# Ontology-based integration of humanitarian aid information for disaster management systems

Pasinee APISAKMONTRI

Japan Advanced Institute of Science and Technology

**Doctoral Dissertation**

# Ontology-based integration of humanitarian aid information for disaster management systems

by

Pasinee APISAKMONTRI

*Supervisor:* Professor Mitsuru IKEDA

*School of Knowledge Science*
*Japan Advanced Institute of Science and Technology*

December, 2015

# Abstract

A humanitarian aid in an emergency system involves information from multidisciplinary environments. The humanitarian aid information systems have been increased in recent years by many humanitarian agents. Collaboration of humanitarian aid information systems is needed in order that they perform more smartly and more effectively. A large number of humanitarian information is separately stored in several relational databases. An ontology is one solution to enable the information reusing and sharing using a common vocabulary across heterogeneous application. Semantic interoperability between existing relational databases and ontology still remains a major practical issue. In this research, we design a pivot ontology framework to present pivot construction methodology and a PivotOntology-to-Database schema matching methodology. The pivot construction methodology is adopted from an ontology engineering technique. The pivot ontology is the semantic neighborhood among various databases. The schema matching between pivot ontology and a database focuses on linguistic relation approach. To integrate humanitarian aid in emergency information from several databases, the Humanitarian Aid for Refugee in Emergencies (HARE) ontology has been proposed. The evaluations demonstrate that the HARE ontology successfully integrates with extensive schema covering of the existing databases. In addition, Humanitarian Aid Information Processing (HAIP) Model has been designed with respect to situation awareness model. The HAIP model is combined the humanitarian aid information integration processes to achieve their knowledge. In order to forecast the future situation, the HAIP model helps to get the information needs for decision making systems.

**Keywords**: Ontology, Pivot Ontology, Humanitarian Aid in Emergency, Relational Database, Schema Matching, Interoperability, Knowledge representation

# Acknowledgments

I would like to express the deepest appreciation to my committee chair Professor Mitsuru IKEDA, who has shown the attitude and the substance of a genius: he continually and persuasively conveyed a spirit of passion in regard to research and scholarship, and an excitement in regard to teaching. I would like to thank my Thai-sided advisor, Associate Professor Ekawit Nantajeewarawat, whose has been tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Without their supervision and constant help this dissertation would not have been possible. I would also like to thank my committee members, Associate Professor Hideaki KANAI, Professor Riichiro MIZOGUCHI, Associate Professor Nam Van Huynh for serving as my committee members even at hardship. I want to thank you for joining my defense and for your comments and suggestions, especially, MIZOGUCHI sensei, thanks to you. I would like to thank all staffs at Japan Advanced Institute of Science and Technology, Sirindhorn International Institute of Technology, Thammasat University, and National Electronics and Computer Technology Center to support me for my Ph.D. thesis. I also want to thank to SIIT-JAIST dual degree program for their financial support granted through predoctoral fellowship.

A special thanks to my family, my mother Nanta Apisakmontri, my father Suthep Apisakmontri for all the sacrifices that you have made on my behalf. I would also like to thank all of my friends who encourage me to strive towards my goal. It is also dedicated to Nareerat Saokham for the support. Thanks a lot for everthing. At the end I would like express appreciation to my beloved husband Varachai Rojchai who spent the times with and was always my support in the moments.

# Contents

"This dissertation was prepared according to the curriculum for the Collaborative Education Program organized by Japan Advanced Institute of Science and Technology and Sirindhorn International Institute of Technology, Thammasat University"

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Introduction

During a critical disaster period, various humanitarian actors such as governments, relief organizations, volunteers, and victims often begin gathering information and creating systems independently with little consolidation. Such information is incomplete and sometimes make a conflicting picture of humanitarian needs [IAS]. Therefore, a collaboration among different systems is limited to information collecting, sharing, and disseminating. Humanitarian aid information systems need their critical interoperability to be smarter and more useful. Achieving that goal requires collaborative technologies with information sharing and domain knowledge. Disaster Management Services (DMSs) have efficiently emerged because of catastrophes. For example, a disaster information system named Sahana, which is an open-source software application, was initiated after the Sumatra-Andaman earthquake in 2004 [CDSDS+06], and another one is the collaboration services that was used during the earthquake in Haiti in 2010, which Haitians used their mobile phones as collaboration tools to report missing persons, shelter problems and food issues. Collaboration should be more than human communication; It should create new ways for computer communication so humans can disseminate information.

Due to the increased focus on humanitarian aid in disaster in recent years, there is a growing need to capture this information and their relationship. Humanitarian aid information is heterogeneous information that affects people to response and rescue during/aftermath a crisis. Information, such as information on situational occurrences, victims, shelters, resources, facilities, etc., is usually rapidly changeable, ambiguous, and large.Diverse information and knowledge are widely distributed and owned by different organizations [ZZNJ02]. That means the information stores in distinct heterogeneous data sources in different locations. Successful and innovative collaboration solutions are limited by a large number of humanitarian aid actors, and incompatible information. The challenges of information integration in the humanitarian aid domain is to identify the correlation of data from several sources [FG03].

## 1.2 Statement of Problems

In information technology, an ontology enables to reuse and share domain knowledge using a common vocabulary across heterogeneous application. The definition of ontology ([PKR97, UJ99]) is a hierarchically structured set of terms and specification of their meanings for defining a structure on the domain and confining the interpretations of terms that can be used as a skeletal structure for a knowledge base. The ontology is a possible approach to deal with the semantic heterogeneity problem. For example, geographic information, which is the most related on disaster information, has been addressed by ontology [KLK06, XZ07b]. Another knowledge has been represented by ontology such as, an emergency preplan ontology [HGXR09], sensors ontology for emergency management [CBB+12]. Although there are existing ontologies for disaster management, these ontologies are application-dependent ones. They are represented for a specific application.

As we mentioned above, information integration is a main issue in this domain. Most developed applications are based-on Relational-Database (RDB). Many humanitarian aid organizations have their own database. The large-scale database integration becomes a critical process. A database-to-database integration is a general problem between two databases that normally achieved by database-to-database schema matching technique [RB01]; however, a schema matching between a large number of different database schemata must be relied on many database administrator and time consuming.

Our research provides a basis for common understanding of humanitarian aid in disaster management in a formal and correct way for both information and expert knowledge interoperability. To achieve a semantic framework for humanitarian aid domain, we design two main complements; pivot ontology construction and ontology-to-database schema matching. Normally, pivot ontology is the semantic neighborhood of ontologies, which is correspondent from the pivot ontology to the other ontologies [NDdMRGAM09]. The existing humanitarian aid information systems have not been completely represented by ontologies. Rather, these systems are based-on relational databases that are not initially developed for integrating. They cannot be explicitly shared information without a shared common understanding for exchange. For this reason, we conduct a research for modeling pivot ontology that has the semantic neighborhood among a pivot ontology and various database neighborhoods for humanitarian information integration. We introduce an ontology engineering methodology to develop a pivot ontology for Humanitarian Aid for Refugee in Emergencies (HARE), which can support interoperation among heterogeneous systems and provide a completely guideline for the ontology-to-database schema matching.

## 1.3 Research Objectives

- To establish a pivot ontology model for ontology-based integration among existing humanitarian aid information systems, which belong to different organizations, in order to prevent the fully connected network problems.

- To capture humanitarian aid knowledge from the documents of United Nations High Commissioner for Refugees (UNHCR), which provides humanitarian principles for humanitarian action in worldwide. Then, we propose a pivot ontology for humanitarian aid information systems called Humanitarian Aid for Refugees in Emergencies (HARE) ontology. The compatibility, coverage, and accuracy of HARE ontology are evaluated by matching the existing humanitarian aid information systems that were not operated by UNHCR through HARE ontology.

- To test the appropriate matching strategies for humanitarian aid in emergencies information integration.

## 1.4 Research Methodologies

To achieve this objective, we are proposing three goals:

1. Formalizing this proposed pivot ontology model in terms of a Humanitarian Aid for Refugee in Emergencies (HARE) ontology. The HARE ontology development needs to overcome the barrier of ontology-based architecture that is ontological engineering. We apply Uschold and King's method and expand activity steps in this method. An extended method has been reformed steps in order to improve knowledge capturing, ontology integrating with upper ontologies, and enhance a schema matching process which combines element and structure levels to the method. HARE ontology consists of conceptualization from many domains, such as geography, healthcare service, transportation, etc. Thus, by including upper ontology integration, it will provide semantic interoperability of ontologies across multiple domains to express and share humanitarian aid in emergencies knowledge among systems.

2. Integrating humanitarian information by modeling a guideline for using the HARE ontology as a pivot between existing relational databases. A manual matching is done between the HARE ontology and the existing relational databases. A case study is developed by considering two relational databases: Sahana Eden and Ushahidi systems. The set of correspondences is returned as an output of the case study for sharing information among multiple sources via the HARE ontology.

3. Analyzing the combination of semi-automatic matching strategies for humanitarian aid in emergencies information integration. Because a fully manual matching is become increasingly infeasible for large and complex ontology or relational databases.

For this purpose, we use applied matching strategies based on manual matching guideline.

## 1.5 Research Impact

The quality of humanitarian response can be enhanced by formalizing the knowledge related to humanitarian aid in emergencies.The expected HARE ontology is able to provide a set of terminologies shared among the humanitarian aid systems. These terminologies enable the common understanding of humanitarian aid system. A formal definition of terminologies enable an explicit specification of a conceptualization. Interoperability is not only just the sharing of concept definitions, but also explicit sharing of intermediate concepts. Semantic interoperability enables the meaningful linkage of information and knowledge to humanitarian information. Our research focuses on realizing semantic interoperability across the whole emergency response systems, such as refugee registration, health care, donation, and shelter management system. Our approach is the cross-system information sharing via the HARE ontology to support the semantic interoperability among humanitarian aid information systems by using the independent-application ontology construction strategy and schema matching techniques. We believe that our research will be able to contribute the realizing for tracking all humanitarian aid information in the future.

## 1.6 Chapter Organization

The dissertation is organized as follows:

- Chapter 2 explores background and literature review
- Chapter 3 explains the pivot ontology framework for humanitarian aid systems and the HARE ontology construction methodology.
- Chapter 4 describes a method for ontology-to-database schema matching.

- Chapter 5 explains the correspondence analysis between existing humanitarian aid systems through the HARE ontology.
- Chapter 6 provides a model of Humanitarian Aid Information Processing (HAIP) for the future work based on situation awareness in order to understand humanitarian aid knowledge in an aspect of information integration process.
- Chapter 7 concludes the dissertation and future work.

# Chapter 2

# Background and Literature Review

Humanitarian aid information is one of the most valuable resources of emergency response participation. Humanitarian aid organizations are finding the information needed to plan and make decisions for saving lives, reducing suffering and respecting to human dignity. Main problems to be faced are related to situation awareness on identifying related information. This kind of information has to be collected from widely different data sources. As a consequence, The humanitarian aid organizations turn to share their information. However, how to share information with minimal modification of existing database schemata is a challenging research.

## 2.1 Information of Humanitarian Aid in Emergencies

There are two groups of disaster classifications as natural crisis (e.g., earth-quakes, hurricanes, floods, wildfires, etc.) and technological crisis (e.g., terrorism, nuclear power plant emergencies, hazardous materials, etc.).Disaster management has four phases: Preparedness, Mitigation, Response, and Recovery.

- Preparedness: this phase includes developing plan for emergency situations.
- Mitigation: this phase includes any activities that prevent an emergency.
- Response: this phase includes abilities to protect yourself, your family, or others.

- Recovery: this phase includes abilities to cope with rearranging your life and environment after an emergency.

Humanitarian aid activities are processes that covered in response and recovery phases. Information of disaster has the unique combination of characteristics [HCL+10], which include:

- Large number of data sources and information needs
- Time sensitivity of the exchanged information
- Lack of common terminology
- Several kinds of data (e.g., incident reports, damage reports, missing person reports, and rescue reports)
- Heterogeneous formats (e.g., free text, document forms, XML, and database)

Humanitarian aid information management systems would be involved with integrations of diverse heterogeneous sources, data ingestion and fusions. From the perspective of the emergency management community, three groups of actors include providers (donors), recipients (affected populations), and implementers (government, foundation, Red Cross, NGOs, and UN agencies). They would follow the humanitarian aid standards, such as UNHCR processes for control, coordination, and communication among entire actors.

During disaster, situation reports are generated by many actors and distributed to other actors for taking rescue actions. The response and recovery actions are recorded by information systems to construct an overall report for making an efficient decision. The implementers will provide a guidance for rescue teams based on each phase of the disaster event. The rescue teams are required to update their rescue information to their original affiliation.

Collecting disaster information from several data sources is still a challenge. Fig. 2.1 depicts an ambiguity of multiple agencies collaboration on information collection, information sharing, and disseminated information. An explicit collaboration is capable only

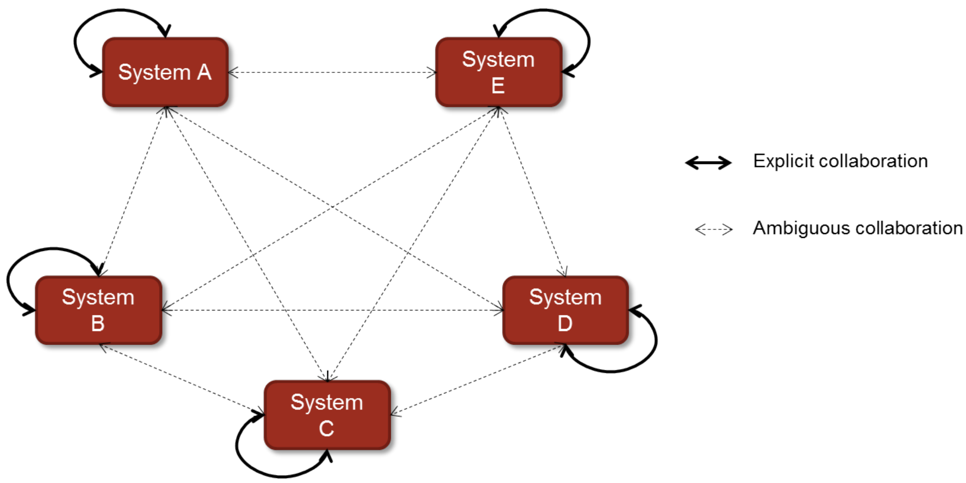



Figure 2.1: Explicit/Ambiguous collaboration among heterogeneous systems

sharing information on internal systems. These systems are usually built on a relational database.The example of those systems are commercial systems (e.g., Web EOC [Web] and E-Teams [ET]), and open source systems (e.g., Sahana [Sah] and Ushahidi [Ush]). These systems facilitate similar functions, such as resources tracking, mapping, document management, and data analysis. Techniques and tools for sharing information are based on individual implementers. A combination of this information is necessary for an effective emergency response. Therefore, the interoperability issue on humanitarian aid information is the primary goal of this research.

