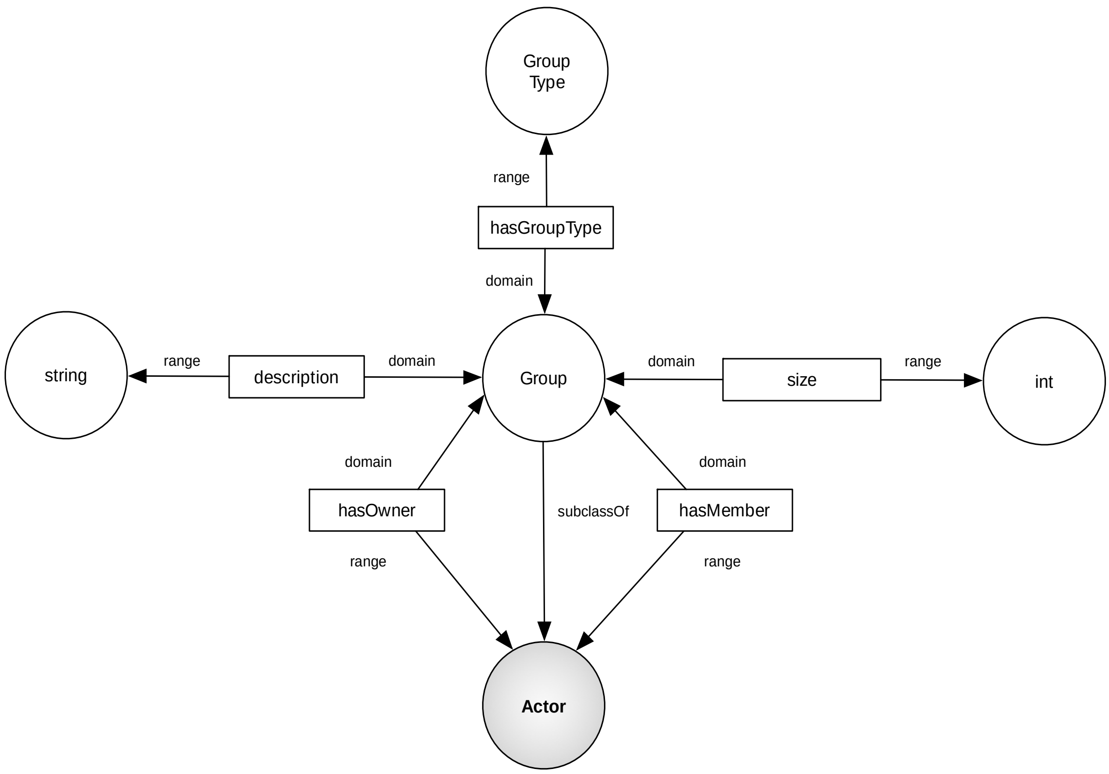


Figure 7: RDF Schema Diagram for *Group* Concept

Within the information model, the relationships between the *Group* class and the *Actor* class are the inverse of the corresponding relationships between the *Actor* class and the *Group* class, thus permitting instances of the *Actor* class to be inferred from instances of the *Group* class and vice versa. An additional relationship is formed between the *Group* class and the GroupType enumeration, which allows an instance of a *Group* to be characterized using a predefined literal as a *community group*, *family*, *network* or *organization*. Furthermore, as Figure 7 illustrates, the attributes *description* and *size* can also be explicitly specified for an instance of the *Group* class within both the information model and data model in order to provide additional details for the COBACORE platform.

3.2.6. The Information Model *Category* Class

In addition to the classes relating to the core concepts, the information model provides a *Category* class, as previously discussed, which facilitates categorization of a *Need* or *Capacity* that has been registered on the platform. Figure 8 provides a high-level illustration of the Category class, which illustrates the set of first-level categories represented by the information model.

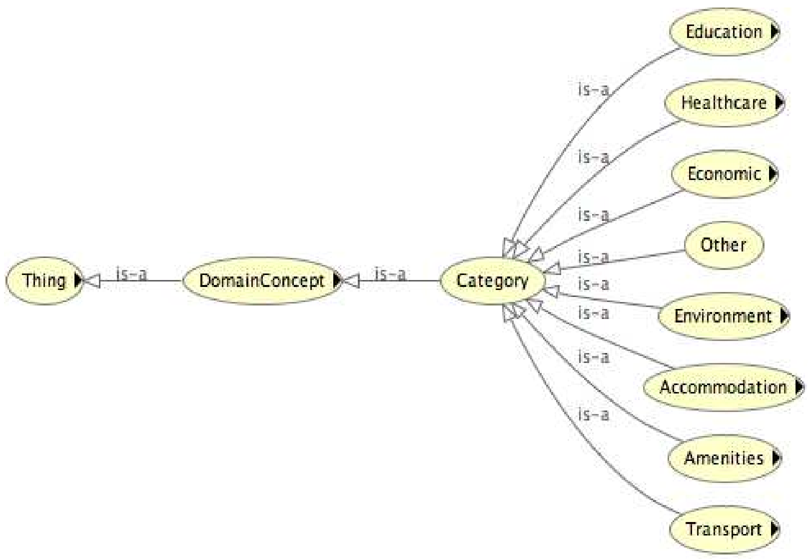


Figure 8: High-Level View of Category Class

From Figure 8 it may be observed that the set of (*first-level*) subclasses based on the Category class are utilized to represent the various recovery domains associated with the COBACORE domain, e.g. *Accommodation*, *Amenities*, *Economic*, *Education*, *Environment*, *Healthcare*, *Other* and *Transport*. An example of the corresponding set of (*second-level* and *third-level*) subclasses associated with a first-level subclass, Transport, is subsequently illustrated in Figure 9.

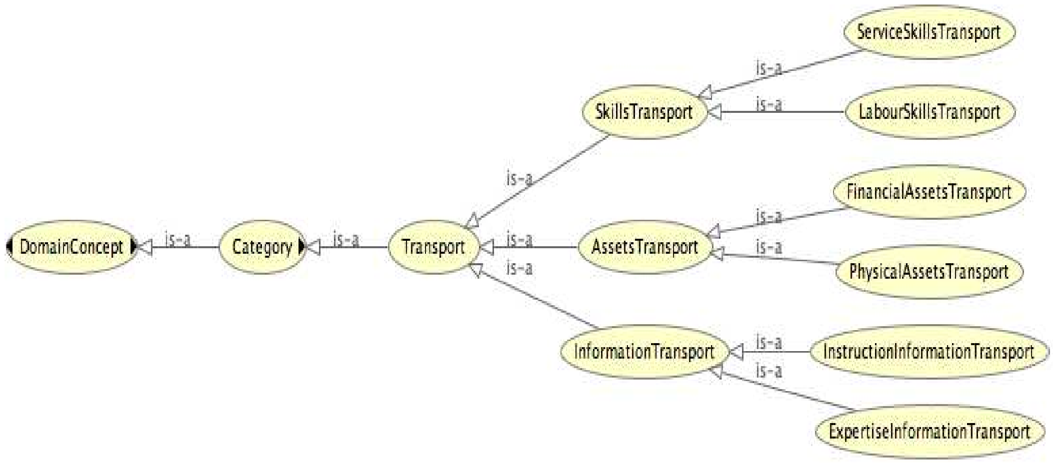


Figure 9: Example Set of Subclasses for the Transport Category Class

As Figure 8 and Figure 9 indicate, each first-level class within the information model also has an associated set of subclasses, which have been determined using a conjunction of the initial set of categories representing the recovery domains and the *Type* classes, e.g. *Assets (*Physical*, Financial)*, *Information (Instruction, Expertise)*, and *Skills (Service, Labour)*. Subsequently, each first-level category will have a set of second-level and third-level sub-categories that represent all possible conjunctions between the recovery domains and *Type* class. The exception to this is the first-level category, *Other*, which is a generalized class that is used for categorization when no other sub-class is deemed appropriate. Within the current iteration of the information model, each third-level subclass contains a number of individuals that form a vocabulary associated with the subclass, which are utilized for the purpose of categorization of a registered *Need* or registered *Capacity*. Subsequently, although a user of the COBACORE platform may explicitly select one or more of the first-level categories, and one or more types during the registration of a *Need* or *Capacity*, the *Category* class of the information model will be subsequently used to ascertain all potential categories (i.e. sub-classes) that the *Need* or *Capacity* may belong to, based on semantic similarity matching between various facets specified for the *Need* or *Capacity*, the explicitly selected categories and the sub-class vocabularies. This implicit categorization facilitates a degree of optimization for the sense making aspect of the COBACORE semantic framework, as discussed later in Section 4.1.

4 The COBACORE Semantic Framework

In order to integrate the information model with the COBACORE platform, the use of a suitable semantic framework is required for the primary purposes of hosting the ontology and facilitating access by means of SPARQL queries. To achieve full compatibility with the COBACORE platform, the use of two .NET-based frameworks have been used: (1) BrightstarDB, which provides a triplestore and related tools suitable for hosting the information model; (2) dotNetRDF, which provides a suite of .NET libraries that support SPARQL-based queries of a triplestore, along with reasoning functionality. Correspondingly, an overview of the architecture for the semantic framework utilized by the COBACORE platform is given in Figure 10.

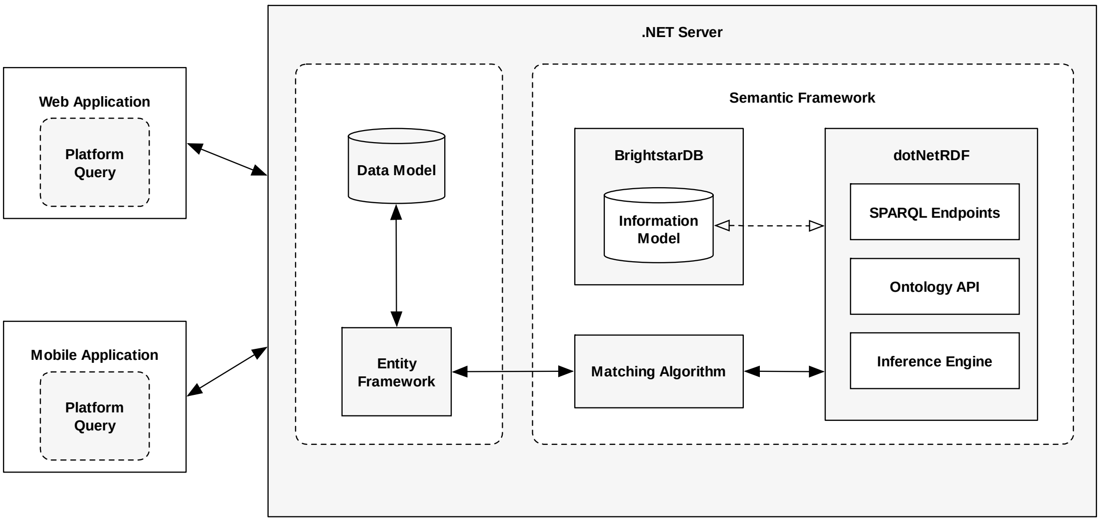


Figure 10: COBACORE Semantic Framework Architecture

From Figure 10 it can be observed that the relational database employed by the COBACORE platform (i.e. the Data Model) is hosted, along with the components comprising the semantic framework, using a .NET server. Communication of user and situation information between the platform and the database is directly carried out via SQL queries, which are also used to provide the semantic framework with specific information from the underlying data model when necessary. Notably, the semantic framework utilizes SQL queries in order to obtain the instances of a *Need* and *Capacity* during the matching process (as detailed in Section 4.2). In terms of the information model, the ontology is hosted within an RDF triplestore using the BrightstarDB libraries, which also provide access to the triplestore via a set of SPARQL endpoints. However, due to the current lack of support for an inference mechanism within BrightstarDB, the SPARQL endpoints provided by the dotNetRDF libraries are utilized to access the triplestore. Further manipulation of the information model may be carried out programmatically using the *Ontology API* provided by the dotNetRDF libraries.

In addition, the dotNetRDF libraries also provide an *Inference Engine* that permits RDFS-based (class-subclass) inference to be conducted on the information model. Such inference can be

utilized by the platform through the SPARQL endpoints in order to validate relationships between concepts represented as instances of the data model. The semantic framework can also utilize the inference capability programmatically to categorize instances of a registered *Need* or *Capacity* prior to the matching process in order to reduce the number of initial candidates for matching. While this approach was initially implemented, it was subsequently discovered during in-house testing that such implicit categorization was potentially detrimental to the successful operation of the prototype platform, hence it was revised prior to the final evaluation (as discussed in Section 5).

4.1 Sense Making in the Semantic Framework

The information framework developed as part of the COBACORE platform exposes a number of requirements, ranging from the necessity of capturing the needs and capacities that arise across affected communities, to the need for enhancing activity generation through the information modelling to conduct matching of needs and capacities.

The Need-Capacity matching algorithm underlying the COBACORE information framework has been conceived as a response to these requirements, being aimed at facilitating the task of identifying potentially relevant capacities in response to a target need across communities. The use of the algorithm potentially mitigates an arduous problem users would be otherwise faced with: identification of important and relevant information within the platform from an overwhelming – and ever growing – amount of available material. To do this, the approach employed within the COBACORE platform is composed of the following stages:

1. **Semantic Matching,** which exploits the COBACORE information model to infer candidate capacities for a target Need based on the common categories and subcategories they might belong to;
2. **Multi-Criteria Matching**, which applies a Multi-Criteria Decision Making approach to further analyse differences in those candidate capacities obtained from (i), predicated on the 4W (Who, What, Where, When) approach.
3. **Candidate Ranking,** which ranks all candidate capacities according to their overall (aggregated) matching degree with the target Need, whereby the set of highest matching capacities is returned by the algorithm.

As a result, a ranked list of relevant *Capacity* instances associated with a registered *Need* is obtained. In a similar manner, the algorithm is also utilized within the COBACORE platform to determine a set of candidate needs for a target *Capacity*, thereby returning a ranked list of relevant *Need* instances associated with a registered *Capacity*. In both cases, the results of the matching process are displayed to the user through the user interface (UI) of the platform. A general overview of the matching process is illustrated in Figure 11.

As may be observed in Figure 11, the Semantic Matching and Multi-Criteria Matching stages are dominant within the approach employed. During the matching process, a semantic inference matching phase (i.e. *Semantic Matching*) is firstly performed in order to filter a

subset of candidate capacities in response to a target *Need* $n_i \in \{n_1, \ldots, n_m\}$ based on the categories or types to which they belong.

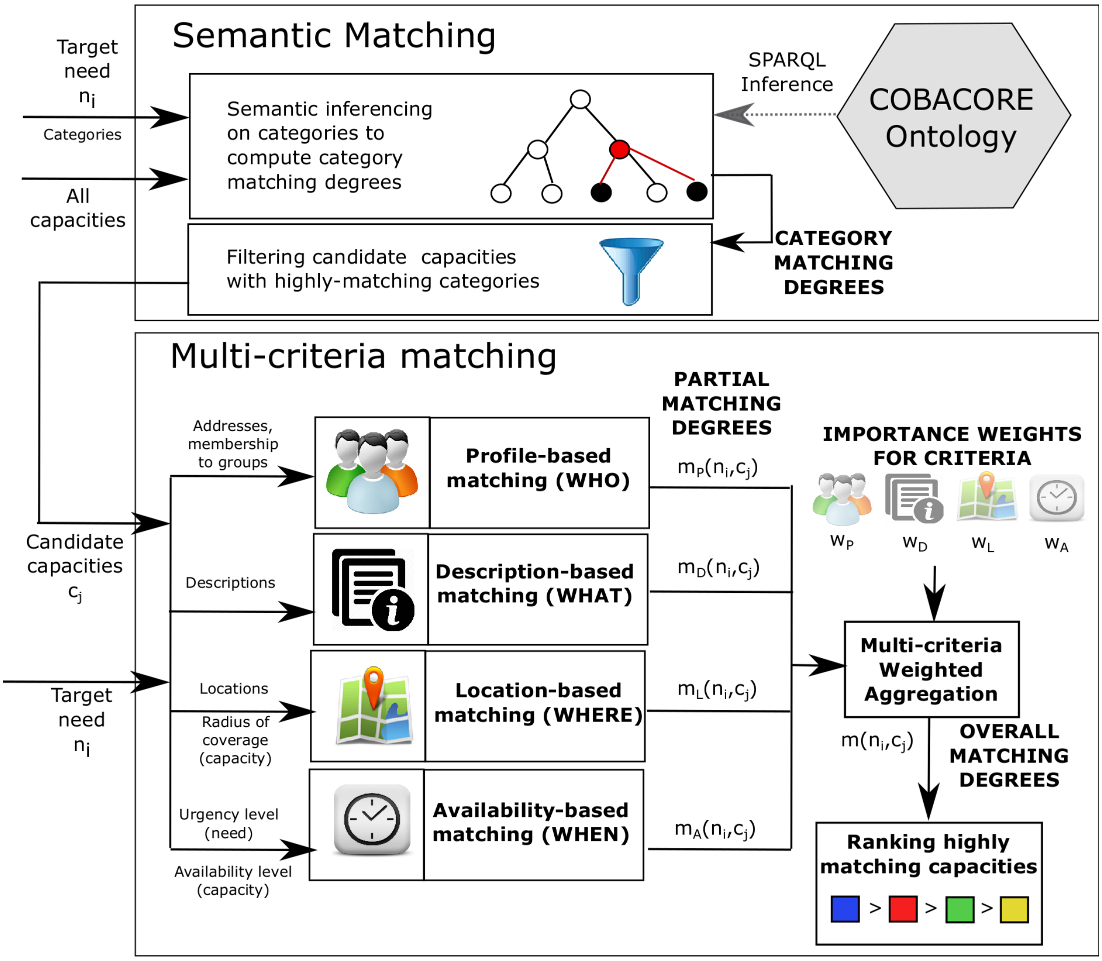


Figure 11: Overarching Need-Capacity Matching Process

Following this, criteria-based matching (i.e. *Multi-Criteria Matching*) is carried out utilizing the 4W (*Who*, *What*, *Where*, *When*) facets of each candidate in order to determine the partial matching degrees $m_k(n_i,c_j) \in [0,1]$ between n_i and each candidate capacity $c_j \in \{c_1,\ldots,c_z\}$. Once all partial matching degrees are computed for a candidate capacity, they are then aggregated into an overall matching degree $m(n_i,c_j)$ for the candidate capacity, which are subsequently utilized in order to provide a ranked set of candidates to return.

4.2 Multi-Criteria Decision Making (Matching Algorithm)

As previously discussed, the Need-Capacity matching algorithm of the COBACORE information model takes as input a *Need* that has been previously registered by a user (*Actor*) of the platform, and outputs a subset of capacities that might provide a suitable way to respond to such a *Need*, in terms of several factors such as: relevance, availability, closeness, etc. The basic approach for implementing this algorithm is based on information fusion methods for

Multi-Criteria Decision Making (MCDM), which refers to a family of methods used to enable effective decision support given the presence of several, often conflicting, criteria[3]. While a number of methodological approaches for criteria aggregation exist, dependent on the nature of the problem, the Need-Capacity matching algorithm employed by the COBACORE platform is inspired by Multi-Attribute Utility Theory, which extends classic utility theory to multiple criteria and represents preferences as values or utility functions[4]. In this particular case, the utility (i.e. matching degree) of a set of alternatives (i.e. candidate capacities) is evaluated against several criteria independently based on existing facets of information relating to a *Need* or *Capacity* (e.g. user description, location, urgency, etc.).

Accordingly, a MCDM framework may be formulated as follows:

There exists a decision problem consisting of $z \geq 2$ alternatives or possible solutions, $X = \{x1,...,xz\}$

Alternatives are assessed according to several independent criteria, $Q = \{q1,...,qn\}$, $n \geq 2$.

Let $m_{jk} \in D$ denote the degree (*utility*) to which alternative $x_j \in X$ matches or satisfies criterion $q_k \in Q$, expressed in an information domain D such that $m_{jk} = 1$indicates that x_j completely satisfies q_k, whereas $m_{jk} = 0$ indicates no satisfaction of the criterion by x_j . To obtain an overall assessment value m_j for x_j, it is necessary to combine its satisfaction degrees m_{jk} over the different criteria q_k using an aggregation function f utilized to compute $m_j = f(m_{j1},...,m_{jn})$. Once a satisfaction degree m_j for each alternative is obtained, it can be subsequently used to determine and select the best alternative as the solution for the problem, or to rank several alternatives depending on the nature of the decision problem.