## 2.2 Interoperability Issues

Interoperability is an ability of two or more systems or elements to exchange information in a heterogeneous network and such exchanged information is able to be used [18291, SH04]. There are different levels of interoperability, i.e., system interoperability, syntactic interoperability, structural interoperability, and semantic interoperability [OS99]. We now focus on the semantic interoperability. Semantic interoperability is an ability to integrate data sources using different structures and different vocabularies but they have similar meaning with unambiguous and shared meaning. Fig. 2.2 presents that semantic interoperability in this research has meaning in two parts. First, in the semantic understanding part, and second, in the domain knowledge part. Semantic understanding is to understand the

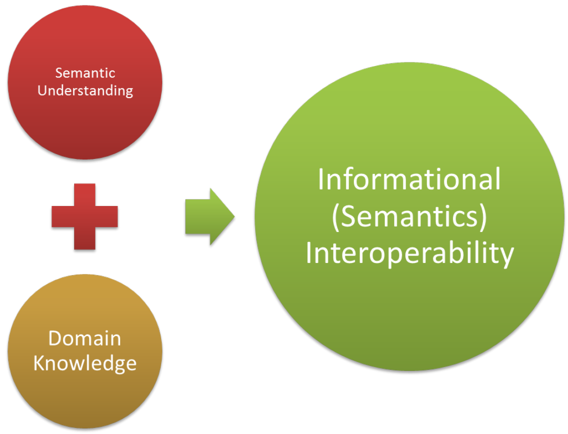



Figure 2.2: Semantic Interoperability Framework

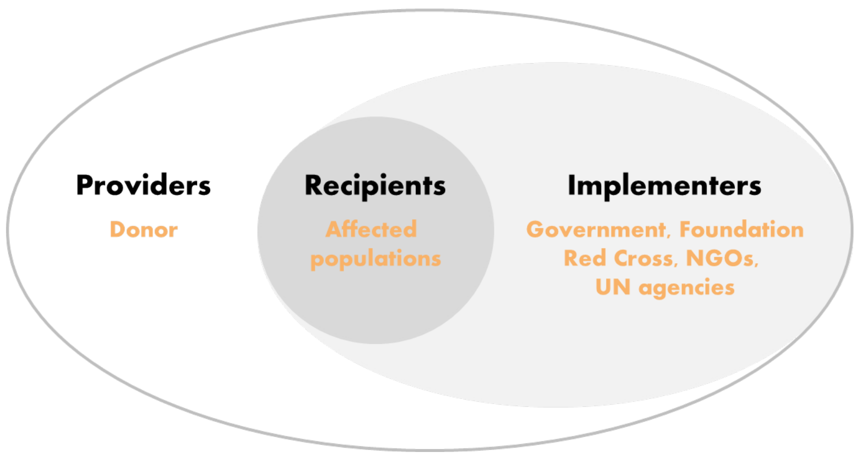



Figure 2.3: Core Actors of Humanitarian Aid System

concepts that contained in the systems.Domain knowledge is to learn the relevant domain knowledge that applies semantics.

To achieve semantic interoperability, it involves a large number of effort and knowledge from many different systems in humanitarian aid domain. Three actors (Providers, recipients, and implementers) have been working on different agencies (Fig. 2.3). Many agencies around the periphery are poorly coordinated. To fix this issue, they need to team up to achieve the interoperability. It will be enable the cross-systems information in both semantic understanding and domain knowledge parts.

The lacking of semantic interoperability among heterogeneous information is a serious issue. There are many approaches that can be applied for the integration, such as semantic modeling, formal logic-based approaches, classifications of terminology, formal languages, knowledge-based systems, context-sensitive information processing, rules of interaction mechanisms, knowledge sharing, semantic correlation, schema matching, and the use of a shared ontology approaches [OS99, RP04].

### 2.2.1 Schema Matching

Schema matching, which is a basic approach to generate correspondences, is the two or more schemata mapped among elements of schemata that semantically correspond to other [RB01]. A schema can be a formal structure, such as a SQL schema, XML schema, entity-relationship diagram, and ontology description. Initially, schema matching is the problem of a database schema integration. A database is usually designed for a particular application. Obviously, different databases have different structures and terminologies.

The integration of information from diverse heterogeneous sources is a challenging issue. Several studies have proposed the techniques for integrating heterogeneous information. Current approaches of semantic interoperability in heterogeneous information have been solved by a database-to-database integration. Batini [BLN86] provided a unifying framework for schema integration problem. his research defined two kinds of schema integration:

1. View integration: schemata are integrated during database design.
2. Database integration: local schemata are taken together in a global schema.

The activities of schema integration [BLN86] are collated into four main steps as follows;

1. Pre-integration: to consider integration processing strategies (i.e., Binary, and N-ary strategies).
2. Comparison of Schemata: to check all conflicts of the same elements in different schema (i.e., Naming, and structural conflicts), and to discover inter-schema properties.
3. Conforming of Schemata: to conform or align schemata to make schemata compatible for integration.

4. Merging and Restructuring: to perform different kinds of operations on either the component schemata or the temporary integrated schema.

Rahm, Madhavan, Shvaiko, and Bernstein [RB01, MBR01, SE05, BMR11] have studied levels of schema-based matching and their techniques for matching databases. Element and structure levels are two main levels of their studies.

**Schema Matching Techniques**

**Individual Matcher**

- *Element level* - using Linguistic matching (e.g., name similarity, description similarity, and global namespaces), Auxiliary information technique (e.g., thesauri, acronyms, dictionaries, and mismatch lists checking), and Constraint-based matching (e.g., type similarity, and key properties) [RB01, MBR01, BMR11]
- *Structure level* - using Constraint-based matching (Graph matching, Usage-based matching, Document link similarity) [RB01, MBR01, BMR11]

**Combining Matcher** is strategies to flexibly combine multiple matching approaches and to compare large schemata, such as parallel matching [GHKR10], workflow-like strategies [HQ07], and early search space pruning [PBR10].

- *Hybrid matcher* - combine various matching approaches to specify match candidates.
- *Composite matcher* - combine the results of various separately executed matcher, including hybrid matcher.

Various studies have been proposed ontology-to-ontology-matching systems in order to address interoperability in heterogeneous sources. For example, Khan [KS14] designed rules that can handle the heterogeneity and inconsistency in hierarchical ontologies. Opu [OAM14] developed an ontology generalization tool that can generalize two ontologies into an ontology.

The reconciling structure and terminology of two or more database schemata are required to solve a database schema integration problem. However, humanitarian aid information is heterogeneous information from diverse databases. An interoperability is needed by information providers in order to enhance information sharing for the emergency response. In knowledge engineering, the idea of ontology has been introduced as knowledge modeling for wider usability of knowledge base. In this research, an ontology is proposed to be a global schema for the database integration as we called a pivot ontology. The pivot ontology is not only reconciling the database integration that prevents fully connected network problems, but also becoming a first step towards standardization that provides explicit intermediate concepts.

## 2.3 Ontology

Ontology is a prominent key component of the possible solution for semantic interoperability. The interoperability facilitates cross-boundary information sharing among organizations and individuals. A term ontology has been used in many ways. A popular definition in computer science is that an ontology is an explicit specification of a conceptualization [Gru93]. Ontologies have been largely used in several researches and incorporated in many fields such as medical, geography, business, tourism, disaster management, etc.

### 2.3.1 Ontologies in Humanitarian Aid

Disaster management will have the success of management as getting the right resources to the right place at the right time; to provide the right information to the right people to make the right decisions at the right level at the right time. Interoperability is one challenge when the involving organizations try to integrate these individual data sources [RK08, Sot07, DM07, GR11, SHM13]. The basis of an ontology can cope with the confusion because an ontology is a logical theory accounting for the intended meaning of a formal vocabulary [XZ07a, FZ11].

In emergency management, ontology has been used in many studies.Maio [Mai07] proposed an idea of open ontology concept for open source emergency response systems to support knowledge and semantic consistency. Xiang [LLL+08] proposed a practical emergency response ontology from emergency response workflows and implemented a prototype of emergency evacuation planning system. Fan [FZ11] proposed a model of dynamic data ontology, and static data ontology to solve the spatial data problems in emergency response. However, neither integrating the existing humanitarian aid systems using pivot model to represent the knowledge nor matching techniques to support integration with minimal modification of the existing systems were addressed by previous studies.

### 2.3.2 Ontology Construction

The core of pivot ontology is a common ontology. Existing approaches to construct ontology include Uschold and King's method [UK95] intended for enterprise ontology construction, the TOVE (TOronto Virtual Enterprise) project ontology [GF95], the METHONTOLOGY methodology [FLGPJ97] intended for building life cycles based on evolving prototypes, the SENSUS methodology [SPKR96] intended for linking domain terms to a large-scale ontology, the On-To-Knowledge methodology [SSSS01] intended for utilizing ontologies for improving knowledge management quality. The Uschold and Kings method and the METHONTOLOGY methodology apply application-independent strategies, and the SENSUS methodology uses an application-semi-dependent strategy. By contrast, the On-To-Knowledge methodology employs an application-dependent strategy. An ontology created using an application-independent strategy is likely to be more reusable, compared to that developed using an application-dependent strategy [G07].

To design a common ontology for Humanitarian Aid for Refugees in Emergencies (HARE), an application-independent strategy, techniques for reusing existing ontologies, semantic hierarchical conceptual models, and ontology engineering techniques for solving interoperability problems [GPFLC04] are applied. We adopt the basic steps from the Uschold and Kings method [UG96], which consists of the following phases: (i) purpose identification, (ii) ontology capture, and (iii) coding and integrating. In their original forms, these phases do not precisely describe the reuse of existing ontologies and hierarchical conceptual models.

### 2.3.3 Matching Database to Ontology

The integration of between ontology and database has been a new problem since the booming ontology. Dejing [DLKQ06] did a research to interactively integrate heterogeneous systems. The study used ontologies to incorporate database schemata. Such ontologies were expressive by first-order ontology language to define the structures, semantics, and mappings of data resources. Bizid [BFBY14] designed a conceptual framework using Geo-web services and an ontology-based data integration approach to support the heterogeneity. The study transformed spatial databases into local ontology (RDF). The spatial information was exchanged by global ontology.

The goal of information integration in humanitarian aid information systems is to integrate the systems across the boundary of agencies. Main problems of matching are related to the identification of semantically related information. There is an expressive gap between database schemata and ontologies. In this research, we adjust existing matching techniques to integrate among relational databases (RDBs) through a pivot ontology. In fact, humanitarian aid information systems are pre-existing and have been developed independently on RDBs.

## 2.4 Conclusion

The use of ontologies can be used for information sharing in heterogeneous sources. This chapter describes the problems of humanitarian information and points out the interoperability issues in humanitarian aid domain. Ontology is a possible solution for interoperability issues. Ontology approach enables us to perform ontology construction, and information integration. We reviewed the ontologies for humanitarian aid, the methods to build ontology, and the researches for information integration.

# Chapter 3

# Ontology Construction Method

## 3.1 Pivot Ontology Framework

Our idea of the pivot ontology construction will fulfill the requirements for heterogeneous information in humanitarian aid domain. As mentioned above, a system of humanitarian aid in emergencies often cooperates with information in diverse domains. The difficulty is further complicated when independent databases are accessed across organizations, where full semantic knowledge of the component databases is most likely not available [NO95]. Information from autonomous databases can possibly have similar meanings but appears structural difference in different databases. To cooperate with this information, semantic conflict is, however, often problematic that leads to mismatching integration when information crosses from one database to another.

Ontology enables to reuse and share domain knowledge using a common vocabulary across heterogeneous application. The definition of ontology ([PKR97, UJ99]) is a hierarchically structured set of terms and some specification of their meaning in order to define a structure on the domain and constrain the possible interpretations of terms that can be used as a skeletal foundation for a knowledge base.

What is required, in fact, is not a combination of all application-dependent knowledge, but rather a major role of the pivot ontology is to encourage very broad semantic interoperability among domain knowledge and reduce the complexity of direct matching between several sources. Fig.3.1 is a pivot ontology model. Obviously, if we compare to

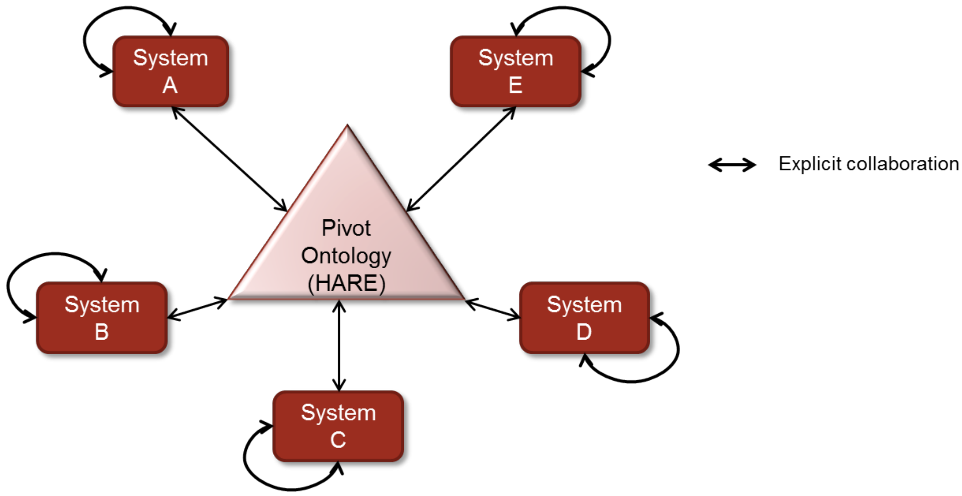



Figure 3.1: Pivot Ontology Model

Fig.2.1, when using a pivot ontology the number of matching is reduced. There are three strategies for building ontology[G07, Pin09] as follows:

- Application dependent: the ontology is built on the basis of an existing application.
- Application semi-dependent: it starts with concrete scenarios of future ontology usage as part of their specification activity.
- Application independent: the process is totally independent of the uses to which the ontology will be put in knowledge-based systems, agents, etc.

As Fig. 3.2, the ontology that created by application-independent strategy is more reusable than built by other strategies. Concepts inside a pivot ontology would be gener-

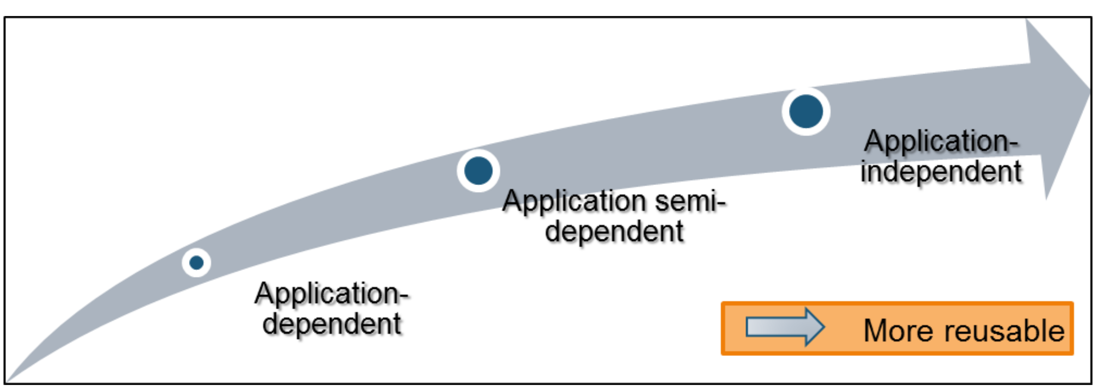



Figure 3.2: Strategies for building ontology

ally designed for broad systems in humanitarian aid domain. The pivot ontology should generally be designed for crossing the independent databases and composed of the upper ontologies, which describe very general concepts extracted from a knowledge source of international standards (e.g., the United Nations High Commissioner for Refugees (UNHCR) handbooks [PS07, PtPS03, Div97], and Sphere handbook [Pro11]). Relevant information from the knowledge source would then be conveyed to unambiguous relationships of concepts in the pivot ontology.

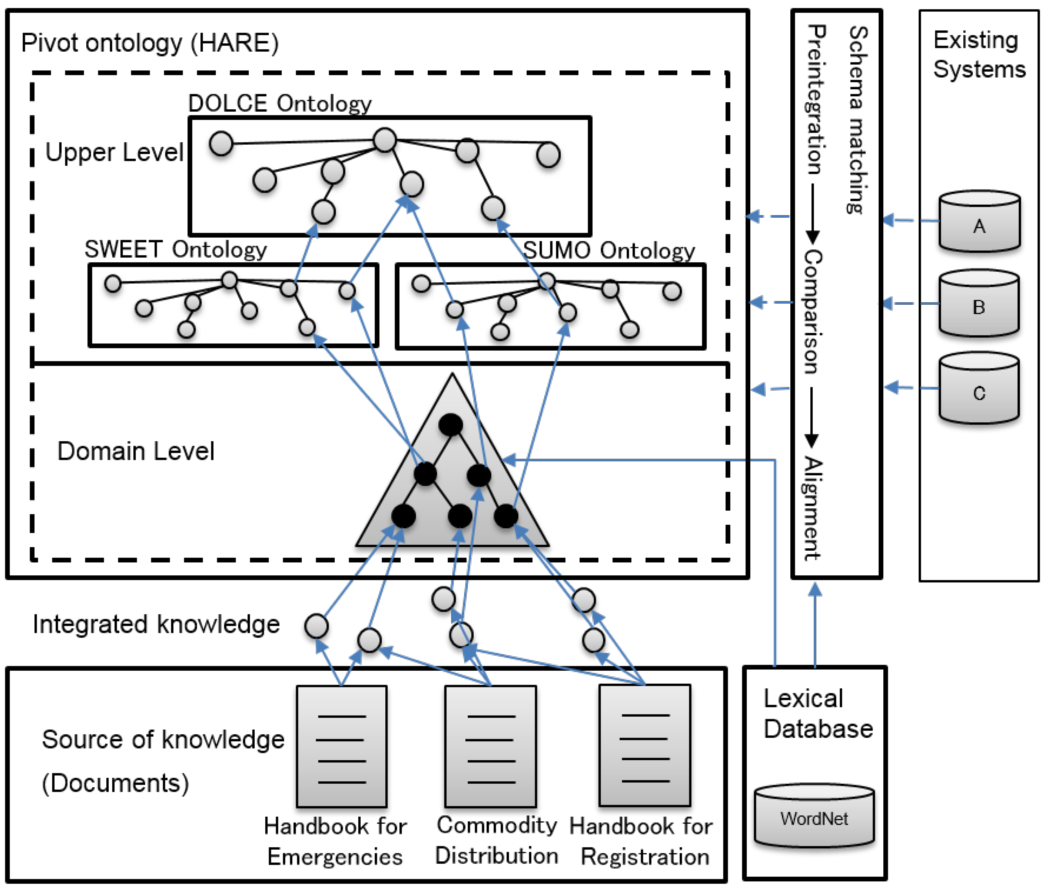



Figure 3.3: Pivot Ontology Framework

In order to be able to capture the concepts of pivot ontology, the knowledge source's core concepts and the upper ontologies' broad concepts would be integrated into a single ontology. As depicted in Fig.3.3, knowledge sources are consumed in the domain ontology. Domain ontology will be classified by exploiting the generic concepts of upper ontologies. Because the upper ontologies represent very broad concepts, the domain ontology cannot be seamlessly generalized by the upper ontologies. WordNet[1] has been chosen to reconcile with domain ontology and upper ontologies. Additionally, the pivot ontology serves a role in matching with exiting systems using PivotOntology-to-database schema matching approach for database integration.

## 3.2 Ontology Construction

Preliminary criteria for ontology design are clarity, coherence, and extensibility[Gru95].

1. Clarity: an ontology should be defined concepts by the intended meaning from social situations. Logical axioms are usually stated and all definitions should be documented with natural language.

2. Coherence: ontology concepts are related to each other in various ways. The coherence relation should be consistent with the definition of concept.

3. Extensibility: an ontology should be extended for vocabulary sharing or can be reused by other ontologies. The another ontology is able to define new vocabularies based on the existing vocabulary without modification of the existing definitions.

We focus on the techniques for reusing existing ontologies and semantic hierarchical conceptual model to design a common ontology (HARE ontology) for all humanitarian aid information sources. For our method, an ontology development process in the ontology engineering is selected as a direction for solving the interoperability problems [GPFLC04].

Three generic ontology architectures have been designed for the requirements of interoperability, namely a monolithic ontology architecture, a meshed ontology architecture, and a hierarchical ontology architecture[Fuc09].

1. Monolithic ontology architecture: an ontology engineer can fully control the entire ontology. Using this architecture, only small-scale ontology is efficient for maintenance because of high coherence and high coupling in the ontology. However, the

[1] https://wordnet.princeton.edu/

medium and large-scale ontology are able to be maintained by multiple engineers. The consistency and reusing of the ontology are difficult to maintain.

2. Meshed ontology architecture: an ontology combines multiple sub-ontologies. Although, a medium-scale ontology can be built by this architecture. Each sub-ontology still has high coherence and high coupling between sub-ontologies. An ontology engineer can possibly be assigned the responsibility for an individual sub-ontology. The maintainability and extensibility are limited.

3. Hierarchical ontology architecture: it is similar to a meshed ontology architecture, i.e., an ontology engineer can possibly be assigned the responsibility for an individual sub-ontology, which has high coherence between individual sub-ontologies. This architecture enforces low coupling between sub-ontologies. A sub-ontology is only allowed to refer to its parent ontology, and the top of all parent ontologies is an upper ontology. Therefore, this architecture is maintainable, extensible, and suitable for large-scale ontology.

The pivot ontology need to support very broad semantic interoperability among domain knowledge and reduce the complexity of direct matching between several sources. The hierarchical ontology architecture approach offers a good structure of inter-organizational ontology engineering. Eventually, we apply this approach and adopt the basic steps from one of the ontology engineering named the Uschold and Kings method [UK95]. This method consists of the purpose identifying, ontology capture, and then coding and integrating. This method does not precisely describe about reusing of existing ontologies and hierarchical conceptual model. With this objective, the first three phases of this method are extended and tailored for the construction of the HARE ontology as follows [ANBI13, ANBI15]:

1. Identifying a purpose and scope

    (a) Getting requirements of refugees in emergencies

    (b) Creating the use case diagrams and use cases descriptions

2. Building the ontology

    (a) Ontology capture - considering knowledge models from the use case diagrams

    (b) Ontology coding and integrating

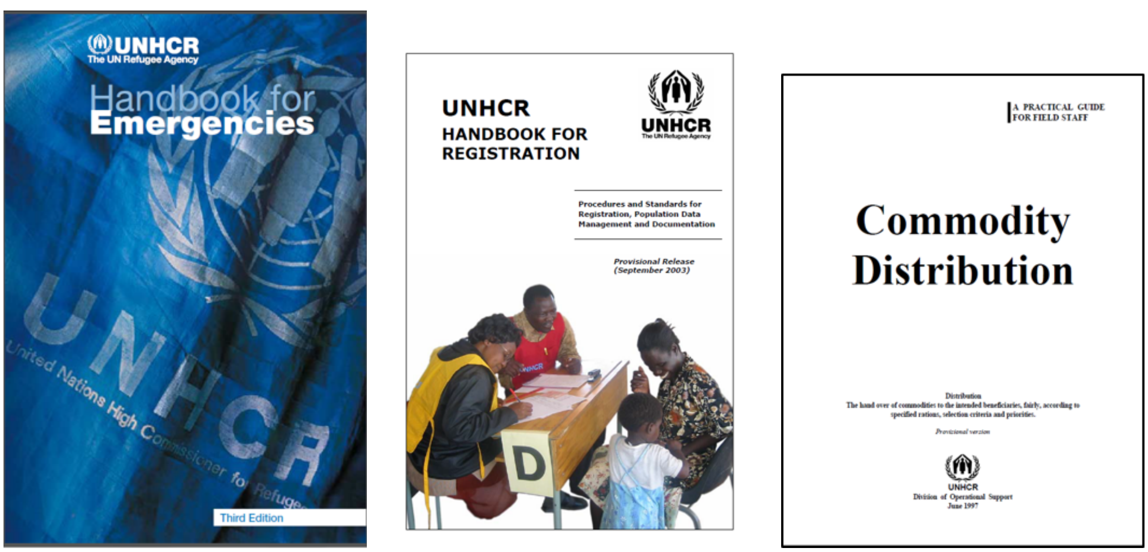
        i. Integrating with upper ontologies

        ii. Finding hypernyms of each concept to create hierarchical ontology

3. Evaluation - Verification with FaCT++, UNHCR handbooks, and existing schemata

## 3.3 Identify Purpose and Scope

**Getting requirements of refugees in emergencies** The HARE ontology is integrated domain knowledge from relevant chapters in the Handbook for Emergencies [PS07] (Fig.3.4a), which is a standard knowledge of international organization contained 15 chapters in 212 pages, and related documents [PtPS03, Div97] to undertake the abstraction and processes of refugee emergencies from UNHCR. The handbook for registration [PtPS03] (Fig.3.4b) contains 25 chapters in 325 pages, and the commodity distribution [Div97] (Fig.3.4c) contains six chapters in 77 pages.




(a) Emergency Handbook (b) Registration Handbook (c) Commodity Distribution

Figure 3.4: UNHCR Handbooks

The operations of UNHCR cover many areas in refugee emergencies, including health, food, sanitation and water, as well as key field activities corroborate the operations such as logistics, community services, and registration [Poj99]. Such operations must be managed and controlled by many associate organizations. In the getting requirements step, information should be extracted carefully from documentations. We determined the domain, scope and purpose of the operations into the refugee emergencies process models that depict the principle of refugee emergencies' operations.

**Process 1: Refugee Registration** This process is capturing the implemented processes of the refugee registration in a model. Refugees should be registered as fast as possible after reach to a refugee center. Refugee profiles must be the first information that organizations would like to know. The following information is recorded for verification of a person of concern: name, unique identifying registration number, date and place of birth, sex, existing identity documents, marital status, special protection and assistance needs, level of education, occupational skills, religion, language, household and family composition, date of arrival, current location and address, place of origin, and photograph. This information will be collected as the properties of Refugee profile concept [PS07, PtPS03].

**Process 2: Identification of Persons of Concern System** After the refugee registration process, if time permits, a pre-screening should take place at this stage to identify those who may not be a person of concern to UNHCR. The refugee profiles will be analyzed for the needs of refugees assessment. An accurately estimation of the number of refugee is a prerequisite for effective protection and assistance, and identification of beneficiaries including persons with special needs [PS07].

**Process 3: Emergency Planning System** The planning process is very important. Efforts should be made to design and construct a shelter as soon as possible. Several organizations must rely on this planning system. The project is the structural planning for the training, logistics, telecommunications, security, sites (camp, shelter), etc., to be better management, e.g., shelter management, non-food and food items' distribution [PS07].

**Process 4: Distribution of Assistance** There is a simple system to handle the distribution of assistance and the provision of service to refugees including emergency health care, distribution of food and non-food items. Many staffs from organizations participate in this system for sharing help to refugees [PS07].

**Process 5: Donation System** The Donation system is designed to receive, manage, and distribute a mass of donated goods and services to affected people. With the help of refugee communities, they identify the refugees into individuals and groups with their needs, especially unaccompanied and separated children. This system operates the distribution through an assistance system in order to support needs of refugees [PS07].