Correspondingly, the MCDM approach developed for use within the COBACORE platform is illustrated in Figure 12. As shown, four partial matching degrees between the *Need* and each candidate *Capacity* are computed according to the 4W criteria. A flexible aggregation procedure is then applied in order to combine partial matching degrees into a global matching degree for each *Capacity*, such that a ranked list of the most highly matching capacities can be returned to the user.

[3] M. Doumpos, E. Grigoroudis, Multicriteria Decision Aid and Artificial Intelligence, Wiley, 2013.

[4] R. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Cambridge University Press, Cambridge, 1993.

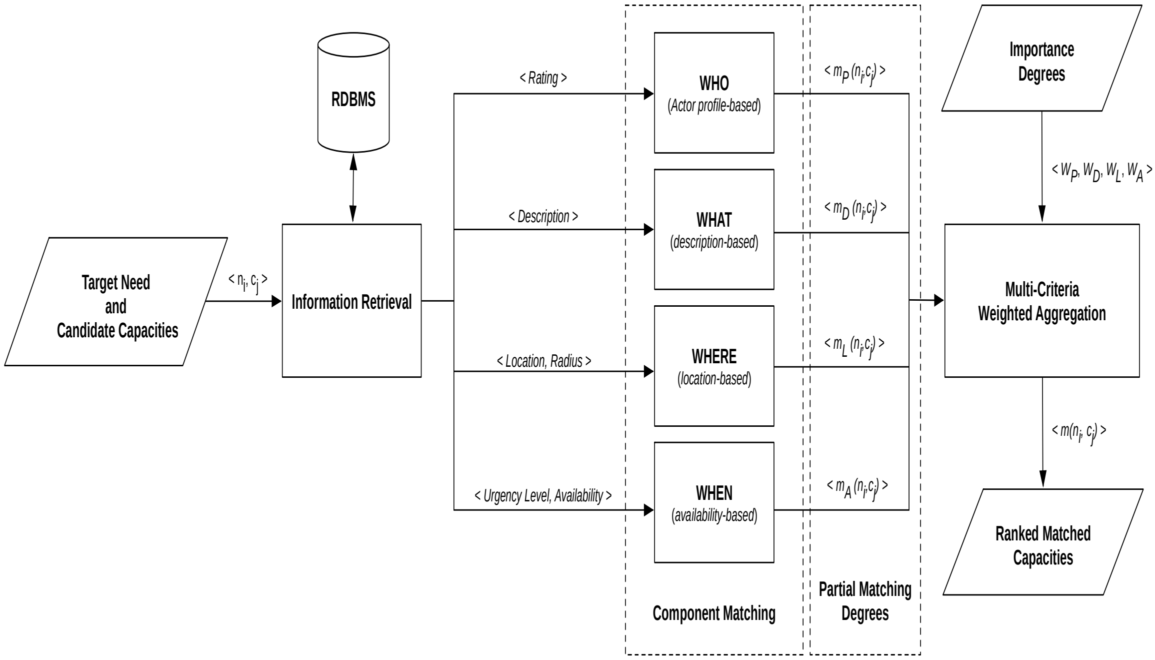


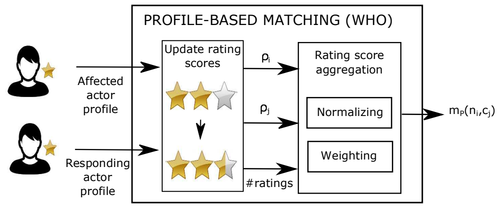
Figure 12: Base Multi-Criteria Matching Algorithm

As a consequence of the iterative development process, this approach was later extended in order to apply inference-based matching based on categories and subcategories to which needs and capacities belong, as previously discussed in Section 4.1 and illustrated in Figure 11. However, as previously mentioned, run-time issues arising from this process, along with the use of the matching algorithm, necessitated a later revision of the platform implementation, as subsequently discussed in Section 5.

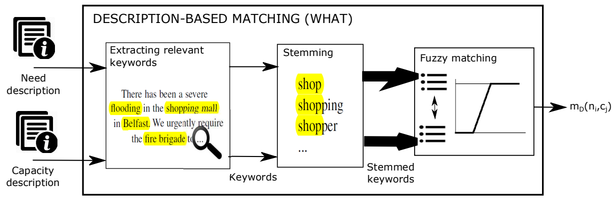
4.2.1. Multi-Criteria Matching

The four matching degrees are all calculated as numerical values in the [0,1] interval, such that the higher the value, the more a given *Capacity* matches the target *Need*. However, the process to obtain them is different for each of the 4W criteria, as illustrated in Figure 13, due to the diverse nature of the existing data related to each criterion in the database:

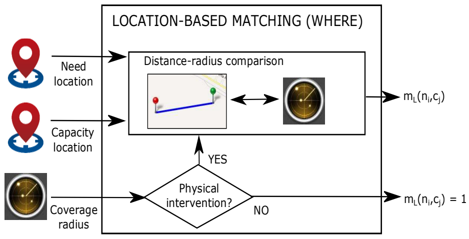
- *Profile-based matching degree (WHO):* This matching degree is computed by taking into consideration the user profile information of the two involved actors, i.e. those who introduced the target *Need* and the candidate *Capacity*, p_i and p_j respectively, as shown in Figure 13(a). More specifically, the average level of reputation (or trust) between the *Actors* is taken into consideration in accordance with an aggregation operator that may be modified to reflect the attitude required by the profile-based matching degree: a higher average *rating* for the individual *Actors* implies a potentially greater influence in the computation of the matching degree.



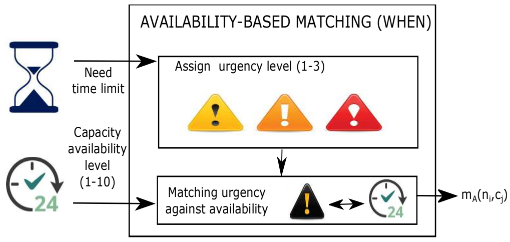
(a) WHO
Average rating score



(b) WHAT
Fuzzy set membership



(c) WHERE
Physical intervention independent coverage



(d) WHEN
Urgency focused time limit

Figure 13: Individual Matching Degrees (using 4W Approach)

- *Description-based matching degree (WHAT):* Descriptions of needs or capacities consist of free text introduced by users as a *Need* or *Capacity* is registered on the platform. The procedure to calculate how similar two descriptions are aims at finding common keywords within the two descriptions. Thus, as illustrated in Figure 13(b), Natural Language Processing techniques are employed, along with fuzzy set theory, to compute the matching degree using a three step process:

  1. Extract relevant keywords: A Part Of Speech tagging algorithm is used to tag each word within the description of the Need/Capacity according to a predefined grammar; words with the least relevance to the overall description meaning (e.g. articles, prepositions, etc.) are subsequently removed prior to the next step of the process.
  2. Stemming: In order to increase the detection of similar words, or families of words, between descriptions, an approach based on the Porter stemming algorithm5 is applied to the remaining words to determine their lemma, i.e. basic form within the corresponding word-family. The set of stemmed keywords is then used in the final step of the process.

[5] M. Porter, An algorithm for suffix stripping, Program 40(3) (1980) 211–218.

3. Determine fuzzy matching degree: Computation of the description-based matching degree consists of determining the degree to which the number of common (stemmed) keywords complies with the concept of similar. Due to the vague and imprecise nature of this concept, fuzzy set theory6 is utilized to define a fuzzy set SIMILAR in the [0,1] interval with a semi-trapezoidal membership function, which is subsequently used to determine the matching degree.

- *Location-based matching degree (WHERE)*: Needs and capacities' data include geographical information about the location where they are registered, which does not necessarily correspond to the associated actors' locations. As shown in Figure 13(c), the location-based matching degree is calculated based on the distance between the geographical locations (i.e. points in the WGS84 coordinate reference system), and the radius of coverage specified by the *Capacity* (if any), which indicates the maximum distance from the capacity's location in which it can be applicable. Correspondingly, if the distance between a target *Need* and candidate *Capacity* exceeds the radius of coverage, the value of the matching degree is zero. However, as some capacities do not necessarily require physical intervention in the location of the *Need* (e.g. providing information resources, bank transfer, telephone or internet services, etc.) they have no specified radius of coverage and are assigned a maximum value for the matching degree.

- *Availability-based matching degree (WHEN)*: As illustrated in Figure 13(d), the computation of this matching degree considers two aspects; (1) the *urgency level* associated with a *Need*, (2) the *availability level* specified for a *Capacity*. On the one hand, the urgency level indicates the time limit under which responders are required to resolve the *Need*. Based on the date and time limit specified by the *Actor* who registered the *Need*, comparing it with the current date and time, an urgency level, urg_i, is assigned to the *Need*, which takes a value within a numerical scale from 1 to 3 such that $urg_i \in \{1 : \text{LOW } (7 + \text{days}), 2 : \text{MED } (1 - 7 \text{ days}), 3 : \text{HIGH } (< 24 \text{ h})\}$. On the other hand, the availability level associated with a Capacity, $avail_j \in \{1,2,...,10\}$, is registered by the associated responding *Actor*, such that the higher $avail_j$, the more availability a *Capacity* has to respond to an urgent *Need*. Under the premise that highly urgent needs should be matched more exclusively to the most available capacities, an urgency-based matching degree is subsequently computed that penalizes low-availability capacities in favour of those that are highly available.

As previously illustrated in Figure 12, the final step of the matching algorithm consists in combining the four partial matching degrees, by means of an aggregation operator, in order to obtain the overall matching degree between the target *Need* and each candidate *Capacity*.

[6] L. Zadeh, Fuzzy sets, Information and Control 8(3) (1965) 338–353.

Consequently, as different degrees of importance may be assigned to the 4W criteria, we employ a weighted average operator to compute the overall matching degree:

$$m(n_i, c_j) = \frac{\sum_{k=1}^{4} w_k m_k(n_i, c_j)}{\sum_{k=1}^{4} w_k}$$

where k ∈ {P: *Who*, D: *What*, L: *Where*, A: *When*}. Thus, some of the 4W criteria can be given greater importance than others in order to identify the most highly matching capacities in response to a given *Need*. This procedure is inspired by aggregation approaches in MCDM; within the context of the COBACORE information model, the aggregation operator provides the following features to enhance flexibility in its future use:

1. Although a weighted average aggregation operator is applied by default, the design of the matching algorithm facilitates future extension of the use of new aggregation operators that are potentially more suited to reflect different decision attitudes in the aggregation process (e.g. optimistic, pessimistic, etc.).
2. A weighted aggregation process has the remarkable advantage that it assigns importance weights to criteria, therefore it easily permits reflection of any prioritisation order between them, such that those criteria that should have more influence in obtaining the overall matching degree are assigned a higher importance weight. Moreover, the platform administrator may flexibly modify these importance weights if potential changes in the way criteria should be prioritised arise as a result of new decision scenarios.

Finally, applying a simple ranking procedure based on the overall matching degree, a list of highly matching candidate capacities is returned to the *Actor* who registered the target *Need*.

4.2.2. Semantic Matching (Pre-Filtering)

As previously mentioned in Section 4.1 and illustrated as *Semantic Matching* in Figure 11, the use of a semantic inferencing mechanism was initially incorporated into the overarching matching process in order to reduce the (presumably large) amount of information about existing capacities registered on the platform, filtering out semantically irrelevant capacities for the target *Need*, and consequently reducing the computational cost of the subsequent multi-criteria matching algorithm. Accordingly, the semantic matching phase is applied to identify a subset of candidate capacities based on the common categories (i.e. disaster recovery domains, as defined by the *Category* hierarchy previously discussed in Section 3.7) to which they belong. To this end, a semantic data-store and the COBACORE ontology are utilized to analyze the categories to which needs/capacities belong (the SPARQL language is used for semantic querying over the data-store).

The following steps were applied prior to use of the matching algorithm in order to obtain a suitable subset of candidate capacities:

- *Implicit Categorization*: When a *Need/Capacity* is registered on the platform, its descriptions are automatically analysed (using keyword-based Natural Language

Processing similar to that used to determine description-based matching degrees), to tag them as belonging to one (and sometimes multiple) specific categories within the given *Category* hierarchy.

- *Inference Phase*: SPARQL queries are conducted to infer the categories and subcategories within the *Category* hierarchy to which each *Need* and *Capacity* belongs. This inference process allows the platform to discover new knowledge that otherwise would not be explicitly exist as part of the *Need/Capacity* information introduced.

Those capacities with at least a (previously established) threshold degree of similarity with the target *Need* in regard to the *Category* hierarchy are subsequently utilised as candidate capacities for the multi-criteria matching process. As an example, Figure 14 illustrates part of the category hierarchy defined within the ontology (edges represent *subClassOf* relations).

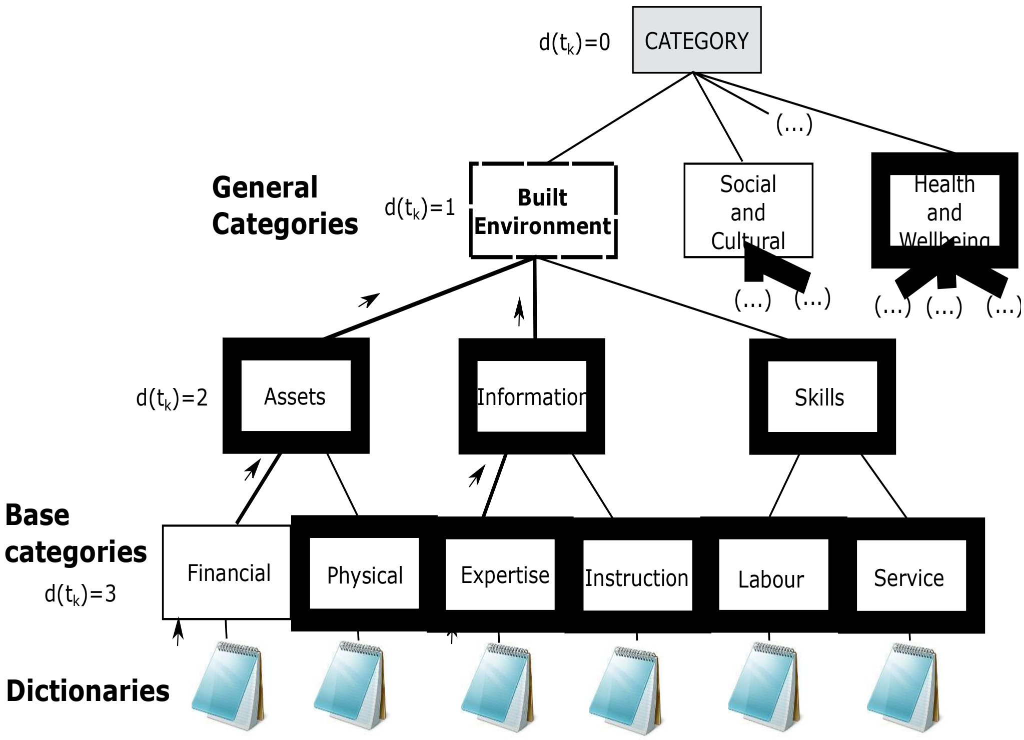


Figure 14: Example of Category Hierarchy for Semantic Matching

From Figure 14, it can be seen that assigned categories are utilized in conjunction with the ontology in order to apply an inference process aimed at extracting additional knowledge about higher-level categories to which a target *Need* or candidate *Capacity* potentially belongs. The inference process facilitates computation of a category-based matching degree, particularly in the case when no explicit categorization was performed by the user (for example, in Figure 14 two base categories are compared via inference, identifying *Built Environment* as a common ancestor; without inferencing, this matching would be disregarded). Additionally, given the tree structure of the category hierarchy, matching degrees are

computed based on the position of the Lowest Common Ancestor[7] (LCA) for each pair of categories assigned to the *Need* and *Capacity*. Since every category is associated to a node in the category hierarchy, applying an upwards inference process against the hierarchy allows higher-level categories to be obtained and, consequently, the LCA category (*node*) for every pair can be easily determined. The category-based matching degree between a target *Need* and candidate *Capacity* can then be computed and used to obtain a sub-set of the top matching (category-wise) candidate capacities, which are subsequently used as the input set of candidate capacities for the multi-criteria matching phase of the matching process.

[7] M. A. Bender, M. Farach-Colton, G. Pemmasani, S. Skiena, P. Sumazin, Lowest common ancestors in trees and directed acyclic graphs, Journal of Algorithms 57 (2) (2005) 75 – 94.

5 Platform Development & Revised Architecture

From early stakeholder analysis it was clear that the type of end user, both from the local community and professional user groups, would differ greatly in terms of technical ability. Hence an overriding concern when putting such a multifaceted algorithm into practice, was for the complexity of the matching algorithm to be hidden from the end user as much as possible. Finding a 'lowest common denominator', in terms of User Interface (UI) design and device accessibility, for both collecting the data required by the algorithm and communicating the results of the algorithm, was a priority when creating the *face* of the algorithm that 'ordinary' users interacted with. Another consideration for the operational implementation of the algorithm and the effective use of its outputs is the UI requirement for the immediacy of the algorithm's results. Accordingly, such results must be made immediately available to users, presented clearly, and provide utility; these considerations have knock-on effects in the way that algorithm was realized. To aid the 'immediacy' of results of the algorithm to the UI, the results must be stored in a high availability data-store, with asynchronous processes made available to alert the UI of any changes to the results. To give the results utility for the users, various UI techniques were employed to provide context for the results, e.g. through showing the results on a map-based interface, and to promote a 'call to action' for each pertinent result. With these design considerations in mind, we set to work to implement the processes that could effectively take inputs from users, process this data, and then to output it in a format that could provide utility to a number of different user groups. Throughout the overarching technology implementation an iterative approach to platform development was undertaken, which employed a series of in-house tests of both platform component integration and platform load, in order to ensure the prototype platform was 'fit for purpose' during the Final Evaluation session (and moving forward beyond the project). Accordingly, issues arising from both Partial Evaluation of the platform and in-house testing could be subsequently addressed in the final version of the platform.

Although an ambition of the implementation was to hide the complexity of the algorithm from the end user, the methodology of using the 4W approach to the criteria of the matching algorithm was also used to logically divide the steps users would carry out during data acquisition. Logically dividing the information in this way lent itself to implementing a 'wizard' style of interface (in both the web platform and mobile device) for users to register and input details for both needs and capacities. To assist in making the user data collection as efficient as possible, geolocation and other auto-generated fields were employed as much as possible throughout the UI. Where possible, the user's location was automatically determined using GPS; if it could not be determined, geolocation aids such as Google Places and an editable map were utilized to assist users in determining location (a screen-shot of the Where criteria during registration of a *Need* is given in Figure 15). Pieces of information which were deemed interesting but not essential to the algorithm, e.g. date of birth, etc., were either made optional or removed from the UI.

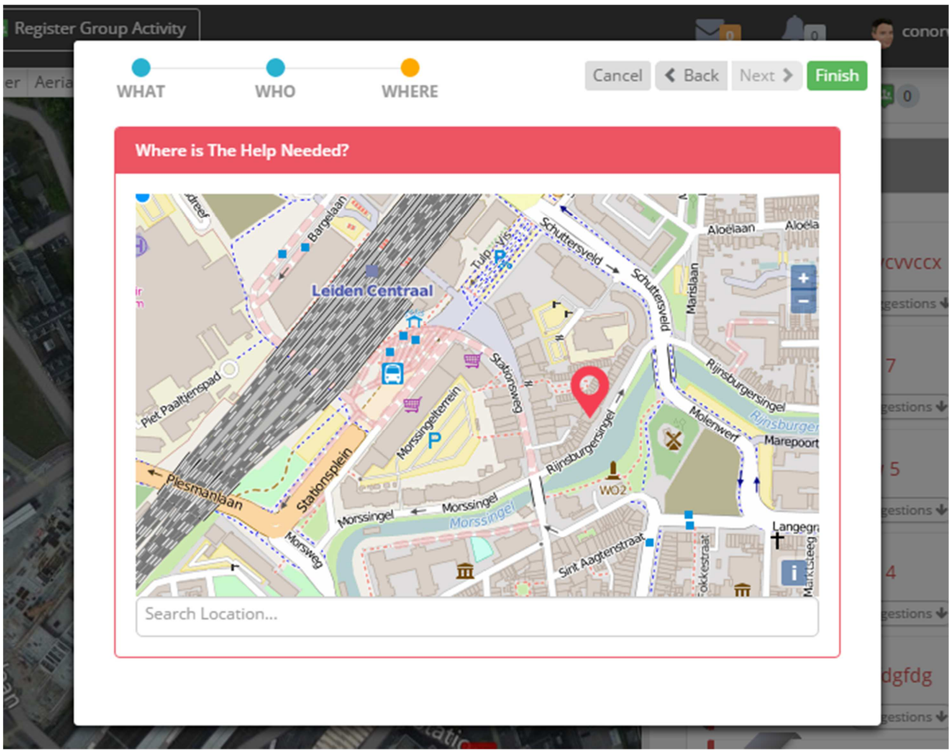


Figure 15: Example of UI for 'Where' during Need Registration

While the data collected from the data acquisition processes is then stored in a relational data store that can be accessed by the algorithm, the algorithm is itself triggered by a two different events:

1. When a new Need/Capacity is registered through the UI.
2. When an existing Need/Capacity is modified through the UI.

When either trigger occurs, the matching process will be performed to evaluate the additional/modified *Need/Capacity*, assign it an overall matching degree against every other existing Capacity/Need currently registered within the platform and stored in the datastore. These matching degrees are then saved in a high availability datastore, and subsequently consumed by the web and mobile clients in a number of different ways:

1. Suggestions: The main use of the matching degrees is to show the user a list of the most relevant suggestions for those capacities that could help them when they submitted a Need (and, vice versa, to show users who registered a Capacity the most relevant set of existing needs). These ranked lists of matching needs/capacities were shown to users immediately after they had registered a Need/Capacity and additionally highlighted on a map (in relation to the user's current location) in order to ascertain the most suitable Need/Capacity available, given their current circumstances. Figure 16 provides a screen-shot of a ranked list of potential capacities for a given Need.