**Creating the use case diagrams and use case descriptions** The domain, scope and purposes of the identified operations are determined, and the Unified Modeling Language (UML) use case diagrams are developed for specifying typical user-visible functions of a humanitarian aid information system and for graphically representing and envisioning the relationships between use cases and actors [TA11]. The flows of interaction between actors and the system in each use case is specified using a textual use case description. Fig. 3.5 - 3.9 are the UML diagrams of the five principles of the HARE ontology in order to understand an extent to which interoperability can be supported.

## 3.4 Ontology Capture

The use case diagrams and use case descriptions obtained from the previous phase provide a source of requirements for establishing ontological conceptualization for developing the HARE ontology. Based on these diagrams and descriptions, concepts and relationships between them are identified and extracted. Resulting core concepts include Commodity, Distribution Cycle, Family, Household, Head of Family, Head of Household, Refugee, Registration Card, RefugeeActivity, RefugeeNeed, Person, Plan, Project, Organization,

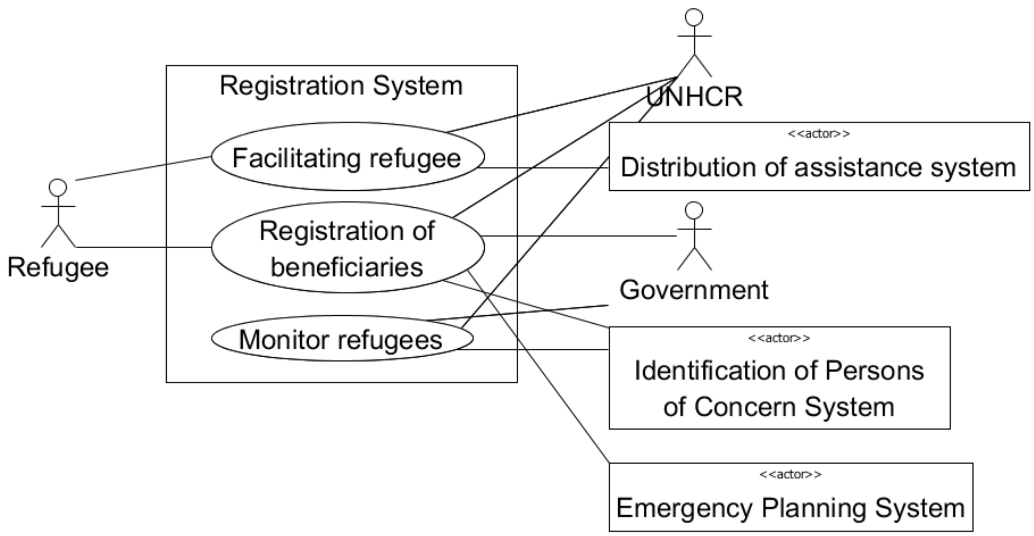



Figure 3.5: Registration System Diagram

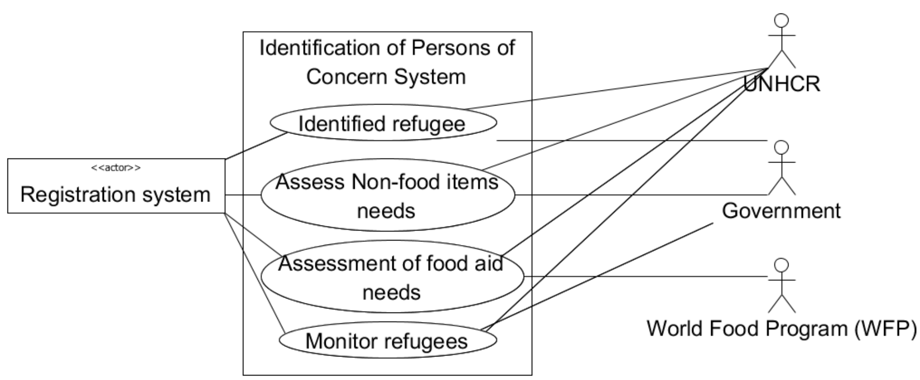



Figure 3.6: Identification of Persons of Concern System Diagram

and Staff. The concepts and relationships between elements are gathered in this step. Fig. 3.10 shows an example of concepts and their relationships, which reflect the real-world representation. A line in the picture is a property and a rectangle is a concept of the HARE ontology. The detailed relationships of Fig. 3.10 are represented in Table 3.1.

After the core concepts are defined, subclasses and disjoint decompositions are also identified; for example, a food product is identified as a specific type of Commodity.

The implementation of the HARE ontology requires an appropriate ontology editor and development environment. The Protégé development platform, which contains the Protégé-OWL ontology editor for the Semantic Web, is used in this research.

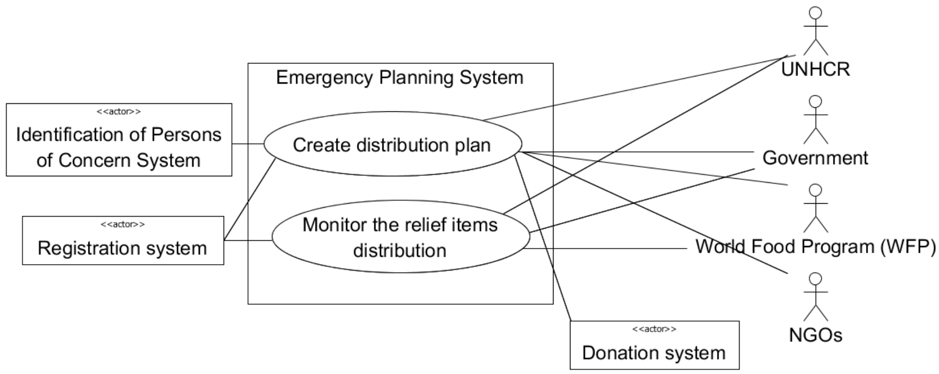



Figure 3.7: Emergency Planning System Diagram

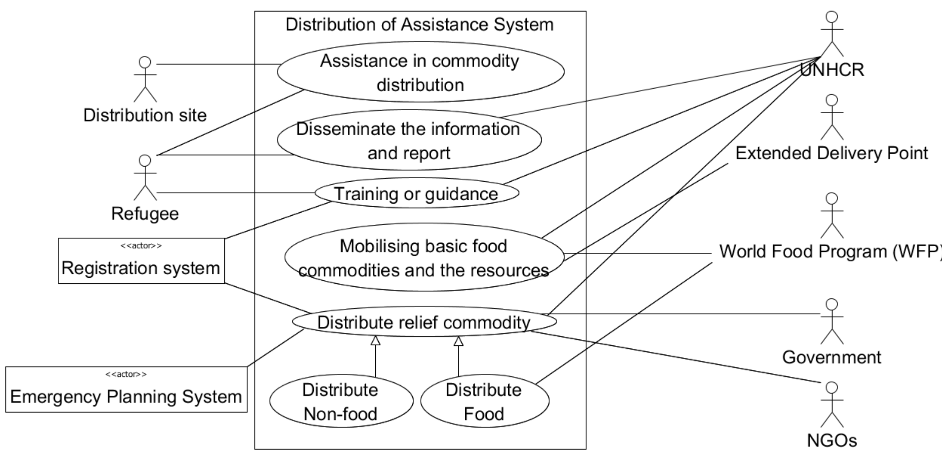



Figure 3.8: Distribution of Assistance System Diagram

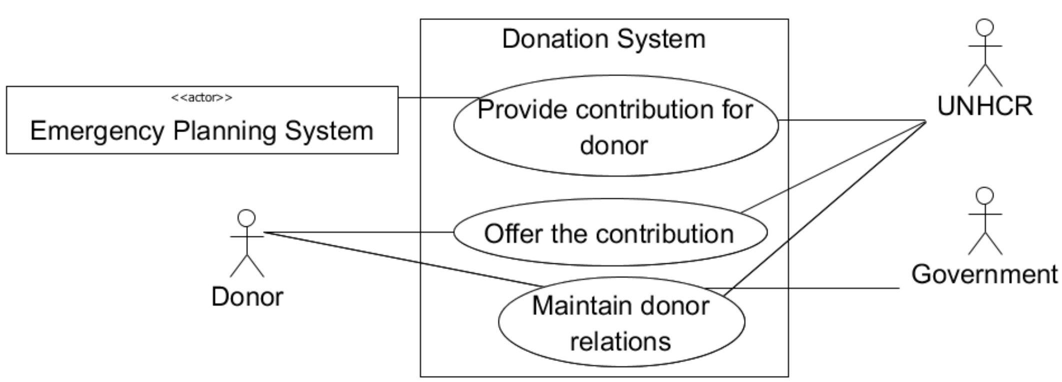



Figure 3.9: Donation System Diagram

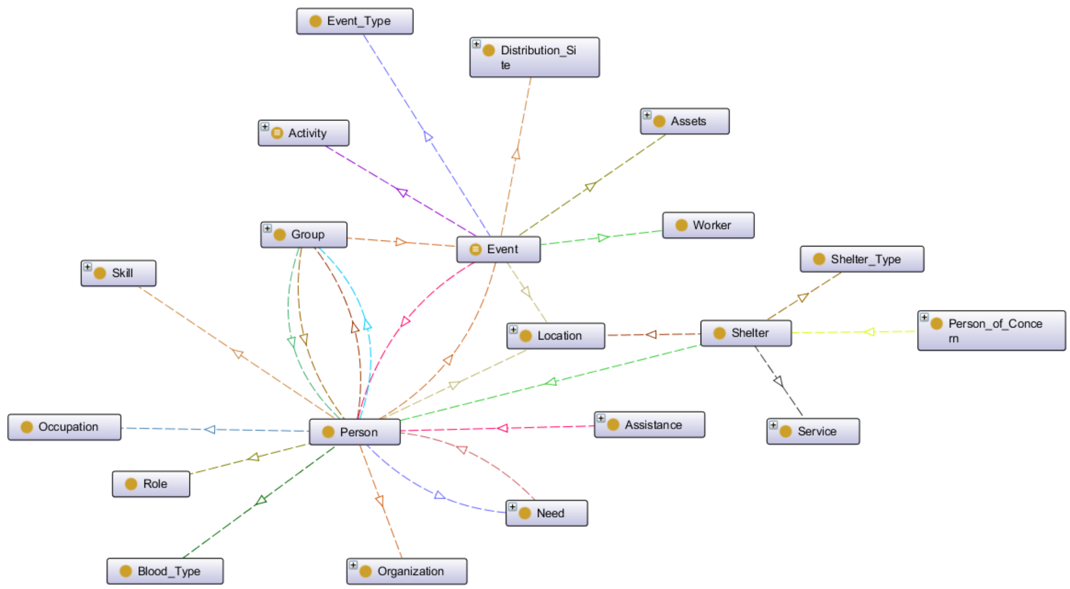



Figure 3.10: Example of concepts and their relationships

## 3.5 Coding and integrating

### 3.5.1 Integrating Upper Ontology

After creating the ontology, we would notice that there are some classes in the ontology that can be hierarchical implementation on upper ontologies. An upper ontology (a top-level ontology or a foundation ontology) can be also called as a top-level ontology or foundation ontology [Wik], which describes general common concepts for many knowledge domains and provides a mechanism for interoperation across domain-specific systems [GPFL10, TB08]. The concepts obtained from the previous phase are associated with more general concepts in three relevant upper ontologies, i.e., the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [DOL],[2] the Semantic Web for Earth and Environmental Terminology (SWEET) [RP05],[3] and the Suggested Upper Merged Ontology (SUMO) [SUM].[4]

[2] http://www.loa.istc.cnr.it/old/DOLCE.html
[3] http://sweet.jpl.nasa.gov/
[4] http://www.adampease.org/OP/

Table 3.1: Example of concepts and their relationships

| **Subject** | **Property** | **Object** |
|---|---|---|
| Event | has-event-type | Event_Type |
| Event | use-asset | Assets |
| Event | has-site | Distribution_Site |
| Event | participant | Person |
| Event | has-location | Location |
| Event | has-activity | Activity |
| Event | has-working-people | Worker |
| Person | has-location | Location |
| Person | participant-in | Event |
| Person | has-job-at | Organization |
| Person | has-role | Role |
| Person | request | Need |
| Person | has-occupation | Occupation |
| Person | is-member-of | Group |
| Person | has-blood-type | Blood_Type |
| Person | has-skill | Skill |
| Person_of_Concern | has-shelter | Shelter |
| Shelter | has-shelter-service | Service |
| Shelter | has-shelter-type | Shelter_Type |
| Shelter | has-location | Location |
| Shelter | has-shelter-member | Person |
| Group | has-member | Person |
| Group | participant-in | Event |
| Group | has-group-head | Person |
| Need | needed-by | Person |
| Assistance | is-responded-by | Person |

**Related upper ontologies**

- **Semantic Web for Earth and Environmental Terminology (SWEET)** The SWEET ontologies include thousand terms in the domain of geography and environments, spanning a wide scope of Earth system science, and related concepts (such as data characteristics). To support such a large collection and the guiding principles, the concepts are divided into orthogonal dimensions or facets in support of reductionism [RP05].

- **Suggested Upper Merged Ontology (SUMO)** It is the most outstanding proposal under consideration by the IEEE Standard Upper Ontology (SUO) working group. Its effort is to link categories and relations that come from different top level ontologies, e.g., the structural ontology containing relations for defining a proper ontology, and the unit-of-measure ontology providing definitions of standard unit systems in order to improve ontologies in the Semantic Web area [PNL02]. SUMO is based on the fundamental distinction between Physical and Abstract. The goal of SUMO is to develop a standard ontology that promotes data interoperability, information search and retrieval, automated inference, and natural language processing. SUMO is implemented in the first-order logic language SUO-KIF that can be automatically translated into OWL, although the translation is lossy. The ontologies that extend SUMO are available under GNU General Public License [SUM, MLR10].

- **Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)** DOLCE belongs to the WonderWeb library of foundational ontologies that provide a set of upper level concepts. It aims at capturing the ontological categories, and underlying natural language and human common sense that assist in making formed conceptualizations explicit. Furthermore, a combination of DOLCE has been published to improve some parts of DOLCE. DOLCE is based on the fundamental distinction between Endurants (i.e., objects or substances) and Perdurants (i.e., events or processes) [DOL].

The DOLCE has clear cognitive artifacts as an upper ontology. The SWEET ontology is a middle-level ontology which describes Earth and Environmental Terminology. The SUMO includes an upper ontology and also be extended with many domain ontologies. We found that the upper ontologies are able to be reused the concepts by the HARE ontology. A relation between the upper ontologies and HARE ontology is possibly established. Fig. 3.11 is an example of equality relation between them. We can conclude that Project should be an equivalent class of DOLCE:project, which is a subclass of DOLCE:plan. Plan should be an equivalent class of DOLCE:plan.

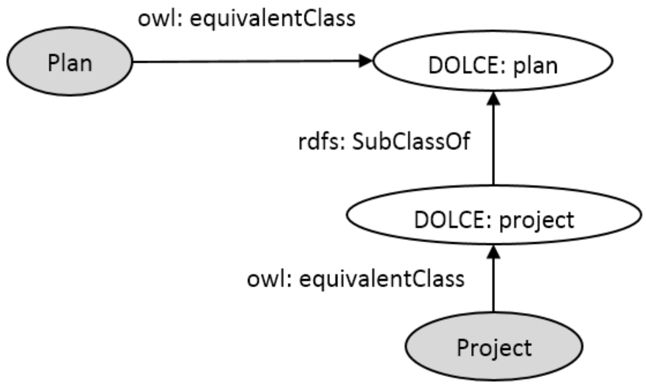



Figure 3.11: Ontology Generalization

An integration problem often arises when HARE concepts are generalized into general concepts in several upper ontologies. To address the integration among upper ontologies, priority levels of upper ontologies should be arranged. The DOLCE ontology is given the highest priority level because of their abstract concepts for cognitive ontological categorization. Both SUMO and SWEET partly comprise domain ontologies; however, SUMO contains more abstract concepts, whereas SWEET contains more specific concepts related to humanitarian aid. SUMO therefore takes priority over SWEET. Fig. 3.12 exemplifies the integration among DOLCE, SWEET, and HARE, where a grey oval represents a HARE concept.

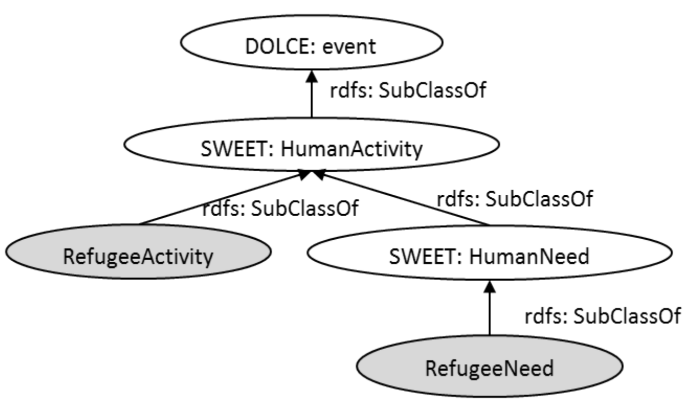



Figure 3.12: Ontology Generalization

### 3.5.2 Create lexical hierarchical ontology

Hypernym is a word or phase whose meaning includes the meaning of other words. A broad meaning of hypernym constitutes a category into which words with more specific meanings fall. Those core concepts are easiest to understand and interoperate when a hierarchy is diagrammed. For ease of understanding and interoperation, core HARE concepts are organized into a hierarchy by using word hyponyms from WordNet. WordNet can also be used for bridging the gap between HARE and upper ontologies. For instance, consider the core concepts 'Distribution_site' and 'Hospital'. Each of them is a material artifact, which is a top-level concept from DOLCE. More concrete representations of abstract concepts are required to connect top-level concepts defined in an upper ontology with the core concepts defined in the previous phase. Some WordNet concepts such as 'Medical_Building', 'Building', and 'Construction' are more specific than 'DOLCE:material_artifact'. Likewise, the 'Construction' concept is more general than 'Distribution_site', and the 'Medical_Building' concept is more general than 'Hospital'. Fig. 3.13 shows a generalization hierarchy of these concepts. After finding hypernyms of the entire concept, a concept that has general meaning same as the meaning of broad concept are grouped into a same category in hierarchy.

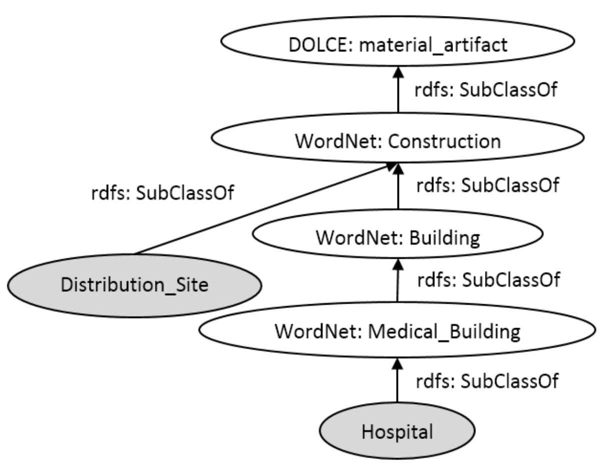



Figure 3.13: Ontology Generalization with WordNet

### 3.5.3 Axioms of the HARE ontology

The common ontology is usually contained concepts, relations, and axioms. In essence, an OWL ontology is a set of axioms. The large-scale ontology has been encountered the maintenance of the large sets of axioms. An axiom can describe classes, individuals, and properties. For example, the following axioms describe about four classes, i.e., Victim, Person, Disaster, and Event, and two property, i.e., participant-in, and has-location.

```
<owl:Class rdf:about="#Victim">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>

<owl:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="participant-in"/>
      <owl:someValuesFrom rdf:resource="#Event"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#Disaster">
  <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>

<owl:Class rdf:about="#Event">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="has-location"/>
      <owl:allValuesFrom rdf:resource="#Location"/>
```

```
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Above axioms mean every instance of Victim is also an instance of Person. Every instance of Person is a participant in some instances of Event. Every instance of Disaster is also an instance of Event. Every instance of Event has a location to instances of Location, respectively. These meaning is fixed through their semantics, which the axioms get the knowledge in the same way from all interpretations. For example, every Victim is a Person. There are no victim that is not a person.

## 3.6 Evaluation

We constructed the HARE ontology for humanitarian aid domain. The concepts compound of subclasses and their properties. For instance, Fig.3.14 presents the is-a relation of the material-artifact concept. That concept is classified into Asset, Vehicle, Document, Distribution_System, Commodity, Construction, Facility, and Donation_Item concept. These concepts contain more specific concepts such as Commodity concept has Food, and Non-food concept as their subclasses. The HARE ontology has been generalized sets of axioms from the DOLCE. A consistency checking of the HARE ontology has been operated from time to time by using FaCT++ reasoner.

Fig.3.15 shows examples of top-level concepts in the HARE ontology structure in the top level that built on top of HARE ontology. The lower layer is more specific concepts for humanitarian aid. Top level concepts provide very general concepts such as endurant (e.g., physical, non-physical endurant), perdurant (e.g., event, process, activity), quality, and abstract. Appendix 8.1 is concepts of the active HARE ontology included partly concepts of upper ontologies. In total, the HARE ontology has 446 elements (268 classes, 105 object properties, and 73 data properties), 90 of which are taken from the upper ontologies and

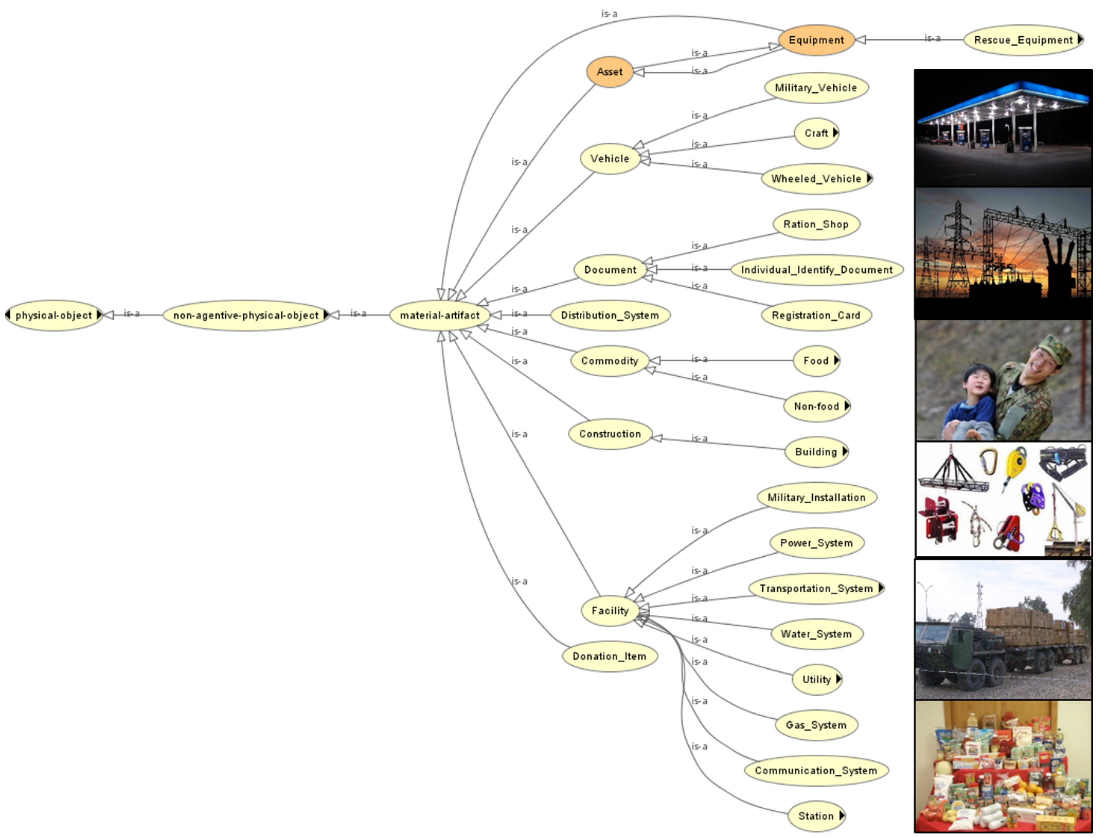



Figure 3.14: The is-a relation of material-artifact concept

a lexical database, i.e., 38 elements from DOLCE, 4 elements from SWEET, 7 elements from SUMO, and 41 elements from WordNet. The 182 remaining classes are bottom-level concepts in the HARE ontology (exclude classes from upper ontology).

### 3.6.1 Evaluation Techniques

For evaluation purposes, the coverage of the HARE ontology is evaluated with respect to comparison against a source of domain data, and comparison of compatibility against two existing schemas, i.e., Sahana [CDSDS+06, Sah], and Ushahidi [Ush], by using the schema matching technique described in Chapter 4. The evaluation details will be given in Chapter 5.

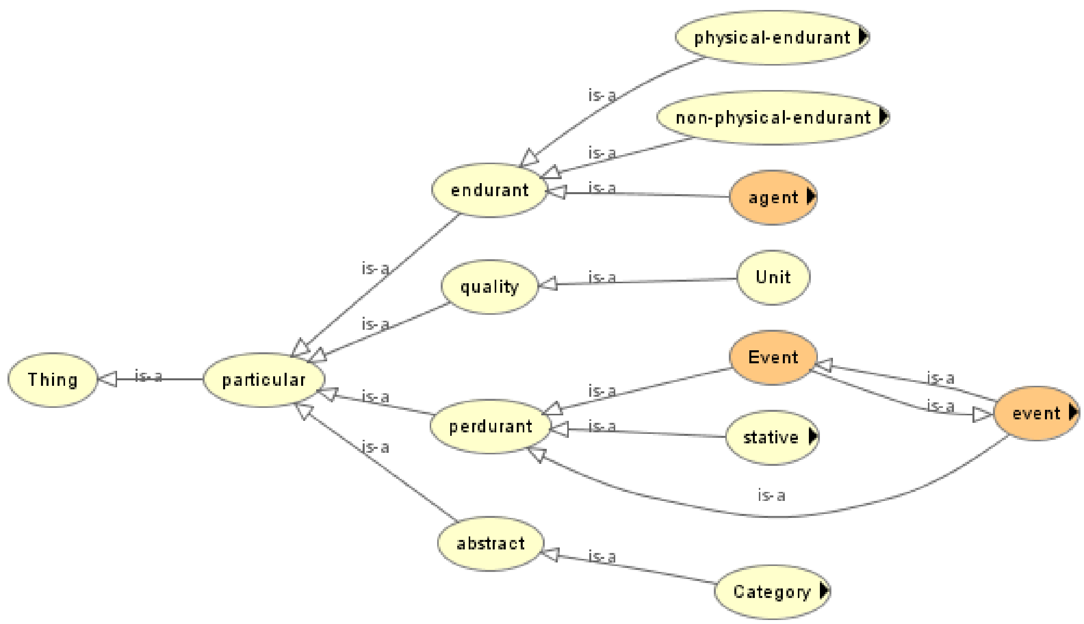



Figure 3.15: HARE Ontology Top Level

#### Comparison of HARE Ontology Against a Source of Domain Data

A Source of Domain Data is combining documents of UNHCR for guiding requirements of specific concepts. A corpus of documents has been indicated by a back-of-the-book index. The indexes of the selected sources from UNHCR contain 154 words. The classes are examined by the corpus. The 110 of 154 words are existed in bottom-level classes that is 71.43% from the corpus. On the other hand, the HARE ontology has 68 exceeding classes from the corpus that is 38.2% of the bottom-level elements and 61.8% of the bottom-level elements are existed in the corpus.

#### Comparison of compatibility against the existing schemata

The proposed ontology can be used as a common conceptualization of humanitarian aid that other related systems would be integrated. As the evaluation, we test the HARE ontology with two existing systems (Chapter 5). The aim of this evaluation is to present an example of interoperability between two systems with the proposed ontology using the schema matching (Chapter 4).

## 3.7 Conclusion

This chapter explains the pivot ontology construction model and the HARE ontology building method using ontology engineering methodology. To construct a common ontology for humanitarian aid domain, we provide the adapted method from the Uschold and King method for the pivot ontology construction. The handbooks from UNHCR are chosen to be sources of domain knowledge because the humanitarian aid processes of UNHCR are operated in world wide. We describe the ontology generalization in order to reuse the upper ontology. WordNet has been chosen to reconcile with HARE ontology and upper ontologies. When an existing system, which contains crisis information, needs to integrate its information, schema of the system will be integrated to the HARE ontology. An alignment will be returned as a matching result. Whenever a correspondence is found, a connection between the system and the HARE ontology is taken place. The pivot ontology model will become the interchange mediator to assist information integration among existing systems.

# Chapter 4

# Pivot Ontology-to-Database Matching

In this chapter presented some matching problems on the pivot model, which is explained in the previous chapter. Then, a matching model for PivotOntology-to-database matching is discussed. Finally, matching algorithms can be found in the last section.

## 4.1 The Matching Problems

The information in humanitarian aid information systems has to be integrated to enhance its efficiency. A matching is required for the pivot model. The matching model is an important model to integrate database schemata, which are matched by either manually or semi-automatically approaches. The humanitarian aid system is generally built on a relational database. A list of correspondences, which are relationships between one or more elements of one database schema to another database schema through pivot ontology, is an alignment that is returned as the matching result. Initially, a technical step of the matching is to identify correspondences between semantically related entities of a database schema and an ontology. Schema matching is the process of identifying correspondences. A database schema and an ontology are totally different in the structure. The schema of database and the structure of ontology have been designed independently.

The table and column names in a relational database schema are specified by database administrator. The names should be short and conveyed the meaning, but naming dose not have consistent conventions, e.g., the names sometimes have prefix or abbreviation, sometimes in the singular, sometimes in plural, sometimes use odd names, which do not reflect good meaning. Naming database schema is dissimilar from naming elements of the ontology. Naming elements must be conveyed the meaning to be clear understanding. Due to the use of different naming design and structure between ontology and database, the semantic interoperability issue is happened. In addition, the information sharing normally makes conflicts. Either Database-to-Database matching or ontology-to-ontology matching has similar conflicts occurred such as naming conflict, structural conflict, abstraction conflict [BLN86, NO95]. Obviously, Ontology-to-Database matching also has same conflicts, particularly abstraction conflict on the generalization. There is a big difference between an ontology and a database in the generalization hierarchy of concepts. For example, an ontological theory of the semantic linking has 'is-a' relationship to link in multi-level hierarchy as same as the generalization/specialization. An 'is-a' relationship in the database is a situation that exists between two tables that the relationship has only a single level of generalization.

To enhance information sharing for an emergency response, humanitarian information integration among diverse databases is necessary. Reconciliation of the structure and terminology of heterogeneous database schemas is required to solve a database schema integration problem. For database integration, a global schema is useful to eliminate duplication, avoid problems of multiple updates, and minimize inconsistencies across systems [BLN86]. A global schema requires establishment of explicit semantics and knowledge reuse. In knowledge engineering, the idea of ontology has been introduced to support wider usability of a knowledge base. In this research, the HARE ontology is employed as a global schema and a PivotOntology-to-Database schema matching methodology is designed for the database integration in the humanitarian aid domain. A matcher model, which consists of matching strategies with respect to the cross-system information sharing via the HARE ontology, is introduced in this research. This Chapter explains the

PivotOntology-to-Database schema matching model and matching algorithms.

## 4.2 Matcher Model

The HARE ontology contains general terminologies from WordNet and the three aforementioned upper ontologies. A relation between database schema elements and HARE concepts can be found using lexical matching and type of data matching. Our lexical matching method uses WordNet for finding synonyms and hyponyms in order to determine lexical entailment [HPnV06] between database schema elements and HARE concepts. Our PivotOntology-to-Database matching model is depicted in Fig. 4.1. Three ontology-to-database matching techniques, i.e., Class-Table, Property-Table, and Property-Column matching techniques, developed in our work [ANBI15] are adopted for ontology-to-database matching.

The matching model consists of two phases. The first phase consists of two different processes, i.e., Class-Table and Property-Table matching, running independently for finding correspondences between a given pivot ontology, say $PO$, and a given database, say $D$. An alignment is a set of correspondences obtained from each matching process. The resulting alignments, say Alignments $A1$ and $A2$, are aggregated into a combined alignment, say $M$. The second matching phase takes $M$, $PO$, and $D$ as input data for determining the final alignment, say $A$, using Property-Column matching.

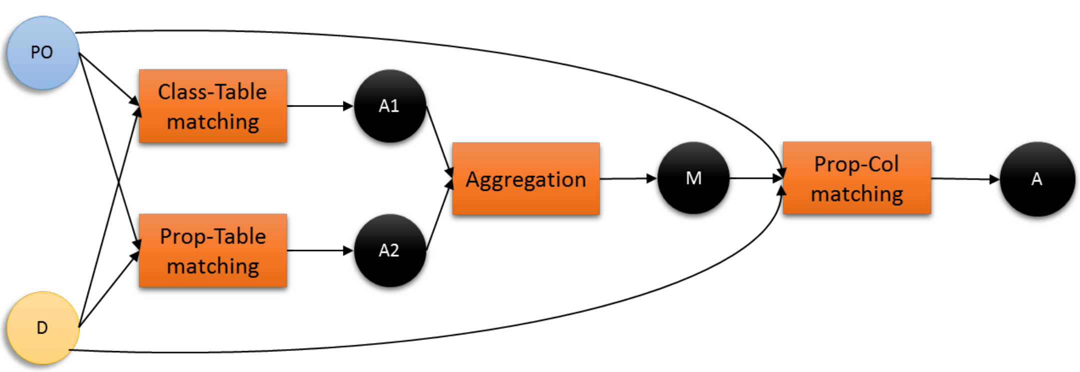



Figure 4.1: PivotOntology-Database matcher model

## 4.3 Matching Algorithms

Based on the correspondences between their elements (Class-Table, Property-Column, Property-Table correspondences [ANBI15]), a given relational database $D$ is matched against the pivot ontology $PO$ with the assistance of a domain expert using the following algorithms:

- Algorithm 1: The main structure for calling other algorithms and returning the output.
- Algorithm 2: Used for checking correspondences of junction tables against HARE concepts.
- Algorithm 3: Used for checking correspondences between HARE classes and database tables.
- Algorithm 4: Used for checking correspondences between HARE properties and table columns.

According the proposed model, to handle Class-Table, Property-Table, and Property-Column matching. There are four algorithms as follows: Algorithm 1 is a main structure for calling other algorithms. We expect the final result of Algorithm 1-4 as Alignment ($A$) that is a set of correspondences between Class-Table, Property-Table, and Property-Column. Algorithm 1 is used to checked for all Concepts ($C$) and Table ($T$) from top to bottom. A junction table is checked by Algorithm 2: Property-Table matching by sending Property of a class (Prop($C_i$)) and a table ($T_j$) to Algorithm 2. A junction table means a table that contains common attributes from two or more database tables. An alignment result from Algorithm 2 is kept in Alignment 2 ($A2$). Other tables are checked by Algorithm 3: Class-Table matching by sending a class ($C_i$) and a table ($T_j$) to Algorithm 3. An alignment result from Algorithm 3 is kept in Alignment 1 ($A1$).

Element-level matching techniques, e.g., string-based, linguistic resources, are applied for algorithm 3, algorithm 2, and algorithm 4. Structure-level matching techniques based

**Data:** $C$ is a concept in a given pivot ontology and $T$ is a table in a given database.
**Result:** Alignment $A$ between Concept $C$ and Table $T$