2. Nudging/Notifications: Another use of the matching degrees was to alert users when a new match for their Need/Capacity was identified. If a new (or modified) Need/Capacity matched highly with a user's Capacity/Need, a notification is sent to the user (via email and directly on the UI).
3. Call-to-Action: For both displaying suggestions and notifications it is also important to offer a call to action to the user so he/she can easily act upon this relevant piece of information. In this case the call to action afforded the user the opportunity to privately message the originator of the Need/Capacity in order to further discuss the match.
4. Metrics/Overview: The resulting matching degree data can also be used to give responding users a situational overview. At a glance, responding users can determine which needs and capacities are failing to find high scoring matches, i.e. quickly identify outliers which may struggle to find a match. In addition, this information can also be cross referenced against other sources of information available through the platform, e.g. open data map layers, in order to identify trends and influence decision making.

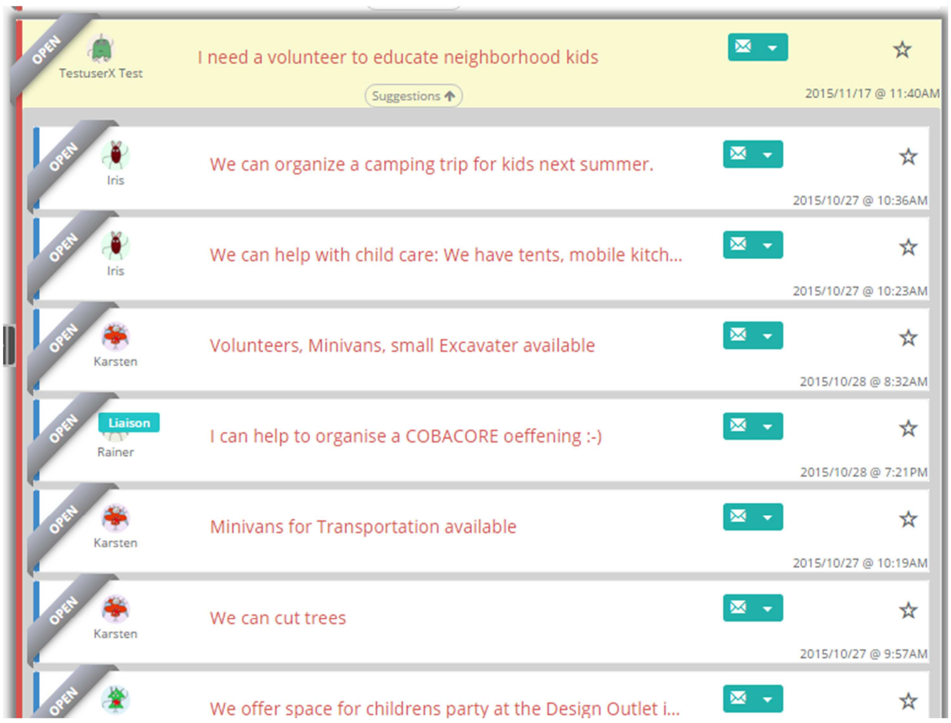


Figure 16: Example of Capacity Suggestion List in Response to a Registered Need

During the development iteration between the last Partial Evaluation and the Final Evaluation event, a key technical issue arose in regard to the operational capacity of the platform. This necessitated a further revision of the platform architecture and underlying semantic matching processes. In particular, during an in-house testing session that occurred after the complete integration of the Semantic Matching component of the platform, it was discovered that the capacity of the platform to support increasing numbers of users was somewhat diminished by

the integration of the matching algorithm; this was especially the case when new users started registering new *Need* and *Capacity* entries on the system, which led to internal processing delays incongruent to the operational requirements of the platform. Correspondingly, it was determined that the functionality of the matching process and semantic platform was suboptimal in terms of the response speed required, which inevitably resulted in the user interface becoming somewhat unresponsive at times. Consequently, the mechanisms inherent in the platform were modified to permit the platform to employ improved concurrency in its operation, thus potentially endowing the overall platform with a higher degree of scalability. In response to the additional concurrency support, a number of key changes to the matching algorithm and semantic framework integration were required as follows:

1. <u>Transactional Concurrency</u>: As the matching algorithm requires access to the data model through repeated SQL calls into the RDBMS associated with the data model, a caching mechanism was implemented to facilitate information reuse. The resulting cache was stored within a relational database table, however, this required further integration of a lock system to prevent issues arising from concurrent access to the table by an arbitrary number of underlying transactions. Such a locking mechanism prevented the potential corruption of data within the cache, along with associated database access violations (which also resulted in data corruption and, at times, system deadlock).
2. <u>Semantic Framework Refactoring:</u> Due to the (existing) limitations of components of the semantic framework, in conjunction with the overheads associated with the underlying communication, it was deemed necessary to remove the use of the dotNetRDF and BrightstarDB components of the platform. Subsequently, the use of these components for implicit categorization was refactored such that the components were replaced by the concatenation of explicit Category and Type values, as specified by the user when registering a Need or Capacity, to the corresponding description prior to the operation of the matching algorithm resulting in an expanded description. Furthermore, a dedicated dictionary was integrated with the platform in order to permit the generation of additional synonyms for all stemmed words, including Category and Type values, contained within the expanded description of a Need or Capacity. Consequently, the NHunspell Library8 was utilized and its thesaurus functionality employed to produce sets of synonyms that were subsequently concatenated to the expanded description prior to matching. While inference was no longer directly applied in order to perform implicit categorization as a pre-filtering stage, the use of an explicit categorization in conjunction with additional synonyms facilitates pre-filtering as part of the normal matching process.

With respect to the changes to the implementation, the revised COBACORE platform architecture is depicted in Figure 17.

[8] http://www.crawler-lib.net/nhunspell

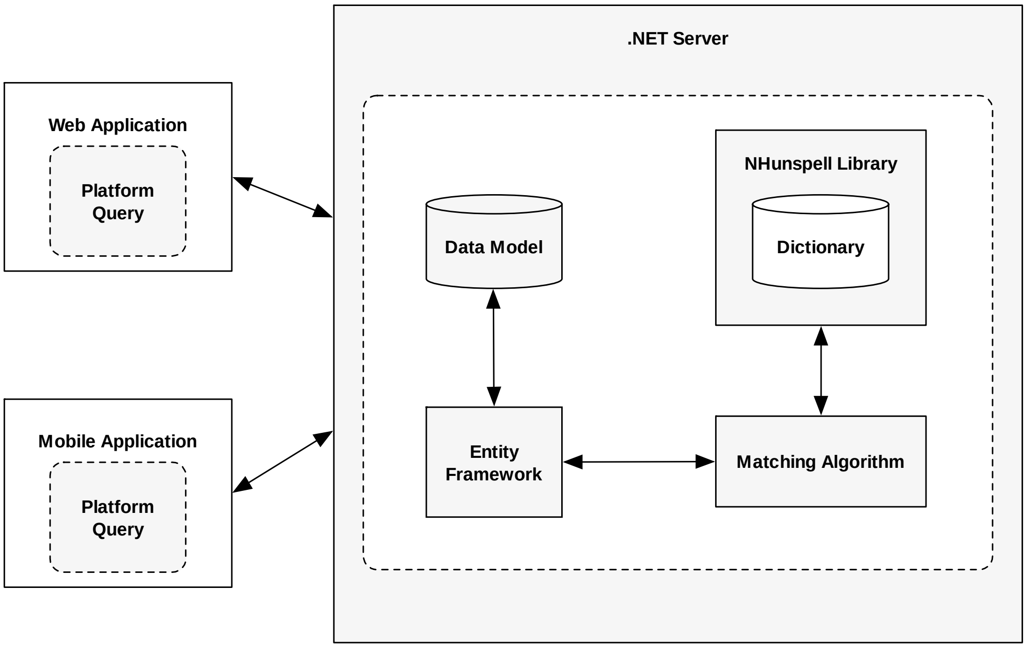


Figure 17: Revised COBACORE Platform Architecture

It can be observed from Figure 17 that the revised platform architecture directly incorporates the NHunspell Library, which is utilized by the matching algorithm. Consequently, the overhead associated with the use of an external semantic framework has been removed. In conjunction with the implementation of transactional concurrency, subsequent in-house testing ahead of the Final Evaluation event demonstrated overall improvements in the operational capacity of the platform, particularly with regard to load testing through increased amounts of user activity. A corollary of the revised architecture is the simplification of the overarching platform architecture, which potentially leads to more effective and direct deployment scenarios.

6 Conclusion

The fully operationalized data framework constitutes the key development outcome from WP2. The final framework specifications, design architecture and operational capacities have been informed and guided by the functional requirements detailed in Deliverable 3.2 as well as feedback from prospective end-users (including disaster recovery professionals) during the partial, mid-term and final evaluation exercises designed and developed in WP5. The COBACORE data framework has been conceived as a much needed technical solution to the requirements of affected communities, facilitating their integration to the disaster recovery and reconstruction process not only as providers of data but as a potentially deployable resource in recovery and reconstruction efforts. From a professional responder standpoint, the COBACORE data framework serves as a deployable resource that not only improve situational awareness but also offers a means of more effectively co-ordinating the inputs and activities of unbound volunteers. This in essence ensures more effective alignment between volunteer efforts and the wider recovery and reconstruction strategy.

This report serves as a narrative to the COBACORE data framework and constitutes a detailed overview of the development processes and the architecture implemented in the development of the COBACORE data framework. The report affords a comprehensive overview of the 'under the bonnet mechanics' operational within the data framework, including the ontologies and vocabularies developed, which in many ways have served to advance the application of computer science within the disaster recovery and reconstruction domain. The detailed overview of the semantic architecture and subsequent implementation changes required to optimize usability (in essence the trade-off between 'immediacy of response' and 'accuracy threshold') serve as a key learning outcome from the development process and afford a solid foundation for future technical development in this area. In this regard, a detailed narrative pertaining to the sense making features and weighting criteria presently applied within the confines of the matching algorithm to facilitate more efficient and effective needs and capacity 'matching' have been provided.

The key added value proposition of the COBACORE data framework is the ability to collate needs and capacities that arise across affected communities and apply matching of the needs and capacities in real-time, resulting in 'activities' which contribute to the reconstruction and recovery efforts. Following the successful integration of the data framework with the COBACORE platform, a series of in-house tests were conducted to determine operational capacity. The results obtained demonstrate the ability of the matching algorithm to effectively align potentially effective capacities to respond to an identified need, in terms of the responsible actors, descriptions, locations and availability, while assigning different degrees of importance to each of these criteria. Presently, such results are predicated on a pre-established prioritization between criteria. The multi-criteria decision making nature of the algorithm allows importance weights to be assigned by prospective end-users to prioritize criteria, depending on the needs arising in a specific disaster recovery and reconstruction scenario.

Insights and feedback received from the Final Evaluation have been utilised to further refine the data framework and improve the efficiency and operational capacities of COBACORE as a proof of concept and have initiated potential deployment opportunities within consortium networks in the Netherlands and the UK. Whilst the matching algorithm in itself does not extenuate the boundaries of technical SOTA, it is the application of matching and the testing of semantic functionality within the recovery and reconstruction domain which serves to advance knowledge and inform practice in both the field and the domain. The application of decision support systems has to date predominantly concentrated on the response phase of the disaster cycle. However, with professional resources continuing to be stretched, allied with the increased willingness to embrace unbound volunteers as a deployable resource, the COBACORE data framework serves as a technical solution upon which integration and co-ordination can be effectively operationalized.

7 References

Bender, M.A., Farach-Colton, M., Pemmasani, G., Skiena, S. and Sumazin, P. (2005), *Lowest common ancestors in trees and directed acyclic graphs*, Journal of Algorithms 57 (2) pp. 75-94.

Doumpos, M. and Grigoroudis, E. (2013), *Multicriteria Decision Aid and Artificial Intelligence*, Wiley Press, London.

Galway, L., Haran, M., Neef, M., and Woods, C. (2015), *Towards an Information Framework for Community Based Comprehensive Recovery*". Proceedings of the 12th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2015), 24-27 May 2015, Kristiansand, Norway.

Keeney, R. and Raiffa, H. (1993), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, Cambridge.

Liu, S., Brewster, C., & Shaw, D. (2013). *Ontologies for crisis management: a review of state of the art in ontology design and usability.* Proceedings of the 10th international conference on Information Systems for Crisis Response and Management (ISCRAM 2013), 12-15 May 2013, Baden-Baden (DE): Karlsruher Institut für Technologie, pp. 349-359.

Palomares, I., Galway, L., Haran, M., Neef, M., Woods, C. and Wang, H. (2015), *A Knowledge Management and Need-Capacity Matching Approach for Community-Based Disaster Management and Recovery*. Proceedings of the 10th International Conference on Intelligent Systems and Knowledge (ISKE 2015), Nov. 24-27, 2015, Taipei, Taiwan.

Palomares, I., Galway, L., Haran, M., Woods, C. and Meesters, M. (2016*), A Semantic Multi-Criteria Decision Support System for Community-based Disaster Response and Recovery*. Future Generation Computer Systems Journal; Special issue on Urgent Computing (Forthcoming), Elsevier, April 2016.

Porter, M. (1980), *An algorithm for suffix stripping.* Program 40(3), pp. 211–218.

Zadeh, L. (1965), *Fuzzy sets*. Information and Control 8(3), pp. 338–353.

Annex: Associated Publications

Galway, L., Haran, M., Neef, M., and Woods, C. (2015), *Towards an Information Framework for Community Based Comprehensive Recovery*". Proceedings of the 12th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2015), 24-27 May 2015, Kristiansand, Norway.

Palomares, I., Galway, L., Haran, M., Neef, M., Woods, C. and Wang, H. (2015), *A Knowledge Management and Need-Capacity Matching Approach for Community-Based Disaster Management and Recovery*. Proceedings of the 10th International Conference on Intelligent Systems and Knowledge (ISKE 2015), Nov. 24-27, 2015, Taipei, Taiwan.

Palomares, I., Galway, L., Haran, M., Woods, C. and Meesters, M. (2016*), A Semantic Multi-Criteria Decision Support System for Community-based Disaster Response and Recovery*. Future Generation Computer Systems Journal; Special issue on Urgent Computing (Forthcoming), Elsevier, April 2016.

Towards an Information Framework for Community Based Comprehensive Recovery

L. Galway
University of Ulster
l.galway@ulster.ac.uk

M. Neef
TNO
martijn.neef@tno.nl

C. Woods
Spatialest
conor@spatialest.com

M. Haran
University of Ulster
m.haran@ulster.ac.uk

ABSTRACT

The Community Based Comprehensive Recovery (COBACORE) platform is intended to facilitate enhanced communication flows between the professional responder community, the affected community, and volunteer responders in order to enhance situational awareness, inform and guide response planning, and ensure more effective co-ordination of the activities of the volunteer responder community. The mobilization, coordination and efficient capture of data from affected communities remain a largely untapped resource within crisis response. Subsequently, the information framework, designed and developed for the COBACORE platform, provides the ability to not only capture the needs and capacities across affected communities but also to improve subsequent activity generation through the application of semantic modeling to conduct inference-based matching of needs and capacities from the underlying communities. This paper provides a brief overview of the core concepts and associated ontology underpinning the semantic modeling facet of the COBACORE platform.

Keywords

Disaster Recovery, Information Model, Ontology, Sense Making

INTRODUCTION

Disaster management continues to attract significant attention by the research community, particularly the field of data management and analysis. In the immediate aftermath of a crisis, needs have to be identified and prioritized before the emergency actions are taken, so as to provide effective humanitarian assistance in a timely and coordinated manner. Decision makers are nonetheless often faced with the challenge of developing a response strategy predicated on a wide variety of data sources, which may be extremely heterogeneous, lack a common terminology, come in a plethora of formats, and are considered *uncertain* in nature (Hristidis, et al. 2010). Consequently, many of the datasets required to inform a crisis response are disparate and lack the levels of integration conducive to facilitating enhanced interoperability amongst the responder community and other stakeholders involved in emergency response planning. Moreover, there is a need for greater information sharing between professional and volunteer responders from within the affected community, in order to more effectively harness community capacity and ensure effective co-ordination of this vastly under-utilized resource base. Crisis incidents over the course of the last decade

highlight the increasingly important role of Information Communication Technology (ICT) systems in informing the response and recovery process. Advances in ICT have served to transform the way in which data and information is exchanged, thus enabling more effective harnessing of up to date *field data* from affected communities, which may subsequently bolster response co-ordination. The principal means of redressing the issue of interoperability and to facilitate enhanced data sharing is to utilize ontologies. In general, ontologies are used within crisis response in order to provide structured information, which creates meaningful relationships between information resources, and allows machines to automatically combine different sources of information into a consistent body of knowledge. Indeed, the multi-faceted and inter-disciplinary nature of crisis response is wholly conducive to semantic modeling in terms of developing relationships and linkages between relevant datasets, and enhancing the usability of the human-technical interface to facilitate commonality of understanding. However, while a number of crisis-orientated ontologies currently exist, they are not explicitly focused on the response and recovery stage of the disaster management cycle (Liu et al., 2013).

The COBACORE platform will utilize semantic modeling to provide an overall view of the data required for situational awareness and sense making within the confines of disaster recovery. Additionally, it will be further utilized in order to integrate existing databases using a common model and vocabulary. This paper presents the core ontology for the COBACORE project, including the core concepts underpinning the associated information model.

BACKGROUND

To date, a variety of domain-specific ontologies have been developed for disaster information management, which enable interoperability within specific scenarios. Moreover, efforts are already underway to enhance the capacity for data sharing and analytical processing by combing existing ontologies with geospatial semantic web technologies for emergency response and disaster management (Zhang et al., 2010). In spite of the progress that has been made in recent years, the utilization and application of crisis management ontologies largely remains relatively immature. Liu et al. (2013) conducted a review of state-of-the-art ontology design and usability for crisis management, which highlighted 26 existing ontologies that could be utilized within the confines of crisis management. Comprising a combination of ontologies created specifically for disaster management, along with ontologies containing relevant concepts from other domains, the highlighted ontologies were identified on the premise that crisis information systems are predicated on two groups of concepts involving 11 subject areas: common concepts (people, organizations, resources, disasters, geography, processes, infrastructure, damage) and unusual concepts (topography, hydrology and meteorology). However, out of the ontologies designed originally for crisis management, very few are formally represented and publicly accessible. Crisis oriented ontologies such as SoKNOS (Babitski et al., 2009), ISyCri (Benaben et al., 2008) and WB-OS (Chen-Huei et al., 2011) are formally represented but are not publicly available. In addition, the majority of the ontologies identified describe concepts within single subject areas, with only a few of the ontologies representing multiple subject areas including SoKNOS (resources, damage, disasters), MOAC (resources, processes, damage, disasters), SIADEX (resources, processes, geography), ISyCri (processes, damages, disasters), HXL (damage, geography, organization, disasters), and AktiveSA (geography, infrastructure, meteorology, processes, resources, organizations, people). While it is clear that several state-of-the-art ontologies exist for describing aspects of disaster response and management, they remain somewhat incomplete in terms of fully describing the disaster and recovery domain, thus provide scope for the development of ontologies for missing subject areas (Liu, et al., 2013). Consequently, the bespoke nature of the COBACORE platform will necessitate an expansion of these existing ontologies, as well as utilization of semantic modeling to interlink them.