```
for (i=0;C_i exists;i++) do
    for (j=0;T_j exists;j++) do
        if (T_j is a junction table) then
            if (PropTableMatching(Prop(C_i),T_j)) then
                ADD(A2, corr(Prop(C), T_j));
            end
        else
            if (ClassTableMatching(C_i,T_j)) then
                ADD(A1, corr(C_i, T_j));
            end
        end
    end
    M = Merge(A1, A2);
end
for (i=0;M_i exist;i++) do
    A = PropColMatching(M_i(C),M_i(T));
end
Return A;
```

**Algorithm 1:** PivotOntology-Database matching

**Data:** $T$ is a junction table; $T0$ and $T1$ are referenced tables of ForeignKey($T$); $C$ is Domain(Prop($C$)); $C1$ is Range(Prop($C$)).

```
if (ClassTableMatching(C, T0))
AND (ClassTableMatching(C1, T1)) then
    Return true;
else
    Return false;
end
```

**Algorithm 2:** Property-Table matching

on internal structure, domains, ranges, foreign keys, and property types are applied in Algorithms 2 and 4. Type of property, and data type are checked in algorithm 4. Resulting alignments from Algorithms 2 and 3 are aggregated. Columns and their corresponding properties in the aggregation result are then checked by using Algorithm 4. Consequently, the final results are coming out as $A$.

```
if Synonym(T,C) then
 | Return true;
else if Hyponym(Synset(T),Synset(C)) then
 | Return true;
else
 | Return false;
end
```

**Algorithm 3:** Class-Table matching

```
for (i=0; Prop_i(C) exist; i++) do
 | for (j=0; Col_j(T) exist; j++) do
 | | if Datatype(Prop_i(C)), Col_j(T)) then
 | | | if Synonym(Prop_i(C)), Col_j(T)) then
 | | | | ADD(A, corr(Prop_i(C), Col_j(T));
 | | | else if Hypernym(Prop_i(C)), Col_j(T)) then
 | | | | ADD(A, corr(Prop_i(C), Col_j(T));
 | | | end
 | | end
 | end
end
Return A;
```

**Algorithm 4:** Property-Column matching

## 4.4 Element Correspondences

A correspondence between an element of the HARE ontology and that of a database is a 4-tuple

$$\langle id, e_1, e_2, CT \rangle,$$

where *id* is a unique identifier of the given correspondence, $e_1$ is an element, e.g., a table or a column, of the database, $e_2$ is an element, e.g., a class or a property, of HARE, and $CT$ is a correspondence type, which is one of the following: equality (=), overlapping ($\cap$), mismatch ($\perp$), more general/ hypernym ($\supseteq$), and more specific/ hyponym ($\subseteq$) [GSY09]. Methods for determining Class-Table correspondences, Property-Column correspondences, and Property-Table correspondences are described below.

### 4.4.1 Class-Table Correspondences

We adopt a linguistic approach to determine correspondences between classes and tables from schema information. The approach exploits linguistic properties of schema elements. We compare name strings for syntactical name matching and compare their meanings for semantic name matching. As a preparation step, names are cleaned by

1. changing uppercase letters to lowercase letters
2. removing special symbols
3. expanding abbreviations to full forms
4. replacing punctuations with spaces

Let $C$ be a concept in HARE and $T$ a table in a RDB. Let $name(C)$ and $name(T)$ denote the names of $C$ and $T$, respectively. A correspondence between $C$ and $T$ is determined by comparing $name(C)$ and $name(T)$ as follows:

1. *Syntactical name matching:* Construct a correspondence $\langle id, C, T, = \rangle$ if a word in $name(C)$ is the same as at least one word in $name(T)$.
2. *Semantic name matching:* Class-table correspondences are also determined based on semantic relationships such as synonym, hypernym, and hyponym relationships, given by WordNet.
   - Construct $\langle id, C, T, = \rangle$ if a word in $name(C)$ is a synonym of at least one word in $name(T)$.
   - Construct $\langle id, C, T, \cap \rangle$ if a word in $name(C)$ and a word in $name(T)$ have a common hypernym.
   - Construct $\langle id, C, T, \supseteq \rangle$ if a word in $name(C)$ is hypernym of at least one word in $name(T)$.
   - Construct $\langle id, C, T, \subseteq \rangle$ if a word in $name(C)$ is hypornym of at least one word in $name(T)$.

### 4.4.2 Property-Column Correspondences

The linguistic approach used earlier for determining Class-Table correspondences is also applied for determining correspondences between properties and columns. In addition, constraints on data types are also used; i.e., a non-foreign key column may correspond to only a data type property, and a foreign key column may correspond to only an object property.

### 4.4.3 Property-Table Correspondences

A correspondence may exist between a property and a junction table. A junction table is a database table that contains common columns from two or more other tables. A correspondence $\langle id, prop, T_J, =\rangle$ between a property *prop* and a junction table $T_J$ is constructed if $T_J$ is a bridge between a table $T_1$ and a table $T_2$, *prop* is an object property of a class $C_1$ with $C_2$ being its range, and $C_1$ and $C_2$ correspond to $T_1$ and $T_2$, respectively.

## 4.5 Conclusion

In this chapter, we have described the matching problems on PivotOntology-to-database matching which is a solution for solving interoperability in humanitarian aid domain. We propose a matcher model for matching a pivot ontology and a database. The matcher model consists of two phases for finding alignment between Class-Table, Property-Table, and Property-Column matching. Schema matching techniques such as element-level matching techniques, and structural-level matching techniques, are performed to find the correspondences between the schemata. Then, we focus on algorithms that follow the matcher model. We have presented the description of each technique for more seamless integration.

# Chapter 5

# Experiments

The main goal of the evaluation is threefold: (i) to investigate the PivotOntology-to-Database matching compared to direct matching without using the HARE ontology, (ii) to examine the compatibility of the HARE ontology against existing systems, and (iii) to explore a case study on database integration via the HARE ontology. In addition, semi-automatic matching is also investigated so as to reduce matching time for dealing with large-scale database schemata. Strategies for semi-automatic matching and combinations thereof are evaluated.

## 5.1 Manual Matching

The goal of this evaluation is to show how the HARE ontology works by comparing two values from two groups of experiments, i.e., the number of correspondences in the integration between two database schemata through the HARE ontology, and the number of those in the database integration without the HARE ontology. Ideally, the results of both groups should be close or equal. The information integration from correspondences among the encountering humanitarian aid databases is presented in this section. This research provides a basis for the pivot ontology of humanitarian aid in disaster management (HARE) and encourages the information sharing across existing databases. PivotOntology-to-Database matching appropriately facilitates finding correspondences between entities of the database and the HARE ontology.

Although showing the explicit collaboration typically needs integrated information from diverse individuals of an agent, the sample of integration with two open source disaster management systems is represented in this case study. The two open-source disaster management systems, i.e., Sahana [CDSDS$^{+}$06, Sah] and Ushahidi [Ush], are used in our case study. The two systems have different schemata and provide different features in the humanitarian aid domain.

1. Sahana Eden was developed by members of the Sri Lankan IT community, including experts in emergency and disaster management, and dedicated to helping people by providing information management solutions. It provides a number of different modules as follows;

   - Organization registry module allows organizations to record their office, warehouse, and field site information.
   - Project Tracking provides a tool to help organizations responding to disasters.
   - Shelter Management module provides functionality to list and to track information on shelters and on the people arriving and departing.
   - Scenario and Event modules help organizations to be better plan for disasters in different scenarios.
   - Human Resources module provides a tool to manage the people involved, such as staffs and volunteers working for different organizations.
   - Inventory module manages inventories of items and match requests for items with warehouses and other available facilities.
   - Assets module manages and tracks a wide range of assets needed to respond to disasters.
   - Assessment module collects and analyzes information to help organizations more effectively plan their emergency activities.
   - Map module has fully integrated mapping functionality.

2. Ushahidi was created by a development team from different countries, and dedicated to gathering crisis information to visualize disaster information on a map. It provides modules as follows;

   - Map module allows users to set the local location and layers of map.
   - Report module tracks disaster situations. User submits a report to provide the report information. A report consists of report title, description, date & time, categories of report, reporter information, location, external link, and photos.
   - Messaging module allows users to send a private message to another user.

The schema of Sahana is far larger than that of Ushahidi, i.e., Sahana contains 3,296 elements in 187 tables, while Ushahidi contains 388 elements in 53 tables. In the part of reconcilement ontology, the HARE ontology has 446 elements (268 classes, 105 object properties, and 73 data properties). The details of both database schema are shown in Appendix 8.2.

Both systems, they do not take into account the semantic interoperability. Although information integration between the two fewer compatible systems cannot get the high correspondences, the explicit information will be appeared when schemata are integrated through HARE ontology.

### 5.1.1 Design Intention of Experiment

| Scenario | Matching without HARE (Number of correspondences) |
|---|---|
| S↔U | Expected result |

| Scenario | Matching S↔U through HARE (Number of correspondences) |
|---|---|
| S↔H | Test Case 1 |
| U↔H | Test Case 2 |
| S↔H↔U | Test Case 3 |

Figure 5.1: The scenarios of manual matching experiment

The evaluation includes database integration scenarios, where an accuracy of pivot ontology model has been assessed by comparing with an expected result from direct matching between databases. In addition, each scenario is used for ontology compatibility assessment against each database. Another assessment is focused on a coverage of HARE ontology that has been explained in Chapter 3: Section 3.6. In this chapter, the ontology accuracy and compatibility are emphasized. The two groups of experiments are shown in Fig. 5.1, i.e.:

- Matching between Sahana and Ushahidi without using the HARE ontology, denoted by $\mathcal{S} \leftrightarrow \mathcal{U}$: The number of correspondences is regarded as an expected matching result.
- Matching between Sahana and Ushahidi through the HARE ontology: This group consists of 3 test cases:
  - Test Case 1: Matching between Sahana and HARE, denoted by $\mathcal{S} \leftrightarrow \mathcal{H}$.
  - Test Case 2: Matching between Ushahidi and HARE, denoted by $\mathcal{U} \leftrightarrow \mathcal{H}$.
  - Test Case 3: Analyzing the matching results of Test Cases 1 and 2 for finding common matching.

The assumption for this experiment are described in the following list:

- The result from test case 1 and 2 are the number of related correspondences that has possibility to match to other systems through HARE ontology.
- The result from test case 3 is the number of common correspondences from matching Sahana and Ushahidi through HARE ontology ($\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$).
- The result from test case 3 should be close or equal to expected result.

### 5.1.2 Matching Sahana ($\mathcal{S}$) and Ushahidi ($\mathcal{U}$) without HARE ontology ($\mathcal{H}$)

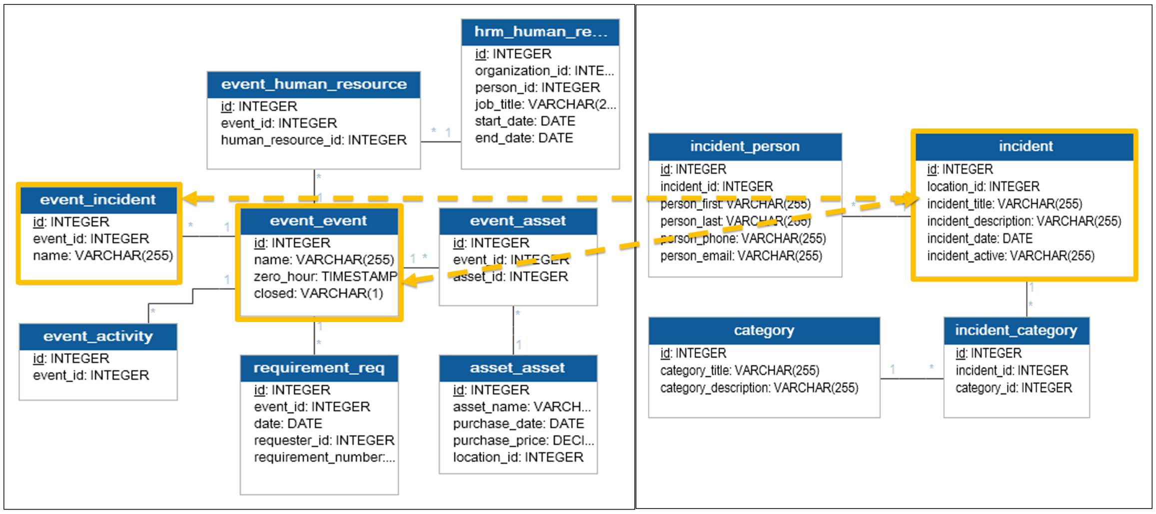
The expected result of direct matching would be a target answer for comparison with the test case 3 result. Sahana and Ushahidi database schemata are shown in the Appendix 8.2.

The schema matching techniques are used to match in element and structure levels of both database schemata. A number of correspondences between Sahana and Ushahidi is 13 (Table 5.1).

Table 5.1: Correspondences of Matching Sahana ($\mathcal{S}$) and Ushahidi ($\mathcal{U}$) without HARE ontology ($\mathcal{H}$)

| **Sahana** | | **Ushahidi** |
|---|---|---|
| event_incident | $\leftrightarrow$ | incident |
| event_event | $\leftrightarrow$ | incident |
| event_incident.name | $\leftrightarrow$ | incident.incident_title |
| event_event.zero_hour | $\leftrightarrow$ | incident.date |
| event_event.name | $\leftrightarrow$ | incident.incident_title |
| gis_location | $\leftrightarrow$ | location |
| gis_location.name | $\leftrightarrow$ | location.location_name |
| gis_location.lat | $\leftrightarrow$ | location.latitude |
| gis_location.lon | $\leftrightarrow$ | location.longitude |
| pr_physical_description | $\leftrightarrow$ | incident_person |
| pr_person | $\leftrightarrow$ | incident_person |
| pr_person.first_name | $\leftrightarrow$ | incident_person.person_first |
| pr_person.last_name | $\leftrightarrow$ | incident_person.person_last |