COBACORE INFORMATION FRAMEWORK AND ONTOLOGY

Prior to creating the overarching information model and initial ontology for the COBACORE platform, a number of core concepts were identified based upon recognition of three conceptual data models deemed necessary in order to ensure optimal reconstruction and recovery; the Situation Model, which provides a contemporary picture of the socio-demographic, economic and physical environment; the Community Model, which maintains relevant information about the known actors within the affected area, surrounding area and in corresponding

online communities; the Needs Model, which facilitates an overview of actual needs on different levels of abstraction. From these models, a number of core concepts were subsequently derived: *Actor*, *Need*, *Capacity* and *Activity*. The *Actor* concept comprises the individuals and groups of individuals who are involved in the domain and can be said to have needs and capacities, and participate in activities. Within the COBACORE platform a group is formed by an *Actor* and can optionally be associated with one or more needs, capacities or activities. However, a group may not exist without inclusion of at least one *Actor*. Every individual will have their own needs, capacities and activities, which can be either taken individually or used to derive a common attribute for a group need, capacity or activity. Both individuals and groups are described as an *Actor*. The *Need* concept comprises the expression of a necessity by an *Actor*. In the COBACORE platform the definition of a *Need* is expanded to not only include essential requirements, such as assets, skills or information, but also crucial requirements, where the distinction is determined by ascribing an urgency attribute and corresponding duration to the *Need*. Subsequently, an *Actor* is responsible for registering a *Need* on the COBACORE platform, which subsequently initiates a lifecycle for the *Need*, thus permitting monitoring and traceability. Although considered as a core concept for the COBACORE platform, the concept of a *Need* is not a commonly described concept within existing crisis-orientated ontologies (Liu et al., 2013). Conversely, a *Capacity* describes a potential resource that could be used to influence the situation and help fulfill a *Need*. This not only includes the physical capability of an *Actor* but also their possible skill, informational and financial capabilities. Similar to a *Need*, a *Capacity* is registered by an *Actor* on the COBACORE platform and is specified along with a duration and sphere of influence that indicate availability in order to initiate and inform the lifecycle of the *Capacity*. In contrast, an *Activity* describes an action within the domain that may be a direct response to a *Need*, or simply an action that will have an influence on the overall situation. Similar to *Need* and *Capacity*, an *Activity* is registered on the COBACORE platform by an *Actor*; for example, a community champion of the responding community may create an *Activity* in order to address a *Need* and mobilize volunteers. Subsequently, an *Activity* is associated with one or more *Need* and, if available, one or more corresponding *Capacity*. If no *Capacity* is matched to a *Need*, the *Activity* may be utilized in order to identify and further assign a required *Capacity* once available. Each *Activity* must eventually be associated with an *Actor*, *Need* and *Capacity* and cannot exist without an *Actor*. Similar to *Need* and *Capacity*, an *Activity* that is registered on the COBACORE platform will be ascribed an urgency in order to inform its corresponding lifecycle. The overall concept of *Activity* aims to link an action with a stated *Need*, which permits the interaction between individuals and groups, and the outcome of a match between one or more *Need* and one or more *Capacity*, to be monitored.

The current iteration of the COBACORE ontology focuses primarily on the four core concepts *Actor*, *Need*, *Capacity* and *Activity* in order to provide a foundation for the future specification of a complete ontology for disaster recovery. Figure 1 provides an example of the Resource Description Framework Schema (RDFS) currently in place for the *Actor* concept utilized within the ontology, which indicates the relationships with the *Need*, *Capacity* and *Activity* concepts. The ontology itself defines a base class, *DomainConcept*, which comprises the primitive classes representing the four core concepts, along with and additional set of supporting classes including *Category*, *Type* and *Location*. The *Category* class contains a number of subclasses that represent the recovery domain associated with a *Need*, *Capacity* or *Activity*, i.e. *BuiltEnvironment*, *SocialAndCultural*, etc. Likewise, the *Type* class contains three subclasses, each with a set of additional subclasses, i.e. *Assets (Physical, Financial)*, *Information (Instruction, Expertise)*, and *Skills (Service, Labour)*. These supporting classes are utilized within the ontology to create a set of defined classes based on the primitive *Need*, *Capacity* and *Activity* classes, which used to conduct reasoning on instances of a *Need*, *Capacity* or *Activity*. For example, an instance of a *Need* that is specified as being of type *Expertise* can be inferred as belonging to the class *InformationBasedNeed*. By building up sets of defined classes in such a way, the ontology facilitates inference-based matching between instances of a *Need* and instances of a *Capacity*, thereby providing the COBACORE platform with a contextualized view of the requirements and capabilities of individuals and groups within the affected and responding communities.

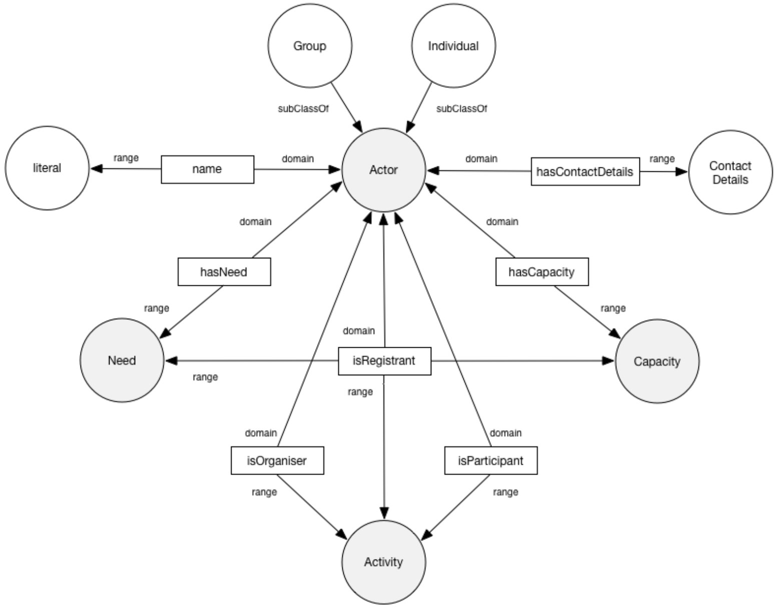


Figure 1. Resource Description Framework Schema for Actor Concept

Within the core ontology, an additional base class, *ValuePartition*, has also been defined in order to facilitate the separate specification of instances belonging to the enumerated classes, *Severity*, *Urgency* and *Status*. Such enumerated values are also utilized to determine defined classes for further inference on instances of the *Need*, *Capacity* and *Activity* classes.

SUMMARY & FUTURE WORK

As a step towards realizing a complete specification of the COBACORE information model, and corresponding ontology, four core concepts identified for the disaster response and recovery domain have been semantically modeled within an initial ontology. Expansion of the COBACORE ontology is currently ongoing, with future work focusing on the integration of existing crisis-orientated ontologies and vocabularies. Where missing ontologies or vocabularies are found, such as those supporting the *Need* concept, new entities will be defined using the 4W (Who, What, Where, When) approach advocated for modeling in the humanitarian domain. Through the eventual creation and release to the public domain of a complete ontology specific to disaster response and recovery, it is hoped that similar, future initiatives may freely avail and further build upon the COBACORE ontology.

REFERENCES

1. Babitski, G., Bergweiler, S., Hoffmann, J., Schön, D., Stasch, C. and Walkowski, A. (2009) Ontology-based Integration of SensorWeb Services in Disaster Management, *Lecture Notes in Computer Science Volume 5892*, 103-121, Springer, Berlin.
2. Bénaben, F., Hanachi, C., Lauras, M., Couget, P. and Chapurlat, V. (2008) A Metamodel and its Ontology to Guide Crisis Characterization and its Collaborative Management, *Proceedings of the 5th International ISCRAM Conference*, Washington DC, USA.
3. Chen-Huei, C., Fatemeh. Z. and Huimin, Z. (2011) Ontology for Developing Web Sites for Natural Disaster Management: Methodology and Implementation, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 41, 1, 50-62.
4. Hristidis, V., Chen, S-C., Li, T., Luis, S. and Deng, Y. (2010) Survey of data management and analysis in disaster situations. *Journal of Systems and Software*, 83, 10, 1701-1714.
5. Liu, S., Brewster, C. and Shaw, D. (2013) Ontologies for Crisis Management: A Review of State of the Art in Ontology Design and Usability, *Proceedings of the 10^{th} International ISCRAM Conference*, Baden-Baden, Germany.
6. Zhang, C., Zhao, T. and Weidong, L. (2010) Automatic search of geospatial features for disaster and emergency management, *International Journal of Applied Earth Observation and Geoinformation*, 12, 6, 409-418.

A Knowledge Management and Need-Capacity Matching Approach for Community-Based Disaster Management and Recovery

(Invited Paper)

Iván Palomares
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast
Belfast, United Kingdom
i.palomares@qub.ac.uk

Leo Galway
School of Computing and Mathematics
Ulster University
Newtownabbey, United Kingdom
l.galway@ulster.ac.uk

Martin Haran
School of Built Environment
Ulster University
Newtownabbey, United Kingdom
m.haran@ulster.ac.uk

Martijn Neef
Networked Organisations
TNO
The Hague, Netherlands
martijn.neef@tno.nl

Conor Woods
Spatialest
Coleraine, United Kingdom
conor@spatialest.com

Hui Wang
School of Computing and Mathematics
Ulster University
Newtownabbey, United Kingdom
h.wang@ulster.ac.uk

***Abstract*—Post-crisis response and recovery necessitates the identification and prioritization of the needs and capacities of the affected community in order to provide efficient and well-coordinated humanitarian assistance. The Community Based Comprehensive Recovery platform aims to facilitate enhanced communication flows between professional communities, affected communities, and volunteer responders to enhance situational awareness, inform and guide response planning, and ensure more effective coordination of activities by volunteer responders. Underpinning the platform, an information framework has been designed to support acquisition and analysis of the needs and capacities that arise across affected communities. In addition, a multi-criteria decision making algorithm has been designed and developed in order to enhance sense making and situational awareness within the platform. Subsequently, this paper introduces the core concepts that provide a basis for the information model, along with the associated ontology. Furthermore, the paper presents details of the decision making algorithm in conjunction with results from its application to a representative set of sample data.**

***Keywords*- Disaster Recovery; Information and Ontology Modeling; Multi-Criteria Decision Making**

I. Introduction

Disaster recovery is the process of turning damaged societies by diverse disasters back to a stable situation, such that their livelihood is regained, e.g. by reconstructing damaged physical and/or infrastructural objects. This process has been a significant focus of attention by researchers worldwide, particularly in the field of data management and analysis [1], [2]. In the immediate aftermath of a crisis, needs must be identified and prioritized before taking action, so as to provide efficient and well-coordinated humanitarian assistance. The Community Based Comprehensive Recovery (COBACORE[1]) platform for disaster recovery, aims at facilitating enhanced communication flows between professional communities, affected communities, and volunteer responders in order to enhance situational awareness, inform and guide response planning, as well as to ensure more effective coordination of the activities and decision making by volunteer responders [3]. In this sense, the information framework, a component designed and developed as part of COBACORE platform, provides the ability to not only capture the needs and capacities that arise across affected communities, but also to improve subsequent activity generation through the application of semantic modeling to conduct inference-based matching of needs and capacities from the underlying communities.

Nonetheless, the efficient capture and management of data from affected communities still remains a largely untapped resource within crisis response scenarios, since decision makers are often faced with the challenge of developing appropriate response strategies predicated on a vast array of data sources [4]. Such data sources may be extremely heterogeneous, lack a common terminology and level of integration, and they may come in a plethora of formats, thus being regarded as uncertain in nature [1]. Furthermore, there is an increasing need for information sharing between professional and volunteer responders within the affected community, in order to harness the available capacities more effectively.

Ontologies are a frequently utilized means of redressing inter-operability issues and facilitating enhanced data sharing [5], since they provide structured information and

[1] http://www.cobacore.eu/

meaningful relationships between information resources, thus allowing machines to automatically combine multiple sources of information into a consistent body of knowledge. However, despite there currently exist a number of crisis-orientated ontologies, they are not explicitly focused on the response and recovery stage of the disaster management cycle [4].

This paper focuses on two main features incorporated into the COBACORE platform information framework at its current development stage. Therefore, the paper proposal is twofold:

- Firstly, we present the core concepts underpinning the COBACORE information model and its associated ontology. The process to identify such concepts and the usefulness of the ontology to perform inference process on information, are described.
- We develop an approach for need-capacity matching, thus facilitating the task of identifying relevant capacities to respond to a need across communities. The approach calculates the matching degree between a set of candidate capacities and a target need, according to four evaluation criteria based on a 4W (Who, What, Where, When) approach. Thus, the proposed algorithm is modeled after a Multi-Criteria Decision Making (MCDM) framework [6].

This contribution is structured as follows: Section II reviews some background and concepts underpinning our proposal, namely a overview on existing ontologies for crisis management and aggregation operators for data fusion in MCDM. Section III presents the current version of the COBACORE ontology, while Section IV describes the multi-criteria need-capacity matching algorithm, highlighting its underlying approaches and techniques utilized to compare needs and capacities under several criteria, and illustrating the algorithm performance through an example. Finally, in Section V we conclude the paper and discuss ongoing work.

II. Background

This section includes an overview on existing crisis management ontologies, followed by a brief introduction to aggregation operators in MCDM.

A. Domain Ontologies for Crisis Management

To date, a variety of domain-specific ontologies have been developed for managing disaster information, in order to enable interoperability within several specific scenarios. There is already some ongoing effort to further enhance data sharing and analytical processing capabilities by combining these ontologies with geospatial semantic web technologies for emergency response and disaster management [2]. Despite the progress made in recent years, the utilization and application stage of crisis management ontologies is still relatively immature.

In [4], Liu et al. reviewed the ontology design and usability for crisis management state-of-the-art. As a result, they identified 26 existing ontologies to be utilized within this application domain, on the premise that crisis-oriented information systems are predicated on two groups of concepts involving 11 subject areas: (i) *Common concepts*: people, organizations, resources, disasters, geography, processes, infrastructure and damage; (ii) *Unusual concepts*: topography, hydrology and meteorology.

However, very few of these ontologies are formally represented and publicly accessible, and most of them describe concepts within single subject areas, tackling only few of them the representation of multiple subject areas from the ones listed above. Additionally, existing state-of-the-art ontologies for describing aspects of disaster response and management, still remain somewhat incomplete in terms of fully describing the disaster recovery domain. Therefore, the bespoke nature of the COBACORE platform necessitates an expansion of these existing ontologies (as will be shown in Section III), as well as the use of semantic modeling to interlink them.

B. Aggregation Operators for Multi-Criteria Decision Making (MCDM)

Decision making is a frequent process in human lives, in which there exist several alternatives and the best potential alternative will ideally be chosen. MCDM refers to a family of methods to deal with decision making problems under the presence of several criteria, so that alternatives are evaluated according to each criterion [6].

Let us consider a MCDM framework formulated as follows:

- There exists a decision problem consisting of $z \geq 2$ alternatives or possible solutions, $X = \{x_1, \ldots, x_z\}$.
- There exist several independent criteria, $Q = \{q_1, \ldots, q_n\}$, $n \geq 2$, under which alternatives must be assessed.

$m_{jk} \in D$ represents the degree to which alternative $x_j \in X$ satisfies criterion $q_k \in Q$, expressed in a information domain D, e.g. the [0,1] interval (according to which $m_{jk} = 1$ indicates that x_j completely satisfies q_k, whereas $m_{jk} = 0$ indicates null satisfaction of such criterion by x_j). In order to obtain an overall assessment value m_j for x_j, it would be necessary to combine its satisfaction degrees over the different criteria. A common approach to do this in MCDM is by using aggregation operators.

An aggregation operator in the unit interval [7] is defined by a function $f : [0,1]^n \to [0,1]$, $n > 1$, with the purpose of combining n input values or arguments $a_1, \ldots, a_n$ into an output value. The following properties are fulfilled by any aggregation operator f in [0,1]:

(i) $f(0, \ldots, 0) = 0$ and $f(1, \ldots, 1)$ (idempotent).
(ii) $a_k \leq b_k$ $\forall k = 1, \ldots, n$ implies $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$ (monotonicity).

In a MCDM framework, aggregation operators are commonly used to compute the overall satisfaction degree of each alternative as:

$$m_j = f(m_{j1}, \ldots, m_{jn}) \tag{1}$$

Some examples of aggregation operators classically used in MCDM approaches are:

- Arithmetic mean:
$$f(a_1, \ldots, a_n) = \frac{1}{n}(a_1 + a_2 + \ldots a_n) \tag{2}$$
- Geometric mean:
$$f(a_1, \ldots, a_n) = \sqrt[n]{a_1 a_2 a_n} \tag{3}$$
- Weighted average:
$$f(a_1, \ldots, a_n) = \frac{\sum_{k=1}^{n} w_k a_k}{\sum_{k=1}^{n} w_k} \tag{4}$$
where $w_1, \ldots, w_n \in [0,1]$ are importance weights assigned to criteria, such that the higher w_k, the more importance a_k plays in the aggregation process.

The three operators shown above are averaging operators, in which $\min(a_1, \ldots, a_n) \leq f(a_1, \ldots, a_n) \leq \max(a_1, \ldots, a_n)$.

Once an overall assessment or satisfaction degree m_j for each alternative is obtained, it can be used to determine and select the best alternative as the solution for the problem, or to rank several alternatives in decreasing order of such a degree, depending on the nature of the decision problem. The proposed matching algorithm in this work aims at obtaining a ranking over a subset of alternatives (candidate capacities for a need), and it deems some criteria as more important than others (i.e. weights must be assigned to individual criteria), as described in Section IV.

III. The COBACORE Information Model and Ontology

In this section, the core ontology for the COBACORE project [3] and the core concepts underpinning the corresponding information model are presented.

Prior to creating the initial ontology and information model, a number of concepts were identified based on the recognition of three conceptual data models deemed necessary to guarantee an optimal recovery and reconstruction domain:

- *Situation Model*: provides a contemporary picture of the socio-demographic, economic and physical environment.
- *Community Model*: maintains relevant information about the known actors within the affected area, surrounding areas and in corresponding online communities.
- *Needs Model*: facilitates an overview of actual needs on different abstraction levels.

From these three models, the following core concepts were derived: *Actor*, *Need*, *Capacity* and *Activity*:

- *Actor*: This concept comprises those individuals and groups who are involved in the domain and can be claimed to have needs and/or capacities, as well as participate in activities. A group in the COBACORE platform is formed by actors and can optionally have associated one or more needs, capacities or activities. The existence of a group implies the inclusion of at least one actor in it. Every individual will also have their own needs, capacities and activities.
- *Need*: It comprises the expression of a necessity by an actor. Its definition is expanded to include not only essential requirements (e.g. assets, skills or information) but also crucial requirements. The distinction in this sense is determined by an urgency level attribute with corresponding duration to fulfill the need.
- *Capacity*: It describes potential resources that could be used to help fulfill a need. This concept not only includes the physical capability of an actor but also their possible skills, informational and financial capabilities (hence some specific capabilities might require physical intervention at a specific place, for instance). In line with needs' urgency level, capacities have associated a duration and sphere of influence indicated by an availability level.
- *Activity*: This concept describes an action within the domain that may be a direct response to a need. For example, an actor representing the community leader of a responding community may create an activity in order to address a need and mobilize volunteers. Subsequently, this target activity is associated with one or more needs and, if available, one or more matching capacities. If no capacities are matched to a need, the activity may be utilized to identify and further assign a capacity once available. Therefore, any activity must eventually be associated with an actor, need and capacity, and cannot exist without an actor. Similarly to needs/capacities, activities are also ascribed an urgency level.

An actor is responsible for registering needs, capacities and/or activities on the COBACORE platform, thus initiating a lifecycle for all of them to permit their monitoring and traceability. Furthermore, the overall concept of activity aims at permitting monitoring the outcome of a match between one or more needs and one or more capacities (further detail on the methodology for matching needs and capacities is given in Section IV).

The current version of the ontology focuses primarily on the four core concepts introduced above, thus providing a foundation for future specifications of a complete ontology for disaster recovery. Figure 1 shows an example of the Resource Description Framework Schema (RDFS) currently in

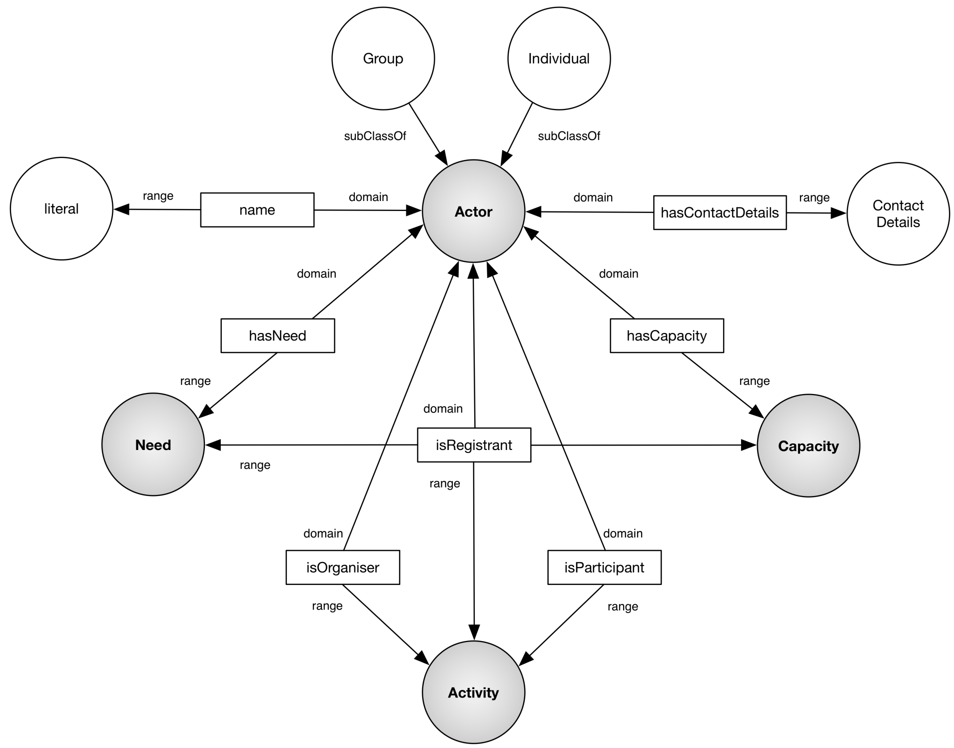


Figure 1. Resource Framework Description Schema for *Actor* Concept

place for the actor concept, depicting the relationships with the need, capacity and activity concepts. A base class in the ontology, called *DomainConcept*, comprises the primitive classes representing the four core concepts along with the following additional supporting classes:

- *Category*: contains a number of subclasses representing the disaster recovery domain associated with a need, capacity or activity, e.g. *Built Environment*, *Social and Cultural*, etc.
- *Type*: contains three subclasses, each one containing in turn additional and more-specific subclasses, e.g. *Assets (Physical, Financial)*, *Information (Instruction, Expertise)* and *Skills(Service, Labour)*.

These supporting classes are utilized to create a set of defined classes upon the primitive classes (i.e. need, capacity, and activity), for instance *SocialNeed*, *ServiceSkillsCapacity*, etc. This information can be potentially used to conduct reasoning processes on instances of primitive classes and infer additional, implicit information about them. For example, an instance of a need that was specified as being of type *Expertise* can be inferred as belonging to the class *InformationNeed*. This way, the ontology may facilitate inference-based matching between instances of a need and instances of a capacity, thereby providing a contextualized view of the requirements and capabilities of individuals and groups within the affected and responding communities. For example, candidate capacities inferred as *InformationCapacity* could be regarded as potentially highly-matching for an *InformationNeed*.

IV. Need-Capacity Matching Algorithm in COBACORE Information Framework

This section presents the need-capacity matching algorithm for the COBACORE information framework. The algorithm applies a multi-criteria aggregation of partial matching degrees $m_k(n_i, c_j) \in [0,1]$ between a target need $n_i \in \{n_1, \ldots, n_m\}$ and a candidate capacity $c_j \in \{c_1, \ldots, c_z\}$, predicated on the 4W (Who, What, Where, When) approach. Then, candidate capacities are ranked and a subset of the most highly-matching ones are returned.

Remark 1: Although the proposed algorithm currently utilized information introduced by an actor in the COBACORE platform exclusively, ongoing work (Section V) focuses on extending it, by incorporating a semantic inferencing phase to provide enhanced filtering of capacities based on inferred categories/types.

Figure 2 depicts the scheme of the proposed matching algorithm, whose steps are further described below:

A. Computing Partial Matching Degrees

The following procedures are followed to compute partial matching degrees relating to each of the 4W criteria.

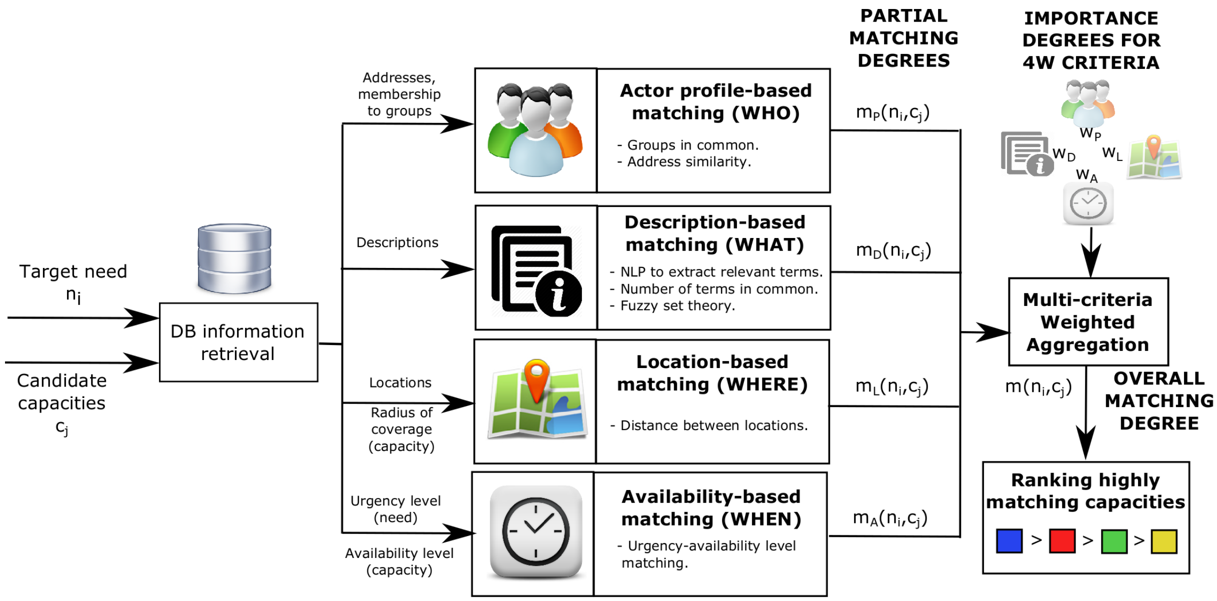


Figure 2. Scheme of the need-capacity matching algorithm

1) Profile-based Matching (WHO): This matching degree is computed by taking into consideration the information in the user profile of the two involved actors, namely their addresses and the common groups to which both actors belong (if any).

Remark 2: An actor address (given by non-spatial data) normally refers to his/her current home or work address, which does not necessarily correspond to the location (spatial data, see location-base matching above) where such an actor registered a need or capacity (see Section IV-A3).

An address is decomposed into five geographical levels: $l \in \{1\ (country), 2\ (county), 3\ (state), 4\ (city), 5\ (street)\}$. Let $addr_{ij}^l \in \{0,1\}$ represent the coincidence between the addresses of the origin actors in resp. n_i and c_j, at level l. For instance, for two actors whose addresses are in two different streets within the same city, we have $addr_{ij}^1 = addr_{ij}^2 = addr_{ij}^3 = addr_{ij}^4 = 1$, and $addr_{ij}^5 = 0$.

On the other hand, since each actor may optionally belong to one or more groups, we define a coefficient δ_{ij} based on the number of groups, $\#gr_{ij} \in \mathbb{N} \bigcup \{0\}$, to which both actors belong simultaneously:

$$\delta_{ij} = \begin{cases} 0.1 & \text{if } \#gr_{ij} = 0, \\ 0.15 & \text{if } \#gr_{ij} = 1, \\ 0.2 & \text{if } \#gr_{ij} \geq 2. \end{cases} \tag{5}$$

Based on $addr_{ij}^l$ and δ_{ij} we propose to compute a profile-based matching degree as follows:

$$m_P(n_i, c_j) = \delta_{ij} \sum_{l=1}^{5} addr_{ij}^l \tag{6}$$

Notice that, δ_{ij} acts as a factor to increase the overall profile matching when both actors have groups in common.

2) Description-based Matching (WHAT): Descriptions of needs or capacities consist of free text introduced by users. The proposed procedure to estimate how similar two descriptions are, aims at finding common keywords between them. Such a procedure applies some simple Natural Language Processing (NLP) techniques and fuzzy set theory on descriptions, and it consists of the following steps:

i) *Extracting relevant keywords*: A POS (Part Of Speech) tagging algorithm is used to tag each word in the description with its corresponding POS (e.g. definite article, noun, preposition, etc.). Based on this tagging, words belonging to the least relevant categories in terms of the overall description meaning (such as articles and prepositions), are left out.

ii) *Stemming*: An approach based on Porter stemming algorithm [8], is applied on relevant words to determine their lemma, i.e. their basic form within the corresponding word-family. This may significantly increase the detection of similar words between descriptions in the cases that they belong to the same word family (e.g. “emergency - emergencies”, “reforest - reforesting”, etc.).

iii) *Determining Fuzzy Matching Degree*: Let $\#T_i$ and $\#T_j$ be the number of relevant (stemmed) keywords in the descriptions of n_i and c_j, respectively. The computation of description-based matching degree consists in determining the degree to which the number of common keywords amongst them, $\#T_{ij}$, is enough to consider both descriptions as similar. Since the

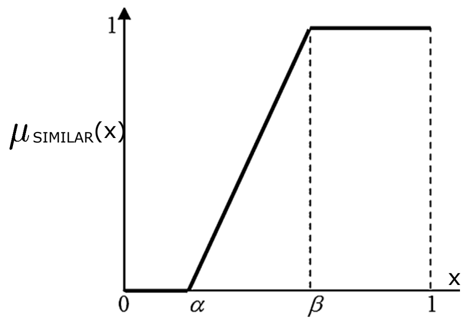


Figure 3. Fuzzy membership function for "SIMILAR"

concept "similar" has a vague and imprecise nature in this scenario, we use fuzzy set theory to model such a concept [9]. To do this, we define a fuzzy set $SIMILAR$ in the [0,1] interval, with semi-trapezoidal membership function (see Figure 3):

$$\mu_{SIMILAR}(x) = \begin{cases} 0 & \text{if } x \leq \alpha, \\ \dfrac{x-\alpha}{\beta-\alpha} & \text{if } \alpha < x < \beta, \\ 1 & \text{if } x \geq \beta,. \end{cases} \quad (7)$$

where $\mu_{SIMILAR}(x) \in [0,1]$ indicates the degree to which a value $x \in [0,1]$ (in our case, the number of common words between two descriptions) complies with the notion of $SIMILAR$.

The values of the two parameters $\alpha, \beta \in [0,1]$ that define the fuzzy membership function allow to flexibly be more or less tolerant with the notion of similarity between descriptions. The description-based matching degree is then computed as:

$$m_D(n_i, c_j) = \mu_{SIMILAR}\left(\frac{\#T_{ij}}{\min(\#T_i, \#T_j)}\right) \quad (8)$$

3) Location-based Matching (WHERE): In this step, a matching degree $m_L(n_i, c_j) \in [0,1]$ is computed based on the distance between the geographical locations (spatial data) of a need n_i and a capacity c_j. An important aspect to consider here is that some capacities might not require physical intervention, hence they can be implemented regardless of the location of the need (e.g. providing informational resources or sending money via bank transfer). On the other hand, capacities requiring physical intervention normally have an associated radius of coverage $r_j \in \mathbb{R}^+$ (expressed in kilometers). Hence, a capacity with no specified value for r_j is regarded as not requiring physical intervention.

Thus, $m_L(n_i, c_j)$ is computed as follows:

$$m_L(n_i, c_j) = \begin{cases} \max\left(0, \dfrac{r_j - d(n_i, c_j)}{r_j + d(n_i, c_j)}\right) & \text{if } \exists r_j > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (9)$$

with $d(n_i, c_j)$ being the distance between the locations of n_i and c_j. This matching measure shows that:

- If $d(n_i, c_j) < r_j$, then n_i is within c_j's coverage radius, therefore $m_L(n_i, c_j) > 0$.
- If $d(n_i, c_j) > r_j$, then n_i is outside c_j's coverage radius, therefore $m_L(n_i, c_j) = 0$.
- If $\nexists r_j > 0$, then c_j does not require physical intervention and can be effectively implemented regardless of the geographical location of n_i, hence we assign $m_L(n_i, c_j) = 1$ in this case.

4) Availabity-based Matching (WHEN): The computation of this matching degree considers two aspects. On the one hand, the urgency level specified for a need n_i indicates the maximum time elapsed to deal with it. It is denoted by urg_i and it takes a value within a numerical scale from 1 to 3, such that $urg_i \in \{1 : LOW\ (7+\ days), 2 : MED\ (1-7\ days), 3 : HIGH\ (< 24\ h)\}$. On the other hand, capacities c_j have associated an availability level $avail_j \in \{1, 2, \ldots, 10\}$, such that the higher $avail_j$, the more availability to respond to urgent needs.

Under the premise that highly urgent needs should be matched more exclusively to the most available capacities to avoid filtering unsuitable results to the end user, we propose computing the urgency-based matching degree $m_A(n_i, c_j)$ as follows:

$$m_A(n_i, c_j) = \left(\frac{avail_j}{10}\right)^{urg_i} \quad (10)$$

By raising to the urgency level urg_i, we have that the higher its value, the more penalizing is applied to the matching degree with low-availability capacities, hence they are more drastically discriminated from highly available (and hence more suitable) capacities in these cases.

B. Multi-Criteria Weighted Aggregation and Ranking

The last step of the algorithm consists in aggregating the 4W-based partial matching degrees to obtain the overall matching degree $m(n_i, c_j)$ between n_i and each candidate capacity c_j. Inspired by aggregation approaches in MCDM (see Section II-B), we use an aggregation operator to obtain such overall matching degrees. Since a requirement in the COBACORE platform is to assign different degrees of importance to the 4W criteria, we utilize a weighted average operator to compute $m(n_i, c_j)$:

$$m(n_i, c_j) = \frac{\sum_{k=1}^{4} w_k m_k(n_i, c_j)}{\sum_{k=1}^{4} w_k} \quad (11)$$

where $k \in \{P : Who, D : What, L : Where, A : When\}$ More specifically, the following priority order between criteria is considered: $Who \succ What \succ Where \sim When$, therefore the weighting vector $W = [w_P, w_D, w_L, w_A]$ must accomplish $1 \geq w_P > w_D > w_L = w_A > 0$.

Table I
INFORMATION OF TARGET NEED n AND CAPACITIES c_1 TO c_6

	WHO	WHAT	WHERE	WHEN
n	UK, -, -, Belfast, Orby Drive $\{g_1, g_2, g_3\}$	"There is no **electricity** in my first aid camp in South **Belfast**. Does anybody provide a **generator** to put there?"	-	2/3
c_1	UK, -, -,Belfast, - $\{g_3, g_4\}$	"I am a qualified tree surgeon. I have the equipment to remove large fallen trees from floods or electrical storms."	5	6/10
c_2	UK, -, -, Belfast, - $\{g_2\}$	"I give a table and six chairs as well as an old sofa which I no longer need."	2	9/10
c_3	UK, Antrim, -, Belfast, Belmont Road $\{g_1, g_2, g_5, g_6\}$	"Telephone assistance against emergencies to all Belfast neighbors"	-	6/10
c_4	UK, Antrim, -, Belfast, Belmont Road $\{g_1, g_2, g_5, g_6\}$	"I can share on-line resources and advice against any kind of first aid events."	-	8/10
c_5	UK, -, -, Belfast, Eia Street $\{g_1, g_5\}$	"I offer **electricity generators** and other blackout resources in **Belfast**"	15	7/10
c_6	UK, -, -, Belfast, - $\{\}$	"I have carpentry skills and can help with general manual labour needs"	20	6/10

Finally, given a fixed value $\gamma \in \mathbb{N}$ representing the number of capacities to be returned as a result of the matching algorithm, the candidate capacities are ranked by decreasing order of the matching degree, and the top-γ capacities in the ranked list are returned.

C. Illustrative Example

In this subsection we demonstrate the operation of the need-capacity matching algorithm with an illustrative example. Consider an input need n, and a set of six candidate capacities[2] whose information is summarized in Table I:

- WHO: The actor profile information, including: (i) the actor address details, in the format *<Country>,<County>,<State>,<City>,<Street>*, where an unspecified optional value is denoted by "-"; and (ii) the set of groups g_i to which the actor belongs, if any.
- $WHAT$: The full description of the need or capacity, as introduced by the actor in the system.
- $WHERE$: An integer number representing the coverage radius (in kilometers) for a capacity, or "-" for capacities that do not require physical intervention. The spatial information regarding the location of the need or capacity is not shown here for clarity reasons, but in this example all of them are located within the Belfast metropolitan area.
- $WHEN$: A value that indicates the urgency level in a scale [1-3] in the case of the needs, and the availability level in a scale [1-10], in the case of capacities. The higher this value, the higher the degree of urgency (resp. availability).

[2]Although the number of existing capacities in the COBACORE platform is significantly larger, in this example we only consider six capacities for space and clarity reasons. In a future version of the information framework, this set of candidate capacities would be the result of firstly applying a semantic inference-based matching for filtering out from an initially large set of capacities (as discussed in Section V).

Table II
COMPUTATION OF MATCHING DEGREES

c_j	m_P	m_D	m_L	m_A	$m(n, c_j)$
c_1	0.6	0.192	0.289	0.36	**0.389**
c_2	0.6	0	0.03	0.81	**0.371**
c_3	0.8	0.625	1	0.36	**0.708**
c_4	0.8	0.625	1	0.64	**0.759**
c_5	0.6	1	0.447	0.49	**0.664**
c_6	0.4	0	0.679	0.36	**0.334**

The parameters for the matching algorithm are:

- Fuzzy membership function parameters for description-based matching: $\alpha = 0$ and $\beta = 0.4$.
- Partial matching degrees are aggregated by using a weighted average operator, with the following weights for the $4W$ criteria considered: $w_P = 1.0$, $w_D = 0.75$, $w_L = 0.5$ and $w_A = 0.5$.
- Number of top-matching capacities returned, $\gamma = 3$.

The results of computing the partial and overall matching degrees between n and the six capacities, are shown in Table II (the notation for partial matching is abbreviated). From these results, the following ranking of capacities is derived:

$$c_4 \succ c_3 \succ c_5 \succ c_1 \succ c_2 \succ c_6$$

Therefore, given $\gamma = 3$, the resulting list of matching capacities with n to be returned, are $\{c_4, c_3, c_5\}$. Notice that, even though c_5 might seem the most highly matching capacity from its description, its location-based matching is much lower than that of c_3 and c_4, both of which do not require any physical intervention. In order to illustrate the computation of matching degrees in further detail, we show their computation for c_5:

- $m_P(n, c_5)$: It is computed, given $\sum_{l=1}^{5} addr_{ij}^{k} = 4$ (both actors are in the same city) and $\delta_{ij} = 1$ (the actors have

one group in common, g_1), as:

$$m_A(n,c_5) = 0.15 \cdot 4 = 0.6$$

- $m_D(n, c_5)$: There exist three common terms amongst the descriptions of n and c_5 (highlighted in bold face in Table I). Consequently, the resulting description-based matching degree is:

$$m_D(n,c_5) = \mu_{SIMILAR}\left(\frac{3}{7}\right) = 1$$

- $m_L(n, c_5)$: Since the capacity requires physical intervention and it is offered within a radius of 15 kilometers, the location-based matching degree is calculated as:

$$m_L(n,c_5) = \max\left(0, \frac{15 - 5.73}{15 + 5.73}\right) = 0.447.$$

- $m_A(n, c_5)$: Given the urgency and availability levels, this matching degree can be easily obtained as follows:

$$m_A(n,c_5) = \left(\frac{7}{10}\right)^2 = 0.49$$

Finally, the overall matching degree for $m(n,c_5)$ is computed as:

$$\frac{1 \cdot 0.6 + 0.75 \cdot 0.577 + 0.5 \cdot 0.447 + 0.5 \cdot 0.49}{1 + 0.75 + 0.5 + 0.5} = 0.546$$

V. Summary and Future Work

Within the domain of crisis response and recovery, there is a need for improved situational awareness in order to fully harness the capabilities and capacities within the affected community. Furthermore, to achieve such improvements, platforms that facilitate enhanced sense making, particularly in terms of improved communication and organization of the affected community, will potentially benefit the overall response and recovery efforts.

This contribution has introduced the COBACORE platform, with a focus on: (i) its core concepts, information model and associated ontology for disaster recovery communities, and (ii) a multi-criteria need-capacity matching algorithm that underpins aspects of the situational awareness capabilities of the platform. In particular, the details of the 4W approach to the matching algorithm have been detailed.

We have illustrated the use of the matching algorithm with an example that shows its usefulness in disaster recovery scenarios. Results obtained from use of the application on an example set of data, indicate that the proposed matching algorithm can align potentially effective capacities to respond to a demanded need, in terms of their responsible actors, descriptions, locations and availability, while assigning different degrees of importance to each of these criteria. Although the results shown are predicated on a pre-established prioritization between criteria, we remark that the advantage of modeling the algorithm as a multi-criteria decision making framework is its capacity to flexibly utilize different importance weights to prioritize criteria, as well as choosing different aggregation operators, depending on the needs arising at each specific scenario.

While the paper has discussed the four core concepts of the information model, which are utilized by both the ontology and matching algorithm, expansion of the ontology is currently ongoing. Future work will concentrate on the integration of inference-based categorization of needs and capacities, employing the resulting ontology, in order to provide optimizations for the matching algorithm. This process will be conceived as a previous phase to the multi-criteria matching process, such that a first filtering of relevant capacities is made from amongst a considerably large data-set of capacities introduced by communities upon the COBACORE platform deployment.

Acknowledgment

The work presented in this publication stems from the COBACORE project (Community-based Comprehensive Recovery, `www.cobacore.eu`) and is funded by the European Union FP7 Security Programme under grant agreement number 313308.

References

[1] V. Hristidis, S. Chen, T. Li, S. Luis, and Y. Deng, "Survey of data management and analysis in disaster situations," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1701–1714, 2010.

[2] C. Zhang, T. Zhao, and L. Weidong, "Automatic search of geospatial features for disaster and emergency management," *International Journal of Applied Earth Observation nand Geoinformation*, vol. 12, no. 6, pp. 409–418, 2010.

[3] M. Neef, K. van Dongen, and M. Rijken, "Community-based comprehensive recovery: Closing collaboration gaps in urban disaster recovery," *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2013)*, pp. 546–550, 2013.

[4] S. Liu, C. Brewster, and D. Shaw, "Ontologies for crisis management: A review of state of the art in ontology design and usability," *Proceedings of the 10th International ISCRAM Conference*, 2013.

[5] S. Staab and R. Studer, *Handbook on Ontologies (International Handbooks on Information Systems)*. Springer, 2009.

[6] J. Lu, G. Zhang, D. Ruan, and F. Wu, *Multi-Objective Group Decision Making*. Imperial College Press, 2006.

[7] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Springer, 2007.

[8] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 40, no. 3, pp. 211–218, 1980.

[9] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

*Manuscript with source files (Word document)
Click here to view linked References

A Semantic Multi-Criteria Decision Support System for Community-based Disaster Response and Recovery

I. Palomares[a], L. Galway[b], M. Haran[c], C. Woods[d], K. Meesters[e]

[a]*School of Electronic, Electrical Engineering and Computer Science. Queen's University Belfast. United Kingdom*
[b]*School of Computing and Mathematics. Ulster University. United Kingdom*
[c]*School of the Built Environment. Ulster University. United Kingdom*
[d]*Spatialest. Coleraine. United Kingdom*
[e]*School of Economics and Management. Tilburg University. Netherlands*

Abstract

Disaster response and recovery processes normally demand time-sensitive action implementation and decision making. A central aspect to consider within these contexts is the necessity of rapid and effective identification, analysis and alignment of the needs demanded by (members of) affected communities with resources offered by responding (members of) communities. The fulfillment of these requirements is one of the core goals of the web-based COBACORE platform for community-based disaster recovery, which aims at providing end-users with a comprehensive decision support tool. This paper presents a two-stage need-capacity matching algorithm, implemented within the COBACORE platform information framework, to assist affected users in the obtention and selection of relevant response resources (*capacities*) that match their needs. The algorithm firstly applies a category-based semantic matching stage, predicated on a domain-specific information model and ontology, to categorize needs and capacities, and filter strongly matched capacities in terms of the underlying categories to which they belong. A multi-criteria matching and aggregation stage is subsequently applied on the filtered capacities in order to rank them based on four predefined criteria, and to select the most suitable capacities as the set of results that are returned to the end user. An experimental study is finally conducted to show the usefulness of the proposed two-stage matching approach in the provision of thoroughly filtered and relevant results to affected users.

Keywords: Algorithmic Matching, Multi-Criteria Decision Making, Disaster Response and Recovery, Time-sensitive Decision Support, Aggregation Functions

1. Introduction

Disaster response and recovery is the process of returning communities uprooted by the occurrance of a disaster back to a more normal, stable situation, e.g. by reconstructing damaged physical and/or infrastructural objects. This subject has attained significant attention by researchers worldwide, particularly in the field of data management and analysis [1, 2], due to the need for improved situational awareness in order to fully harness the capabilities and capacities inherent within affected communities. Furthermore, in the immediate aftermath of a crisis, needs must be identified and prioritized before taking action, so as to provide efficient and well-coordinated humanitarian assistance. Consequently, in order to meet these requirements, platforms that facilitate enhanced sense making and decision support, with a view to improving communication and organization of the affected community, will potentially benefit the overall response and recovery efforts.

The COBACORE[1] (Community Based Comprehensive Recovery) platform is a Web-based platform aimed at supporting common needs assessment and facilitating planning efforts for recovery in complex multi-sectorial, multi-stakeholder crisis environments by building upon the community as an important source of both information and capacity. The platform facilitates enhanced communication flows between professional communities, affected communities, and volunteer responders in order to enhance situational awareness, inform and guide response planning, as well as to ensure more effective activity coordination and, most importantly, time-sensitive decision making by volunteer responders [3]. The underlying Information Framework is a component developed as part of the platform that provides the ability to capture the needs and capacities that arise across affected and responding communities, respectively. Importantly, the identification of needs and capacities is aimed at improving subsequent activity generation in order to adequately respond to the myriad diversity of needs typically inherent in crisis management and disaster recovery.

Nonetheless, the efficient capture and management of data from affected communities still remains a largely untapped resource within crisis response scenarios, since decision makers must often face the challenge of developing appropriate response strategies predicated on a vast array of extremely heterogeneous data sources [4]. Such data sources may also lack a common terminology and level of integration, and potentially come in a plethora of formats, thus may be regarded

[1] http://www.cobacore.eu/

as uncertain in nature [1]. Ontologies are a frequently utilized means of redressing interoperability issues and facilitating enhanced data sharing [5], since they provide structured information and meaningful relationships between information resources, thus allowing machines to automatically combine multiple sources of information into a consistent body of knowledge. There currently exist a number of crisis-orientated ontologies, but unfortunately they are not explicitly focused on the response and recovery stage of the disaster management cycle [4].

Furthermore, there is an increasing need for information sharing between professional and volunteer responders within the affected community, in order to harness the available capacities more effectively and ensure the available information is not insufficient. Consequently, the COBACORE platform aims at bridging this gap by facilitating collaboration across communities and making information readily shareable amongst its members, thus harnessing the available capacities more effectively. However, this level of information sharing brings with it a tradeoff regarding the size and complexity of available information that users in affected communities must deal with. In particular, users of the platform may generate an overwhelming repository of resources, many of which could result in potentially incompatible work practices or lead to misaligned decision making processes. In order to improve situational awareness and serve as a robust and credible tool to guide resource deployment and allocation, we propose a methodology to align needs with suitable capacities, by: (i) the application of semantic modeling to conduct inference-based matching of needs and capacities from the underlying communities, and (ii) the use of multi-criteria decision making approaches [6] for evaluating candidate capacities according to multiple criteria and combining these evaluations into overall ratings. As a result of this two-stage matching process, users in affected communities can easily access relevant, customized information on potentially beneficial capacities to meet their needs.

Based on the challenges outlined above, this paper focuses on an essential feature of the Information Framework of the COBACORE platform, by presenting an approach for need-capacity matching that will facilitate the task of identifying relevant capacities in response to an identified need across communities in high-stake environments where accessing relevant information efficiently becomes vital. The proposed need-capacity matching algorithm for disaster response and recovery is composed by the following stages:

1. Firstly, it applies semantic inference matching on all available capacities registered on the platform, predicated on the COBACORE underlying information model and ontology. As a result, capacities are both implicitly

and explicitly categorized, and those capacities falling under compatible categories with those for the target need are returned as candidate capacities.

2. Secondly, the algorithm calculates a matching degree between each of the candidate capacities and the target need, according to four evaluation criteria based on a 4W (Who, What, Where, When) approach. Information related to each of these criteria have a different nature within the underlying database associated with the platform, therefore we consider the use of different approaches to calculate each of the four matching degrees accordingly.
3. Finally, candidate capacities are ranked according to their overall (*aggregated*) matching degree with the need, and the highest matching ones are returned to the end-user.

Thus, the proposed methodology can be largely modeled after a Multi-Criteria Decision Making (MCDM) framework [6, 7].

This paper is structured as follows: Section 2 reviews some background and theoretical concepts underpinning the work presented herein, namely an overview of MCDM and aggregation functions. Section 3 presents the COBACORE Information Framework and ontology, while Section 4 describes the proposed two-stage need-capacity matching algorithm, highlighting the different approaches utilized to apply semantic filtering of candidate capacities and assess them based on several criteria. An application example and experimental evaluation that illustrates the capabilities of the matching algorithm in practice is presented in Section 5. Finally, in Section 6 we conclude the paper and discuss ongoing work.

2. Preliminaries

This section provides a concise overview of Multi-Criteria Decision Making, followed by some basic concepts and examples of aggregation functions and their use within the application context considered.

2.1. Overview of Multi-Criteria Decision Making (MCDM)

Decision making is a frequent process in human lives, in which there exists several alternatives and the best ones will ideally be chosen. MCDM refers to a family of methods that are utilized to deal with decision making problems under the presence of several, often conflicting, criteria. The major goal of MCDM approaches originally consists of providing criteria aggregation methodologies

in order to enable effective decision support [6]. Four main methodological approaches can be distinguished for the resolution of MCDM problems depending on their nature [8]:

(i) *Multi-Attribute Utility Theory (MAUT)* [9]: Extends classic utility theory to deal with multiple criteria, and is regarded as the cornerstone for developing MCDM methodologies, consisting in representing preferences as value or utility functions.

(ii) *Multi-Objective Mathematical Programming* [10]: An extension of mathematical programming to problems involving multiple objectives in conflict with each other. Goal programming formulations have been utilized as an alternative approach to model this kind of problem.

(iii) *Outranking Methods* [11]: The development of the ELECTRE models led to this family of techniques, characterized by the construction and subsequent exploitation of an outranking relation, i.e. a not necessarily transitive nor complete relation indicating preference, indifference or incomparability between pairs of alternatives. Outranking methods can be applied for selecting, ranking or classifying alternatives.

(iv) *Preference Disaggregation Analysis* [12]: These methods were conceived for situations where direct MCDM approaches may not be applicable, due to a large amount of information and/or the presence of time restrictions. Disaggregation methods aim at inferring and calibrating a suitable decision model for the decision maker based on a set of reference examples.

Remark 1. *The need-capacity matching methodology presented herein is inspired by MAUT approaches, with the utility (matching degree) of alternatives (candidate capacities in response to a need) being evaluated according to several criteria independently [7], and an overall (aggregated) rating being subsequently calculated for each alternative [13].*

Remark 2. *In the application context tackled within this paper, utility values are not regarded as preferences, since they are calculated upon a priori existing multimodal information (location data, user information, urgency information, etc.) related to the need or capacity (unlike preferential information usually provided by one or several decision makers, experts or stakeholders [7]). Therefore, the term preference is avoided hereinafter to avoid confusion.*

Under the settings considered above, a MCDM framework can generally be formulated as follows:

- There exists a decision problem, consisting of $z \geq 2$ alternatives or possible solutions, $X = \{x_1, \ldots, x_z\}$, e.g. different car models amongst which a customer chooses.

- Alternatives are assessed according to several independent (and usually conflicting) criteria, $Q = \{q_1, \ldots, q_n\}$, $n \geq 2$. For example, some possible criteria to consider when evaluating a car model could be: size, engine power and price.

Let $m_{jk} \in D$ denote the utility degree to which alternative $x_j \in X$ matches or satisfies criterion $q_k \in Q$. This value is expressed in an information domain D, some examples of which are: numerical, boolean, interval-valued or linguistic [14]. Intuitively, criteria with a qualitative nature, e.g. comfort level, may be easier to assess linguistically, e.g. *very low*, *high* [15, 16, 17]. Our work focuses on assessing alternatives as values in the [0,1] interval, such that $m_{jk} = 1$ indicates that x_j completely satisfies q_k, whereas $m_{jk} = 0$ indicates null satisfaction of the criterion by x_j.

In order to obtain an overall assessment value m_j for x_j, it is necessary to combine its n satisfaction degrees m_{jk} over the different criteria q_k. A common approach to do this in MCDM is by using aggregation functions [18]. An overview of aggregation functions is provided in the following subsection. An aggregation function f is thus utilized to compute m_j as follows:

$$m_j = f(m_{j1}, \ldots, m_{jn}) \tag{1}$$

Once an overall assessment or satisfaction degree m_j for each alternative is obtained, it can be used to determine and select the best alternative as the solution for the problem, or to rank several alternatives in decreasing order, depending on the nature of the decision problem. The proposed matching algorithm in this work aims at ranking *a subset of* candidate capacities for a need based on their matching degree, since the initial set of existing capacities may be large. Furthermore, the algorithm deems some criteria as more important than others, i.e. importance weights must be assigned to criteria, therefore it aggregates information about multiple criteria by using weighted aggregation functions [18].

2.2. Aggregation Functions in MCDM

Aggregation and fusion of information is a paramount process in most decision making processes; the application of intelligent aggregation techniques to simultaneously make use of information from different sources (e.g. multiple stakeholders and/or criteria) becomes vital to enable effective decision making, hence

it has unsurprisingly become an active subject of research amongst the MCDM community [19].

Aggregation functions have the purpose of combining a n-tuple of objects or values belonging to a set (usually the unit interval [20]) into a unique representative value in the same set. They are formally defined as follows:

Definition 1. *[20] An aggregation function in the unit interval is a mapping* $f : [0,1]^n \rightarrow [0,1]$, $n \geq 1$, *that produces an output value from a set of* n *input values or arguments* $A = a_1, \ldots, a_n$. *In the [0,1] interval,* f *satisfies the following properties:*

(i) ***Identity when Unary****:* $f(a) = a$.
(ii) ***Boundary conditions****:* $f(0, \ldots, 0) = 0$ *and* $f(1, \ldots, 1) = 1$.
(iii) ***Monotonicity or Non-decreasing****:* $a_k \leq b_k \, \forall k = 1, \ldots, n$ *implies* $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$.

Besides these three properties, which are recurrent in any aggregation function in the unit interval, there exist other mathematical and behavioral properties characterizing the different (families of) functions defined in the literature [18]. Some interesting examples of mathematical properties frequently considered for aggregation of information in MCDM include, amongst others:

1. **Idempotence**: f$(a, a, \ldots, a) = a$.
2. **Compensation**: $\min_{i=1}^{n} a_i \leq f(a_1, \ldots, a_n) \leq \max_{i=1}^{n} a_i$.
3. **Associativity**: $f(a, f(b, c)) = f(f(a, b), c)$.
4. **Reinforcement**: It refers to the tendency of (i) a set of high values to reinforce each other, leading to an even higher aggregated value (*upward reinforcement*), or (ii) a set of low values to reinforce each other, leading to an even lower aggregated value (*downward reinforcement*).

In addition to mathematical properties, there also exists some interesting behavioral properties, including [19]: (i) the possibility of allowing the decision maker to reflect attitudes within the aggregation process, such as optimistic, pessimistic, etc., and (ii) the use of weights to assign importance to the values to aggregate, thus privileging the influence of some criteria above others.

A widely known family of aggregation functions utilized in many domains are the *averaging operators*. These functions output a representative value that lies between the lowest and the highest values in A, and they include the "prototypical" aggregation operator, i.e. the *arithmetic mean*, as well as the median, k-order

statistics, minimum, maximum, etc., and, more generally, the quasi-arithmetic means family. Special attention in MCDM domains has attained the *Ordered Weighted Average* (OWA) operator [21], due to its versatility:

$$OWA_W(a_1,\ldots,a_n)=\sum_{k=1}^{n} w_k b_k \tag{2}$$

where $\sum_k w_k = 1$, and b_k is the k-ith smallest element in A, i.e. elements to aggregate are ordered decreasingly before assigning weights to them. The OWA operators encompass a family of averaging functions depending on the weighting vector $W = [w_1 \ldots w_n]$ used, ranging between the minimum (with $w_1 = 1$ and $w_i = 0, i \neq 1$) and the maximum ($w_n = 1$ and $w_i = 0, i \neq n$).

Besides the averaging operators, another two widely used families of aggregation functions are the *t-norms* and *t-conorms*. Instead of returning an "intermediate result", operators belonging to these families output values below the minimum (t-norms) or above the maximum (t-conorms) [22]. For this reason, they can be regarded as a generalization of the AND and OR logical operators, respectively, and their use becomes suitable in domains demanding a conjunctive (pessimistic) or disjunctive (optimistic) behavior in aggregation. T-norms fulfill the downward reinforcement property, whereas t-conorms present upward reinforcement.

Uninorm operators were proposed in [23, 24] as a generalization of t-norms and t-conorms accomplishing the full reinforcement property, which becomes attractive in many real-world decision problems because of its presence in many types of human information processing. Uninorms are associative operators with a neutral element g lying anywhere in the [0,1] interval. Given two values to aggregate $a, b \in [0,1]^2$, these operators behave as t-norms (downward reinforcement) in $[0,g]^2$, or as t-conorms (upward reinforcement) in $[g,1]^2$, and have a compensation behavior otherwise, i.e. in $[0,g) \times (g,1] \cup (g,1] \times [0,g)$. An example of a uninorm function is the cross ratio operator, with $g = 0.5$ [25]:

$$f(a,b)=\begin{cases} 0 & (a,b)\in\{(0,1),(1,0)\}, \\ \dfrac{ab}{ab+(1-a)(1-b)} & \text{otherwise.} \end{cases} \tag{3}$$

Fuzzy integrals are another family of aggregation functions widely studied for their application in MCDM frameworks. They are based on the concept of *fuzzy measure*, which assigns a weight to all of the different subsets of a set Q of criteria, i.e. it weighs each element in the power set 2^Q. This facilitates the

modelling of interactions between (subsets of) criteria. A well-known example of a fuzzy integral is the discrete Choquet integral [26], defined for a fuzzy measure μ as:

$$Choquet_{\mu}(a_1, \ldots, a_n) = \sum_{i=1}^{n} (b_i - b_{i-1})\, \mu(C_i) \tag{4}$$

where elements $b_i = a_{\sigma(i)}$ are arranged in increasing order (σ is a permutation function of $1, \ldots, n$), $b_0 = 0$ and $C_i = \{c_{\sigma(i)}, \ldots, c_{\sigma(n)}\}$.

Remark 3. *The choice of the most adequate aggregation function to be used in a particular application context largely depends, in most cases, on the attitude (e.g. optimistic) that should be reflected in the aggregation process.*

3. The COBACORE Information Framework and Ontology

To date, a variety of domain-specific information models have been developed for disaster information management. In the work by Liu et al. in [4], the ontology design and usability for crisis management state-of-the-art is reviewed. As a result, Liu et al. identified 26 existing ontologies that may be utilized within this application domain on the premise that crisis-oriented information systems are predicated on two groups of concepts involving 11 subject areas: (i) *Common concepts*: people, organizations, resources, disasters, geography, processes, infrastructure and damage; (ii) *Unusual concepts*: topography, hydrology and meteorology. However, few of these ontologies are formally represented and publicly accessible, and most of them describe concepts belonging to single subject areas, with a few only tackling the representation of multiple subject areas. Additionally, existing state-of-the-art ontologies for describing disaster response and management concepts still remain largely incomplete in terms of fully describing the disaster recovery domain.

The COBACORE information model attempts to overcome these drawbacks by defining an ontology upon a number of core concepts deemed necessary within the disaster recovery domain:

- *Actor*: Individuals and groups who are involved in the domain and can be claimed to have needs and/or capacities, as well as participate in activities. A group in the COBACORE platform is formed by an actor and can optionally have associated one or more needs, capacities or activities. The existence of a group implies the inclusion of at least one actor within it.

Every individual will also have their own needs, capacities and activities. Figure 1 illustrates the RFDS representation for this concept.

- *Need*: Expression of a necessity by an actor. Its definition is expanded to include not only essential requirements (e.g. assets, skills or information) but also crucial requirements. The distinction in this sense is determined by an urgency level attribute.

- *Capacity*: Potential resources that could be used to help fulfill a need. This concept not only includes the physical capability of an actor but also their possible skills, informational and financial capabilities (hence some specific capabilities might require physical intervention at a specific location). In line with the urgency level associated with a need, capacities have an associated duration and sphere of influence, which are indicated by an availability level.

- *Activity*: This concept describes an action within the domain that may be a direct response to a need. For example, an actor representing a community leader of a responding community may create an activity in order to address a need and mobilize volunteers. Subsequently, this target activity is associated with one or more needs and, if available, one or more matching capacities. If no capacities are matched to a need, the activity may be utilized to identify and further assign a capacity once available. Therefore, any activity must eventually be associated with an actor, need and capacity, and cannot exist without an actor.

Additional supporting classes were also defined to facilitate interlinking between concepts and to support an inference process within the model for subsequent knowledge discovery. One of these supporting classes is *Category*, which contains a number of subclasses representing the disaster recovery domain and type of resource associated with a need, capacity or activity, e.g. *Built Environment*, *Information*, *Transport*, etc. This information can be potentially used to conduct reasoning processes on instances of primitive classes and infer additional, implicit information about them. For example, an instance of a need that was specified as being of category *Transport* and type *Expertise* can be inferred as belonging to the class *InformationTransportNeed*. This way, the ontology is designed to facilitate inference-based matching between instances of a need and instances of a capacity, thereby providing a contextualized view of the requirements and capabilities of individuals and groups within the affected and

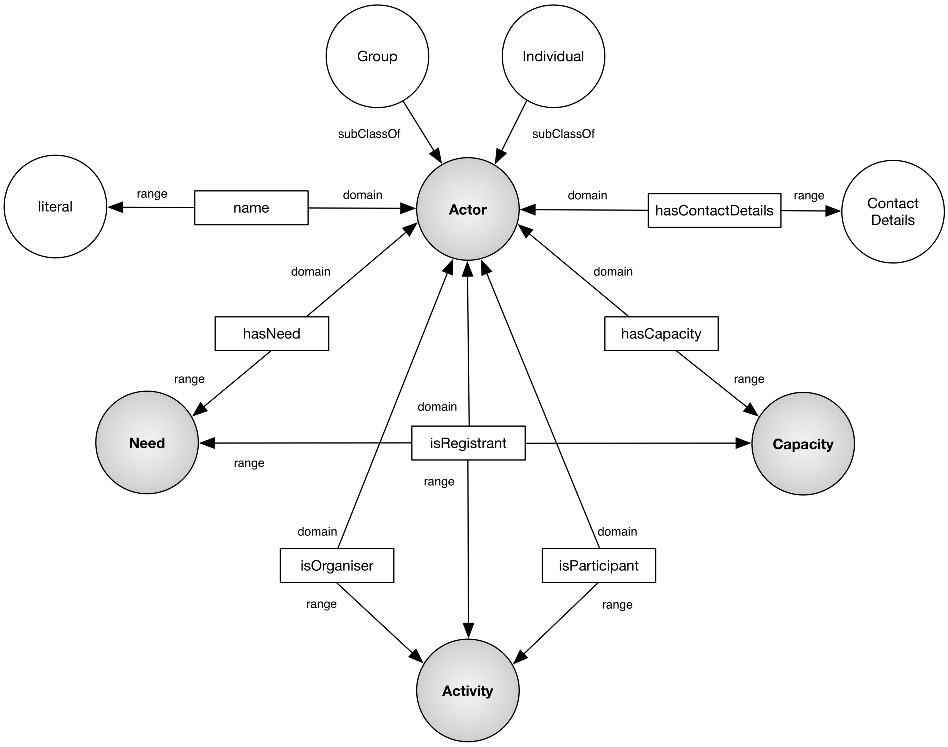


Figure 1: Resource Framework Description Schema for *Actor* Concept

responding communities. For example, candidate capacities inferred as *InformationTransportCapacity* could be regarded as potentially highly-matching for an *InformationTransportNeed*.

4. Semantic and MCDM-based Need-Capacity Matching Algorithm

This section presents a semantic inference and multi-criteria need-capacity matching algorithm for the COBACORE Information Framework. The goal of the proposed algorithm is to enable efficient and effective situational awareness and decision making by users across communities affected by disasters of different nature through its integration within the COBACORE platform.

The overall matching process applied by the algorithm consists of three stages:

(i) Firstly, the algorithm performs a semantic inference matching phase in order to filter a subset of candidate capacities in response to a target need $n_i \in \{n_1, \ldots, n_m\}$ based on the categories or types to which they belong.

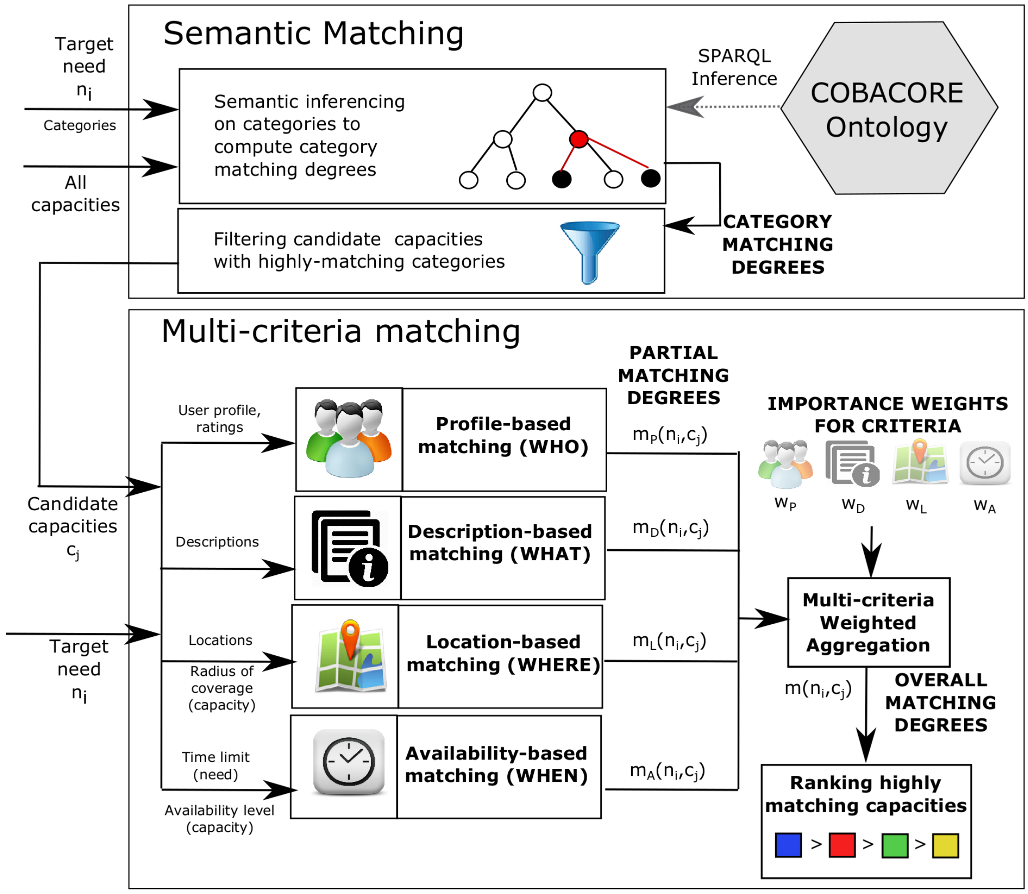


Figure 2: Scheme of the need-capacity matching algorithm

(ii) Secondly, it applies multi-criteria matching in order to calculate partial matching degrees $m_k(n_i, c_j) \in [0, 1]$ between n_i and each candidate capacity $c_j \in \{c_1, \ldots, c_z\}$, predicated on the 4W (Who, What, Where, When) approach.

(iii) Finally, all partial matching degrees are aggregated into an overall matching degree $m(n_i, c_j)$ for each candidate capacity. Based on $m(n_i, c_j)$, candidate capacities are ranked and a subset of the most highly-matching ones are returned.

Figure 2 depicts the general scheme of the matching algorithm being proposed, whose individual steps are further described below.

4.1. *Semantic Matching*

The aim of applying this phase prior to the multi-criteria matching stage is to reduce the initial (and presumably large) amount of information about existing capacities registered on the platform, filtering out semantically irrelevant capacities for the target need, and consequently reducing the computational cost of the subsequent multi-criteria matching process. The semantic matching phase is applied to identify a subset of z candidate capacities c_j for n_i, based on the common overarching categories to which they belong. A semantic datastore and the COBACORE ontology are utilized to analyze the categories to which needs/capacities belong (the SPARQL language is used for semantic querying over the datastore).

When a resource (i.e. need or capacity) is registered on the platform, two modalities of categorization are conducted:

(i) *Explicit Categorization*: The actor responsible for registering the resource can optionally indicate one or more *general categories* to which the registered need/capacity belongs. Examples of general categories defined in the ontology include *Built Environment*, *Transport*, etc.

(ii) *Implicit Categorization*: An automatic procedure is applied to discover additional, more specific categories (referred to as *base categories*) to which the registered need/capacity might belong. To do this, words in the resource description are sought within a set of pre-defined thesauri or subject *dictionaries* containing relevant vocabulary associated with each base category. Those base categories whose dictionary contains some words in the resource description are implicitly assigned.

Figure 3 shows an excerpt of the tree-structured category hierarchy defined in the ontology (edges represent *subClassOf* relations in the ontology). We point out that the hierarchy can be potentially extended to permit the flexible addition or deletion of categories in the future, hence the matching approach described herein is completely adaptable to possible future refinement.

Assigned categories are utilized together with the ontology in order to apply an inference process aimed at extracting additional knowledge about higher-level categories to which the need/capacity potentially belongs. This inference process has the advantage of facilitating more accurate computation of the category-based matching degree, particularly in the case when no explicit categorization was performed by the user (see Figure 3 as an example where the two shaded base categories are compared via inference, identifying *Built Environment* as a common ancestor of both; without inferencing, this matching would be disregarded).

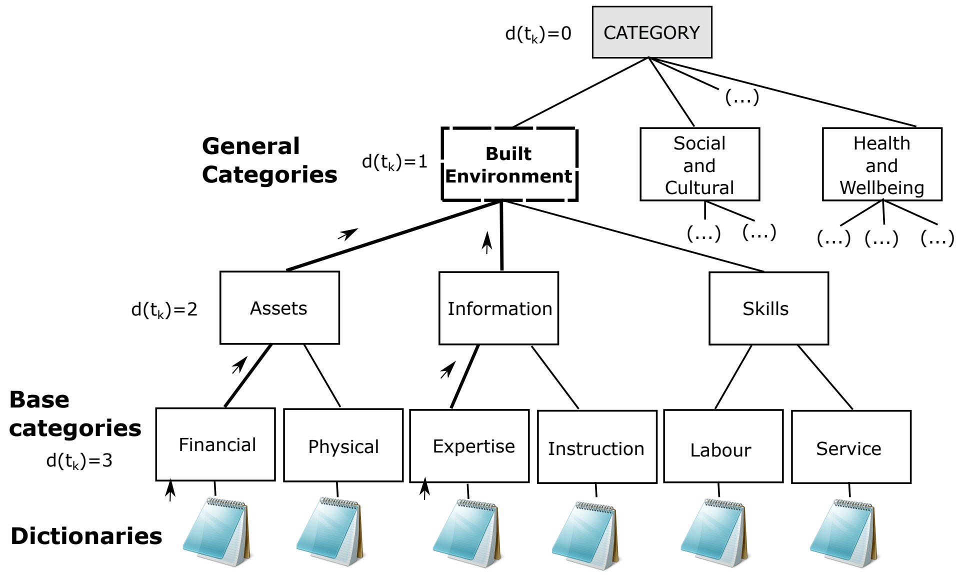


Figure 3: Excerpt of the ontology's category hierarchy and example of inference

Given the tree structure of the category hierarchy, matching degrees are computed based on the position of the Lowest Common Ancestor (LCA) [27] for each pair of categories assigned to the need and capacity. Let $T_i = \{t_{i1}, \ldots, t_{ig}\}$ and $T_j = \{t_{j1}, \ldots, t_{jh}\}$ be the categories assigned to a need n_i and a capacity c_j, respectively. Since every category t_k is associated to a node in the category hierarchy, applying an upwards inference process against the hierarchy allows the higher-level categories for t_k to be obtained and, consequently, the LCA category (*node*) for every pair t_l, t_k such that $t_k \in T_i$ and $t_l \in T_j$, is easily determined. Let d_k, d_l denote the depth of categories t_k and t_l in the hierarchy, and d_{LCA} denote the depth of the LCA, with $d_k, d_l, d_{LCA} \in \mathbb{N}$. d takes a value of 0 if the LCA is the root node (i.e. the abstract class representing all categories), and a value greater than 0 otherwise. Then, the category-based matching degree between n_i and c_j in categories t_k, t_l, denoted by $m_C^{kl}(n_i, c_j) \in [0, 1]$ is computed as follows:

$$m_C^{kl}(n_i, c_j) = \frac{d_{LCA}}{\max(d_k, d_l)} \tag{5}$$

Pairwise matching degrees are combined using the maximum operator (assuming here an optimistic aggregation attitude to consider the most highly-matching pair(s) of categories between the need and the capacity) in order to obtain the overall category-based matching degree between n_i and c_j:

$$m_C(n_i, c_j) = \max_{kl}(m_C^{kl}(n_i, c_j)) \tag{6}$$

Finally, given $z \in \mathbb{N}$ we apply a filtering process in order to obtain the subset of candidate capacities as the top-z matching capacities based on $m_C(n_i, c_j)$. These z capacities are subsequently used as the input set of alternatives for the multi-criteria matching phase, as outlined below.

4.2. Multi-Criteria Matching

Once a subset of candidate capacities have been extracted as an output to the semantic matching phase, multi-criteria matching is applied, which consists of procedures that analyze information relating to the needs and capacities that is stored within the platform's relational database. A different procedure is followed to compute partial matching degrees for each of the 4W criteria:

(i) *WHO*: involves analyzing the user *profiles* of the actors that registered either a need or a candidate capacity.
(ii) *WHAT*: consists of determining the similarity between the textual *descriptions* of the target need and candidate capacities.
(iii) *WHERE*: calculates the physical distance between the actors' *locations* in order to identify the most highly matching capacities.
(iv) *WHEN*: involves comparing the *urgency* level of the registered need against the *availability* level of candidate capacities.

The procedures followed to obtain these four matching degrees have been adapted to deal with the type(s) of information related to each criterion accordingly. They are described in the following subsections.

4.2.1. Profile-based Matching (WHO)

This matching degree is computed as shown in Figure 4, by taking into consideration the information in the user profile of the two involved actors, namely: (i) the rating, ρ_i, associated with the affected actor that registered a need n_i, and (ii) the rating, ρ_j, associated with the responding actor that registered a candidate capacity c_j, with both values in a (continuous) numerical rating domain $\rho_i, \rho_j \in [0, 5]$. More specifically, an aggregated pairwise rating between both actors is taken into consideration such that a higher aggregated rating implies a higher potential matching degree under this criterion.

A rating is defined as the score representing the average trust or reputation assigned to an actor by other actors in the system who rated him/her. An actor may

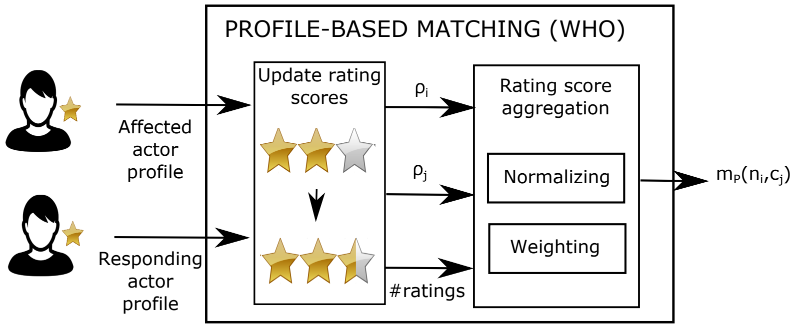


Figure 4: Profile-based need-capacity matching

rate another actor with a value in the discrete scale $\{0, 1, \ldots, 5\}$, and $\#ratings_i > 0$ is the number of ratings the actor received so far. When the actor receives a new rating $\rho_i^{new} \in \{0, \ldots, 5\}$, his/her average rating $\rho_i \in [0, 5]$ is updated as follows:

$$\frac{\rho_i^{new} + \rho_i \cdot \#ratings_i}{\#ratings_i + 1} \tag{7}$$

Remark 4. *By default, the system assigns a default rating score of 3 to a new user who has not been rated yet, in which case $\#ratings_i = 1$.*

The average rating score between the actor that registered n_i and the actor that registered a candidate capacity c_j is subsequently utilized to obtain the profile-based matching degree, $m_P(n_i, c_j) \in [0, 1]$,

$$m_P(n_i, c_j) = \phi_W(\overline{\rho}_i, \overline{\rho}_j) \tag{8}$$

with $\overline{\rho}_i$ a normalized score in the unit interval and ϕ an aggregation operator. The choice of an aggregation operator in Eq. (8) depends on the attitude we wish to reflect in the user profile matching (see Section 2.2). If a weighted operator is chosen, the weighting vector $W = \{w_i, w_j\}$ is defined such that:

$$w_i = \frac{\#ratings_i}{\#ratings_i + \#ratings_j} \tag{9}$$

$$w_j = \frac{\#ratings_j}{\#ratings_i + \#ratings_j} \tag{10}$$

Therefore, actors with a higher number of ratings are deemed more influential in the computation of $m_P(n_i, c_j)$ since their average score is further consolidated.

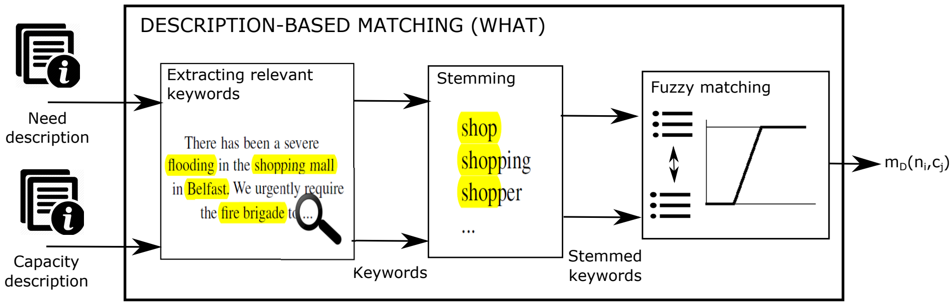


Figure 5: Description-based need-capacity matching

4.2.2. Description-based Matching (WHAT)

Descriptions of needs or capacities consist of free text of variable length introduced by the actors who registered them. The proposed procedure to estimate how similar two descriptions are aims at finding common relevant words (keywords) between them. To do this, the algorithm firstly applies Natural Language Processing (NLP) to extract keywords from raw descriptions. Moreover, since the concept of *similar* is ambiguous within this context, we consider the use of fuzzy set theory to calculate a (fuzzy) matching degree between both descriptions. The overall procedure is illustrated in Figure 5, and consists of the following three steps:

(i) *Extracting relevant keywords*: A Part Of Speech (POS) tagging algorithm is utilized to tag each word in the description with its corresponding POS (e.g. definite article, noun, preposition, etc.). Based on this tagging, words belonging to the least relevant categories, in terms of the overall description meaning (such as articles and prepositions), are left out.

(ii) *Stemming*: An approach based on the Porter stemming algorithm [28], is applied on relevant words in order to determine their lemma, i.e. their basic form within the corresponding word-family. This may significantly increase the detection of similar words between descriptions in cases where they belong to the same word family (e.g. "emergency - emergencies", "reforest - reforesting", etc.).

(iii) *Determining fuzzy matching degree*: Let $\#T_i$ and $\#T_j$ be the number of relevant (stemmed) keywords in the descriptions of n_i and c_j, respectively. The computation of the description-based matching degree consists in determining the degree to which the number of common keywords amongst them, $\#T_{ij}$, complies with the concept of *similar*. Due to the vague and imprecise

nature of this concept within this application, we use fuzzy set theory to model the concept [29]. To do this, we define a fuzzy set $SIMILAR$ in the [0,1] interval, with semi-trapezoidal membership function (as illustrated in Figure 6):

$$\mu_{SIMILAR}(x) = \begin{cases} 0 & \text{if } x \leq \alpha, \\ \dfrac{x-\alpha}{\beta-\alpha} & \text{if } \alpha < x < \beta, \\ 1 & \text{if } x \geq \beta,. \end{cases} \tag{11}$$

where $\mu_{SIMILAR}(x) \in [0,1]$ indicates the degree to which a value $x \in [0,1]$ (in our case, the number of common words between two descriptions) complies with the notion of $SIMILAR$.
The values of the two parameters $\alpha, \beta \in [0,1]$ that define the fuzzy membership function allow more or less tolerance with the notion of similarity between descriptions. The description-based matching degree is finally computed as:

$$m_D(n_i, c_j) = \mu_{SIMILAR}\left(\frac{\#T_{ij}}{\min(\#T_i, \#T_j)}\right) \tag{12}$$

4.2.3. Location-based Matching (WHERE)

In this step, which is summarized in Figure 7, a matching degree $m_L(n_i, c_j) \in [0,1]$ is computed based on the distance between the geographical locations (spatial data) of a need n_i and a capacity c_j. Importantly, a special case occurs for capacities that may not require physical intervention, hence can be implemented regardless of the location of the need (e.g. providing informational resources or

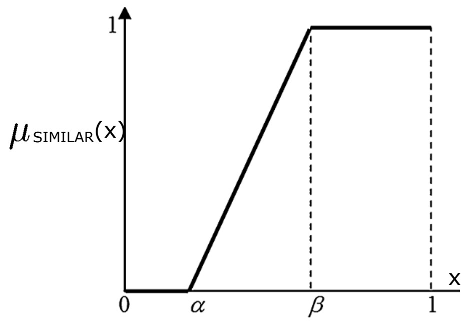


Figure 6: Fuzzy membership function for “SIMILAR”

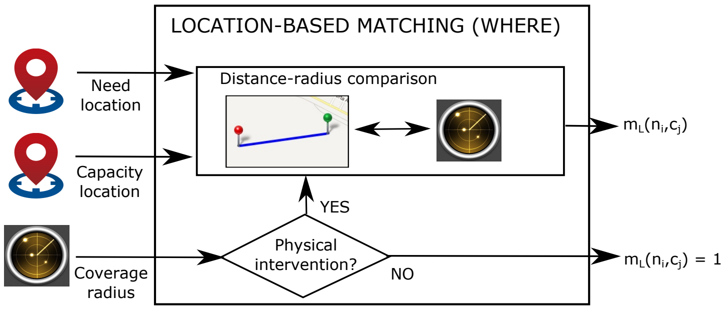


Figure 7: Location-based need-capacity matching

sending money via bank transfer). Conversely, capacities requiring physical intervention have an associated radius of coverage $r_j \in \mathbb{R}^+$ (expressed in kilometers). Capacities that do not require physical intervention show an empty value for the coverage radius within the platform's relational database.

Thus, $m_L(n_i, c_j)$ is computed as follows:

$$m_L(n_i, c_j) = \begin{cases} \max\left(0, \dfrac{r_j - d(n_i, c_j)}{r_j + d(n_i, c_j)}\right) & \text{if } \exists r_j > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

with $d(n_i, c_j)$ being the distance between the locations of n_i and c_j. This matching measure shows that:

- If $d(n_i, c_j) < r_j$, then n_i is within c_j's coverage radius, so $m_L(n_i, c_j) > 0$.
- If $d(n_i, c_j) > r_j$, then n_i is outside c_j's coverage radius, so $m_L(n_i, c_j) = 0$.
- If $\nexists r_j > 0$, then c_j does not require physical intervention and can be effectively implemented regardless of the geographical location of n_i, hence we assign $m_L(n_i, c_j) = 1$ in this case.

4.2.4. Availability-based Matching (WHEN)

The computation of this matching degree considers two aspects, as shown in Figure 8. On the one hand, it takes account of the urgency level specified for a need n_i, which is determined by the time limit under which responders are required to deal with it. Based on the date and time limit specified by the actor who registered the need, comparing it with the current date and time, an urgency level is assigned to the need. This urgency level is denoted by urg_i and takes a value within a numerical scale from 1 to 3, such that $urg_i \in \{1 : LOW\ (7+\ days), 2 :$

$MED\ (1-7\ days), 3:HIGH\ (<24\ h)\}$. On the other hand, capacities c_j have an associated availability level, $avail_j \in \{1, 2, \ldots, 10\}$, registered by their associated responding actors; the higher $avail_j$, the more availability to respond to urgent needs.

Under the rationale that highly urgent needs should be matched more exclusively to the most available capacities, to avoid filtering unsuitable results to the end-user we propose computing the urgency-based matching degree $m_A(n_i, c_j)$ as follows:

$$m_A(n_i, c_j) = \left(\frac{avail_j}{10}\right)^{urg_i} \tag{14}$$

By raising to the urgency level urg_i, the higher its value, the larger the penalty that is applied to the matching degree for low-availability capacities, hence they are more drastically discriminated from highly available (and hence more suitable) capacities in these cases.

4.3. Multi-Criteria Weighted Aggregation and Ranking

The last step of the algorithm is applied once all four matching degrees have been calculated. It consists of aggregating the four partial matching degrees in order to obtain the value for the overall matching degree $m(n_i, c_j)$ between n_i and each candidate capacity c_j. An aggregation function is utilized in order to obtain the overall matching degree. Since a requirement of the COBACORE platform is to assign different degrees of importance to the 4W criteria, we utilize a weighted average operator to compute $m(n_i, c_j)$:

$$m(n_i, c_j) = \frac{\sum_{k=1}^{4} w_k m_k(n_i, c_j)}{\sum_{k=1}^{4} w_k} \tag{15}$$

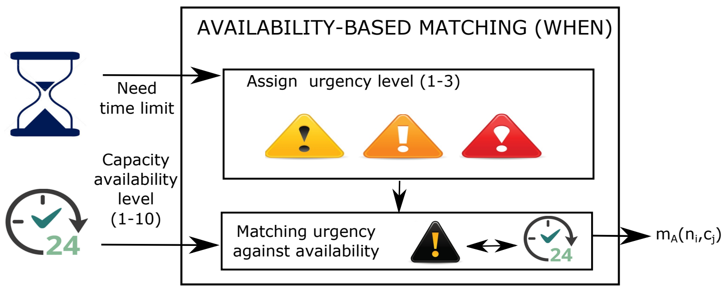


Figure 8: Availability-based need-capacity matching

where $k \in \{P : Who, D : What, L : Where, A : When\}$. Thus, some of the 4W criteria can be given greater importance than others in order to identify the most highly matching capacities in response to a given need. For instance, the following priority order between criteria, $Who \succ What \succ Where \sim When$, indicates that the user profile of actors is the most important criterion, whereas location and availability are the least relevant ones, in terms of coming up with an overall matching of capacities. In this case, the weighting vector $W = [w_P, w_D, w_L, w_A]$ must accomplish $1 \geq w_P > w_D > w_L = w_A > 0$.

Finally, given a fixed value $\gamma \in \mathbb{N}, \gamma \leq z$ representing the number of capacities to be returned as a result of the matching algorithm, the candidate capacities are ranked by decreasing order of matching degree, and the top-γ capacities in the ranked list are returned. This parameter is utilized in order to control the number of results returned, thereby preventing an overwhelming amount of information being returned to enable a rapid response procedure, while showing the most relevant results (matching capacities) first.

5. Application Example and Experimental Evaluation

In this section we demonstrate the performance and benefits of the proposed two-stage need-capacity matching algorithm. Throughout this evaluation, we refer to a target need n that will also serve to illustrate the overall matching algorithm.

We start by listing the relevant information of the the target need currently being considered (the actor personal data and location information of the need are partly omitted for the sake of brevity):

- *Categories*[2]: {Housing and Accommodation Physical Assets, Health and Basic Needs Service Skills, Safety and Security Service Skills}.
- *Actor profile*: (i) Normalized average rating: $\overline{\rho}_n = 0.85$, (ii) number of ratings: $\#ratings_n = 4$.
- *Description*: "My **house** just collapsed! I **need help**! My husband is under the **debris** and I can not see or hear him. I need people to rescue him!".

[2]The categories listed above are the result of applying implicit categorization at the beginning of the semantic matching stage, as no explicit categories were introduced by the actor when registering the need. The same applies to the example capacity c shown in this section.

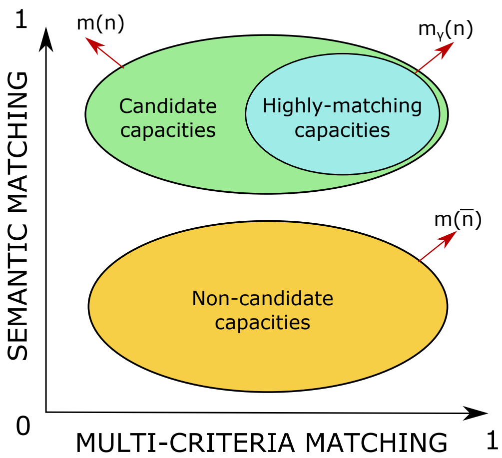


Figure 9: Subsets of capacities utilized for comparison of average matchings in the evaluation.

- *Location*: Schwalmtal, Germany.
- *Urgency level*: 3 (specified time limit of 1-2 hours).

The matching algorithm is repeatedly run for different values of z in the semantic matching phase, i.e. different sizes for the set of alternatives (candidate capacities) taken as input for the multi-criteria matching phase. More specifically, we consider the 5%,10%,15%, and 20% top-matching capacities in terms of $m_C(n, c_j)$, since for a larger z, capacities with a considerably low, or even null, semantic match with the target need would also be taken into consideration.

For each setting of z, the matching degree between n and each candidate capacity in $X = \{c_1, \ldots, c_z\}$ is determined, and an average matching overall, denoted $m(n)$, is subsequently calculated. Additionally, an average matching $m(\overline{n})$ between n and all non-candidate capacities (i.e. those capacities filtered out after semantic matching) is calculated and compared with $m(n)$ for evaluation purposes, as well as an average matching $m_\gamma(n) \geq m(n)$ focused on the top-γ matching capacities within X, i.e. those highly matched capacities that are returned to the end-user of the platform, with $\gamma = \lfloor \frac{z}{2} \rfloor$. For the sake of clarity, Figure 9 illustrates the relationship between these three subsets of capacities and the corresponding average matching to be computed.

We now showcase the different phases of the matching algorithm for an example of candidate capacity, denoted c, that moderately matches the target need:

- *Implicit Categories*: {Housing and Accommodation Physical Assets, Hous-

ing and Accommodation Expertise Information, Housing and Accommodation Labour Skills$\}$.

- *Actor profile*: (i) Normalized average rating: $\overline{\rho}_c = 0.67$, (ii) number of ratings: $\#ratings_c = 6$.
- *Description*: "I can offer to come and help with cleaning **debris** from demolished buildings, **houses**, etc. Please contact me if you **need help**".
- *Location*: Ameersfort, The Netherlands. Coverage radius (km): 200.
- *Availability level*: 6.

The different matching degrees calculated are illustrated below:

- *Semantic Matching*: Clearly, since n and c have one base category in common at the bottom of the ontology hierarchy (Figure 3), *Housing and Accommodation Physical Assets*, this pairwise matching is equal to 1 and, consequently, the use of the maximum aggregation function in Eq. (6) leads to $m_C(n, c) = 1$.
- *Profile-based Matching*: $m_P(n, c)$ is calculated by using a weighted average operator. Based on the number of ratings in the profiles of the two actors, by applying Eqs. (9) and (10) we have that $w_n = 0.4$ and $w_c = 0.6$, therefore:

$$m_P(n, c) = 0.85 \cdot 0.4 + 0.67 \cdot 0.6 = 0.74$$

- *Description-based matching*: After applying the NLP described in Section 4.2.2, the number of relevant keywords in each description is $\#T_n = 10$ $\#T_c = 8$, whilst $\#T_{nc} = 4$. Supposing a fuzzy set $SIMILAR$ has been defined such that $\alpha = 0.2, \beta = 0.6$, by applying Eq. (12) we have:

$$m_D(n, c) = \mu_{SIMILAR}(0.5) = 0.75$$

- *Location-based Matching*: The distance between the locations of n and c is $d(n, c) = 121$, and the responding actor specified a radius of coverage $r_c = 200$ for this capacity, hence:

$$m_L(n, c) = \max\left(0, \frac{200 - 121}{200 + 121}\right) = 0.24$$

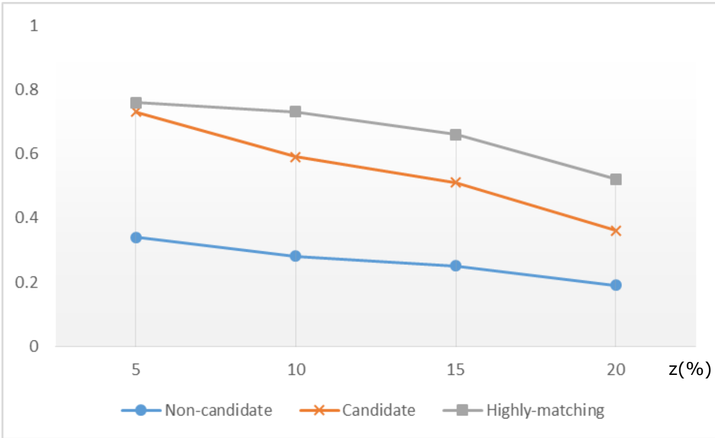


Figure 10: Average matching between n and (i) non-candidate capacities, $m(\overline{n})$, (i) candidate capacities, $m(n)$, and (iii) highly-matching capacities, $m_\gamma(n)$, for different sizes of z.

- *Availability-based Matching*: Taking $urg_n = 3$ and $avail_c = 6$ we have:
$$m_A(n, c) = 0.6^3 = 0.216$$
- *Multi-Criteria Matching (overall)*: Based on the weights currently assigned to criteria in the COBACORE platform, $W = [w_P : 1, w_D : 0.8, w_L : 0.7, w_A : 0.7]$, the resulting, overall matching between n and c is (Eq. (15)):
$$m(n, c) = \frac{1 \cdot 0.74 + 0.8 \cdot 0.75 + 0.7 \cdot 0.74 + 0.7 \cdot 0.216}{1 + 0.8 + 0.7 + 0.7} = 0.62$$

It is noteworthy to observe that, despite the high profile and description-based matching degrees obtained, the availability level of c does not properly suit the high urgency level required by n, as well as the large physical distance between the responding actor location and the location of the need. Consequently, this has led to low matching degrees being obtained in terms of the last two criteria.

Figure 10 illustrates the results of the experiments proposed at the beginning of this section. These results prove the effectiveness of applying semantic matching to filter out mostly irrelevant (i.e. weak matching) results before calculating the final (multi-criteria) matching degrees, regardless of the percentage of candidate capacities utilized. Consequently, this can translate into an improved end-user experience, i.e. the affected actor who registered a need is offered more relevant (matching) capacities to meet the registered need. Furthermore, it is shown how larger sizes of z and γ may lead to increasingly irrelevant results being returned, as the average matching degree tends to decrease.

6. Concluding Remarks

Within the scope of the Information Framework of the web-based COBACORE platform, this paper presented a semantic inference and multi-criteria approach to need-capacity matching for disaster response and recovery scenarios. The approach facilitates time-sensitive decision support through the analysis and identification of relevant capacities in response to targeted needs across responding and affected communities. This is achieved by (i) applying a semantic inference matching stage, based on categorization to extract a subset of candidate capacities, and (ii) applying a multi-criteria decision analysis and aggregation stage to determine the best matching candidate capacities for the targeted need in terms of their associated actors, descriptions, geographical locations and urgency-availability matching. The application of the matching algorithm has been illustrated, together with an experimental evaluation that highlights the effectiveness of the proposed two-stage approach for the extraction of highly relevant results for the end-user.

Acknowledgment

The work presented in this publication stems from the COBACORE project (Community-based Comprehensive Recovery, `www.cobacore.eu`) and is funded by the European Union FP7 Security Programme under grant agreement number 313308.

[1] V. Hristidis, S.-C. C. ahd T. Li, S. Luis, Y. Deng, Survey of data management and analysis in disaster situations, Journal of Systems and Software 83 (10) (2010) 1701–1714.

[2] C. Zhang, T. Zhao, L. Weidong, Automatic search of geospatial features for disaster and emergency management, International Journal of Applied Earth Observation nand Geoinformation 12 (6) (2010) 409–418.

[3] M. Neef, K. van Dongen, M. Rijken, Community-based comprehensive recovery: Closing collaboration gaps in urban disaster recovery, Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2013) (2013) 546–550.

[4] S. Liu, C. Brewster, D. Shaw, Ontologies for crisis management: A review of state of the art in ontology design and usability, Proceedings of the 10th International ISCRAM Conference.

[5] S. Staab, R. Studer, Handbook on Ontologies (International Handbooks on Information Systems), Springer, 2009.

[6] M. Doumpos, E. Grigoroudis, Multicriteria Decision Aid and Artificial Intelligence, Wiley, 2013.

[7] J. Lu, G. Zhang, D. Ruan, F. Wu, Multi-Objective Group Decision Making, Imperial College Press, 2006.

[8] P. Pardalos, Y. Siskos, C. Zopounidis, Advances in Multicriteria Analysis, Kluwer Academic Publishers, 1995.

[9] R. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Cambridge University Press, Cambridge, 1993.

[10] M. Ehrgott, Multicriteria Optimization (2nd Ed.), Springer, Berlin, 2005.

[11] B. Roy, The outranking approach and the foundations of ELECTRE methods, Theory and Decision 31 (1991) 49–73.

[12] E. Jacquet-Lagréze, Y. Siskos, Preference disaggregation: 20 years of MCDA experience, European Journal of Operational Research 130 (2001) 233–245.

[13] J. Fodor, M. Roubens, Fuzzy Preference Modelling and Multicriteria Decision Support, Kluwer, Dordrecht, 1994.

[14] F. Herrera, L. Martínez, P. Sánchez, Managing non-homogeneous information in group decision making, European Journal of Operational Research 166 (1) (2005) 115–132.

[15] L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning (i), Information Sciences 8 (3) (1975) 199 – 249.

[16] L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning (ii), Information Sciences 8 (4) (1975) 301 – 357.

[17] L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning (iii), Information Sciences 9 (1) (1975) 43 – 80.

[18] G. Beliakov, H. Bustince, T. Calvo, A Practical Guide to Averaging Functions, Springer International Publishing, 2016, Ch. Review of Aggregation Functions, pp. 1–53.

[19] M. Detyniecki, Fundamentals on aggregation operators, LIP6 Research Report:2001-2002, University of California, Berkeley.

[20] G. Beliakov, A. Pradera, T. Calvo, Aggregation Functions: A Guide for Practitioners, Springer, 2007.

[21] R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Transactions on Systems, Man and Cybernetics 18 (1) (1988) 183–190.

[22] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.

[23] R. Yager, A. Rybalov, Uninorm aggregation operators, Fuzzy Sets and Systems 80 (1996) 111–120.

[24] J. Fodor, R. Yager, A. Rybalov, Structure of uninorms, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 5 (1997) 411–427.

[25] E. Klement, R. Mesiar, E. Pap, On the relationship of associative compensatory operators to triangular norms and conorms, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 04 (02) (1996) 129–144.

[26] G. Choquet, Theory of capacities, Annales de L'Institut Fourier 5 (1954) 131–295.

[27] M. A. Bender, M. Farach-Colton, G. Pemmasani, S. Skiena, P. Sumazin, Lowest common ancestors in trees and directed acyclic graphs, Journal of Algorithms 57 (2) (2005) 75 – 94.

[28] M. Porter, An algorithm for suffix stripping, Program 40 (3) (1980) 211–218.

[29] L. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.