We present an example of the matching in Fig 5.2. Left of the figure is a sample database schema of Sahana and right of the figure is Ushahidi's a sample database schema. The schema of Sahana has eight tables, namely, event_event, event_incident, event_human_resource, hrm_human_resource, event_asset, event_activity, requirement_req, and asset_asset. These tables store event information that contains incidents in an event, staffs working in an event, assets used in an event, requirements required in an event, and activities operated in an event. The schema of Ushahidi has four tables, namely, incident_person, incident, incident_category, and category. These tables store incident information that contains people in an incident, and incident category. We found two correspondences from the sample schema that shown in Fig 5.2. This figure has two dotted lines represented two correspondences. These correspondences have been realized because the 'event' is hypernym of incident, i.e., Sahana:event_incident $\leftrightarrow$ Ushahidi: incident, Sahana: event_event $\leftrightarrow$ Ushahidi: incident. One table from Ushahidi has a correspondence to two tables of Sahana.




UML Diagram of Sahana Database Schema　　UML Diagram of Ushahidi Database Schema

Figure 5.2: Example of the matched tables between Sahana and Ushahidi using direct matching

### 5.1.3 Matching between Sahana and Ushahidi through HARE ontology ($\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$)

**Test case 1: Matching between Sahana and HARE ontology ($\mathcal{S} \leftrightarrow \mathcal{H}$)**

The event concept is a core concept in disaster management. This example would like to be clarified the information sharing through pivot ontology. The HARE ontology is designed to facilitate interoperability among existing humanitarian aid databases by providing broad humanitarian aid vocabularies and their relationships. An event information is described by the HARE ontology. Each event uses an asset. An event has related activities in a particular time period. The activity concept is a subclass of the event concept. An activity contains the information to identify its location. Each of an event, an activity, and a location has a name as a data type property. A location is identified by its latitude, longitude, and address. In the portion of the Sahana database illustrated in Fig. 5.3, an event has many activities at a location in a particular period. Assets used by an event are kept in a location. An activity belongs to a project. Events, activities, and locations have names as their attributes. A location is identified by a latitude, a longitude, a street, and a postcode.

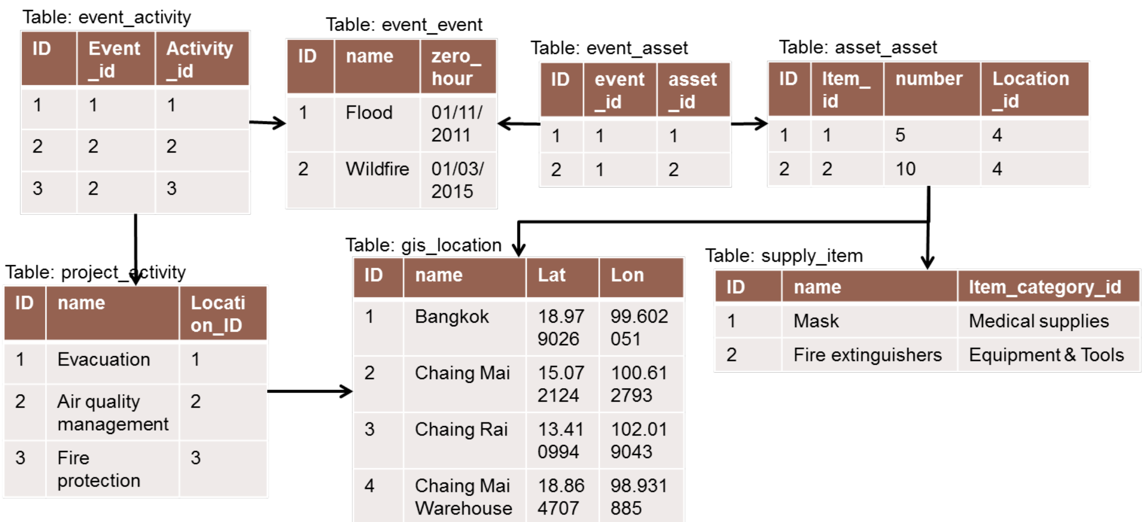



Figure 5.3: Example of Sahana schema

**Alignment Result between Sahana and HARE** The alignment resulting from matching Sahana with HARE consists of 107 correspondences, some of which are shown in Table 5.2. According to this table, information in Sahana can be shared with other systems through three HARE concepts, i.e., an event, an activity, and a location. Fig. 5.4 depicts information and schema integration between Sahana and HARE. A red oval represents a concept in the HARE ontology. A label on a line represents a property of a concept. A brown rectangle denotes information embedded in HARE after matching. Properties of the event concept are inherited through the 'is-a' relation to the activity concept. As a result, an activity also has the 'has-location' property.

Table 5.2: Example of an alignment between Sahana and HARE

| Correspondence pairs | |
|---|---|
| **Sahana** | **HARE** |
| event_event.name | Event.has-event-name |
| event_event.zero_hour | Event.has-start-date |
| event_activity | Event.has-activity.Activity |
| project_activity.name | Activity.has-activity-name |
| project_activity.location_id | Activity.has-location.Location |
| gis_location.name | Location.has-location-name |
| gis_location.lat | Location.has-latitude |
| event_asset | use-asset |

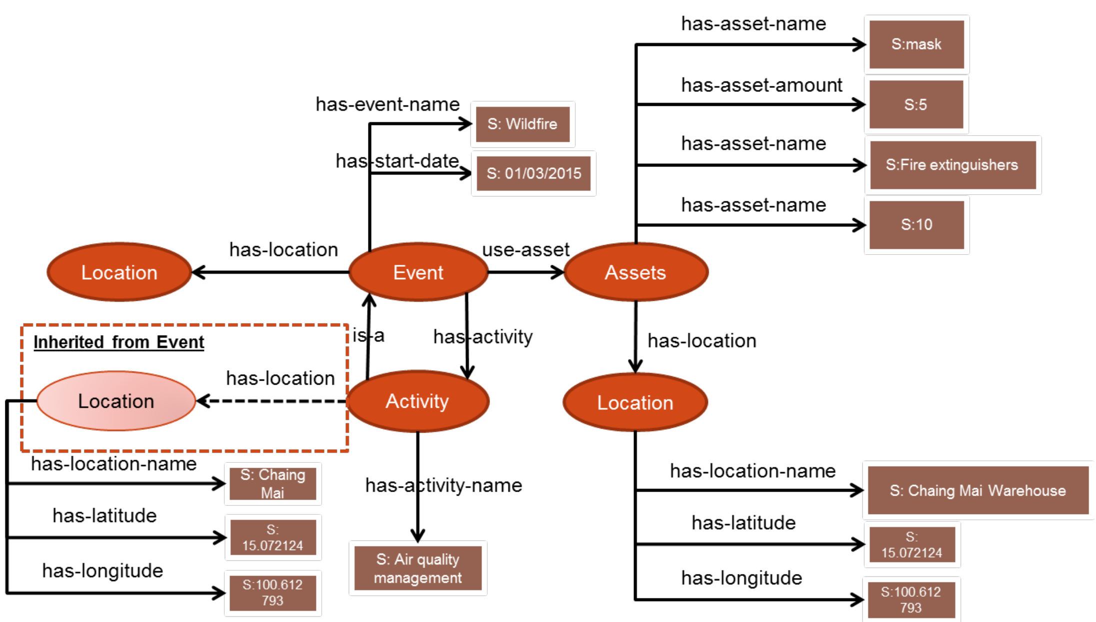



Figure 5.4: Sahana and HARE matching

### Test case 2: Matching between Ushahidi and HARE ontology ($\mathcal{U} \leftrightarrow \mathcal{H}$)

In the Ushahidi database illustrated in Fig. 5.5, an incident person will be recorded to identify the incident in which he/she participates. An incident contains the information to identify its location. Incidents, incident persons, and locations have names as their attributes. A location is identified by its latitude and longitude. Fig. 5.5 also shows information of two incident people, their incidents encountered, and the location of these incidents.

Table: incident_person

| ID | First | Last | Incident_id |
|---|---|---|---|
| 1 | Peter | Galen | 2 |
| 2 | Steve | Nash | 2 |

Table: incident

| ID | title | Incident_date | Location_id |
|---|---|---|---|
| 1 | Flood | 01/11/2011 | 1 |
| 2 | Haze | 01/03/2015 | 2 |

Table: location

| ID | latitude | longitude | name |
|---|---|---|---|
| 1 | 8.276727 | 99.206543 | Bangkok |
| 2 | 16.045813 | 103.557129 | ChaingMai |

Figure 5.5: Example of Ushahidi schema

We also consider the possibility of generalizing elements for interoperability. Linguistic relations are employed for database matching. A linguistic resource, i.e., WordNet, is used for finding linguistic relations, e.g., synonyms, hypernyms, hyponyms, and equality.

**Alignment Result between Ushahidi and HARE** The alignment result consists of 24 correspondences. Table 5.3 shows examples of some correspondences and their relations. For instance, the concept incident is more specific than the concept event. An implication is that information about an incident can be represented by an event, but some information concerning an event may be not represented by an incident. Similarly, the relation between Incident_Person and Person is also hyponym. The information of Incident_Person can be entirely shared to Person concept in HARE ontology because Incident_Person is a person who is falling in serious event. Fig. 5.6 depicts information and schema integration between Ushahidi and HARE.

Table 5.3: Example of an alignment between Ushahidi and HARE

| Correspondence pairs | | Relation |
|---|---|---|
| Ushahidi | HARE | |
| incident | Event | Hyponym |
| incident_person | Person | Hyponym |

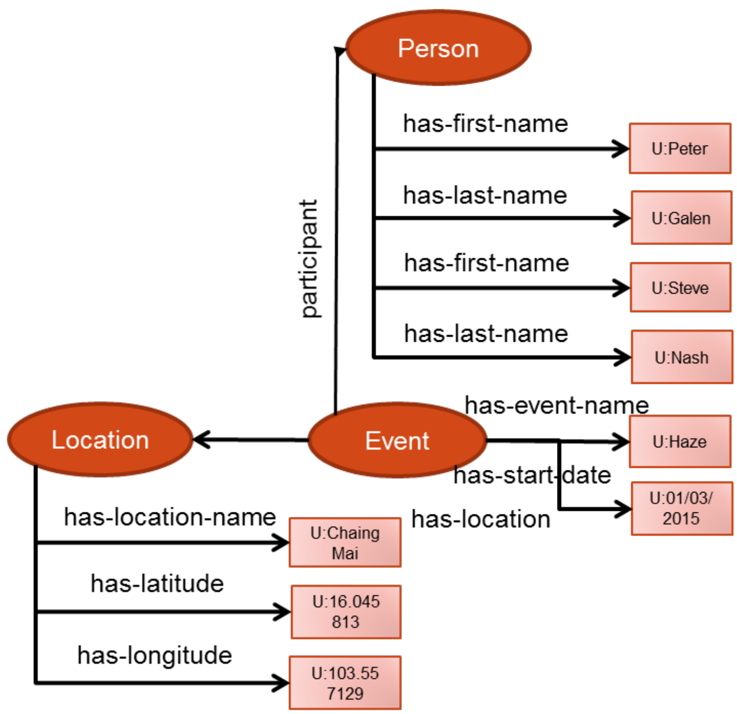



Figure 5.6: Ushahidi and HARE matching

**Test case 3: Analyzing matching result from test case 1 and 2 for finding the final result**

The result of test case 3 is derived from aggregation of the result of test case 1 and 2. Fig. 5.7 is depicted the aggregated result and yellow highlight rows are selected to be the test case 3 result because these rows have been completely correspondent by both a concept of Sahana and a concept of Ushahidi through a HARE concept. Therefore, test case 3 has 13 as a result.

| SAHANA | HARE | USHAHIDI |
|---|---|---|
| event_incident | Event | incident |
| event_event | Event | incident |
| event_incident.name | Event.has-event-name | incident.incident_title |
| event_event.name | Event.has-event-name | incident.incident_title |
| event_event.zero_hour | Event.has-start-date | incident.incident_date |
| event_activity | Event.has-activity.Activity | – |
| event_asset | Event.use-asset | – |
| event_asset.asset_id | Event.use-asset.Asset | – |
| – | Event.has-event-type.Event_Type | incident_category |
| – | Event.has-location.Location | incident.location_id |
| – | Event.has-start-date | incident.incident_date |
| gis_location | Location | location |
| – | Location.has-country.Country | location.country_id |
| gis_location.name | Location.has-location-name | location.location_name |
| gis_location.lat | Location.has-latitude | location.latitude |
| – | Location.has-location-name | location.location_name |
| gis_location.lon | Location.has-longitude | location.longitude |
| pr_physical_description | Person | incident_person |
| pr_person | Person | incident_person |
| pr_physical_description.blood_type | Person.has-blood-type | – |
| pr_physical_description.eye_color | Person.has-eye-color | – |
| pr_person.first_name | Person.has-first-name | incident_person.person_first |
| pr_person.gender | Person.has-gender | – |
| pr_physical_description.hair_color | Person.has-hair-color | – |
| pr_physical_description.hair_length | Person.has-hair-length | – |
| pr_physical_description.hair_style | Person.has-hair-style | – |
| pr_physical_description.height | Person.has-height | – |
| hrm_human_resource | Person.has-job-at.Organization | – |
| pr_person.last_name | Person.has-last-name | incident_person.person_last |
| pr_physical_description.skin_marks | Person.has-masks | – |

Figure 5.7: Aggregated Matching Result from Test Case 1 and 2

## Experiment Result of Manual Matching

| Scenario | Matching without HARE (Number of correspondences) |
|---|---|
| S↔U | 13 |

=

| Scenario | | Matching S↔U through HARE (Number of correspondences) |
|---|---|---|
| S↔H | TC:1 | 107 |
| U↔H | TC:2 | 24 |
| S↔H↔U | TC:3 | 13 |

Figure 5.8: The experiment results for manual matching

Table 5.8 shows the results of manual matching. The number of correspondences in Test Case 3, i.e., $\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$, is equal to that of direct matching $\mathcal{S} \leftrightarrow \mathcal{U}$ without using HARE. However, the use of HARE greatly extends the possibility of integration and fusion of information in Sahana and Ushahidi. The 13 common correspondences obtained from Test Case 3 provide a bridge connecting the correspondences in Test Case 1 and those in Test Case 2, i.e., correspondences in Test Cases 1 and 2 can be joined using these common HARE-based correspondences. Table 5.4 is examples of correspondences that mean an attribute ‘name’ from a table ‘event_incident’ in Sahana can be matched with an attribute ‘incident_title’ from a table ‘incident’ in Ushahidi through a concept ‘event’ in HARE. Information from Sahana and Ushahidi flows through such join operations. Without using HARE, the possibility of joining Sahana elements to Ushahidi elements is limited.

Table 5.4: Example of the result from test case 3: correspondences of matching Sahana and Ushahidi through HARE ontology

| $\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$ |
|---|
| *event_incident.name* ↔ Event.has-event-name ↔ incident.incident_title |
| *event_incident.zero_hour* ↔ Event.has-state-date ↔ incident.date |

### 5.1.4 Compatibility of HARE ontology

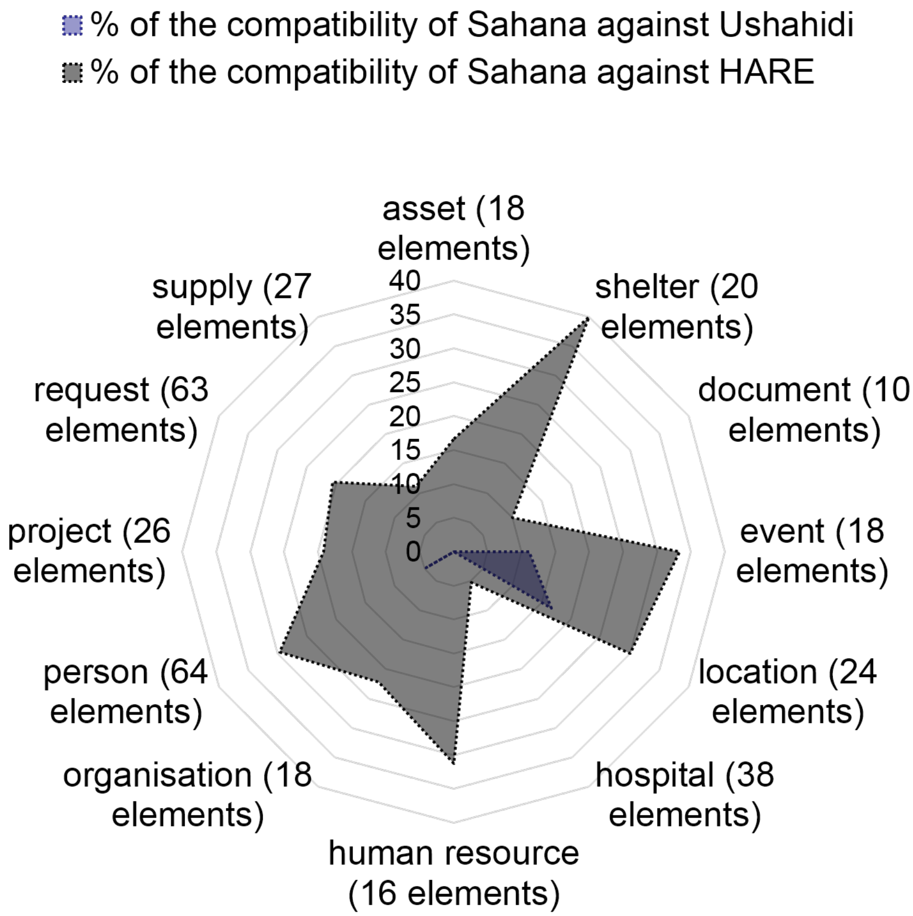



Figure 5.9: Compatibility of Sahana against Ushahidi and HARE

Fig. 5.9 depicts the percentage of compatibility of Sahana against Ushahidi and HARE. For information diversity, the HARE ontology has overcome Sahana. The labels above the axes of Fig. 5.9 show twelve categorized terms and the total number of Sahana elements in each categorized term. The categorized terms are asset, shelter, document, event, location, hospital, human resource, organisation, person, project, request, and supply. The black area represents the compatibility percentage of Sahana against HARE, while the purple area points out the compatibility percentage of Sahana against Ushahidi, i.e., 40% of Shelter elements in Sahana are compatible with HARE elements, which Sahana has totally 20 elements in Shelter-group elements. The purple area is apparently smaller than the black area, i.e., the compatibility of Sahana against HARE is higher than the compatibility of Sahana against Ushahidi. Compatibility of Sahana against HARE is higher than the compatibility of Sahana against Ushahidi. Usability of Sahana is increasing if the compatibility of Sahana against HARE is expanding.

Usability of Sahana increases as the compatibility of Sahana against HARE expands. The higher compatibility of a pivot ontology and anonymous databases can be achieved by ontology modification, which is a process in ontology engineering. An ontology is flexible and changeable under control of the main structure of the pivot ontology.

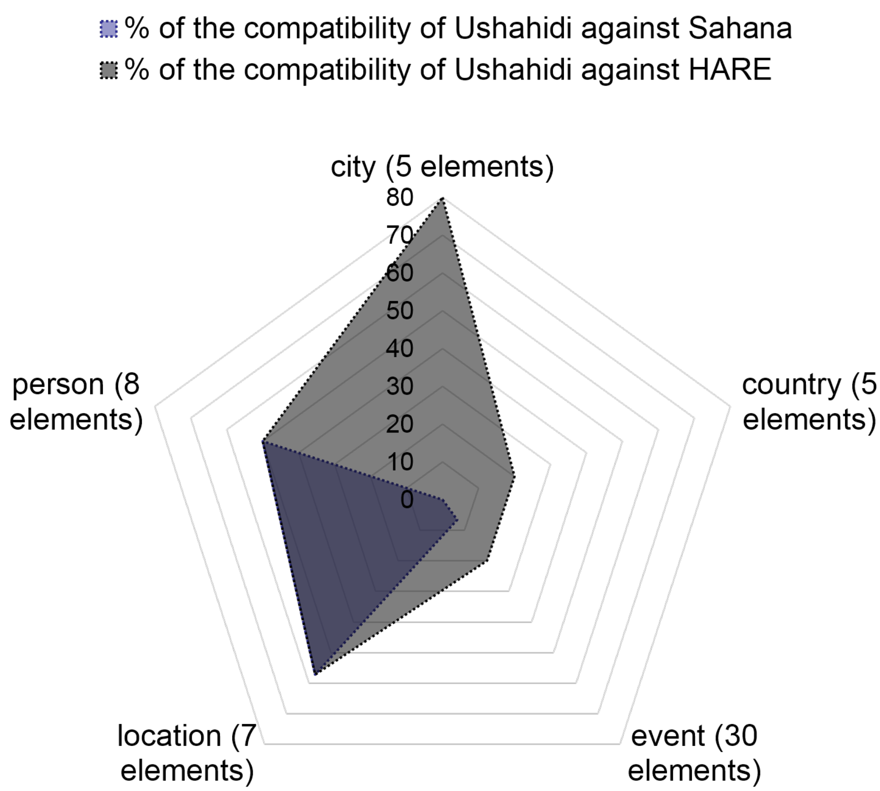



Figure 5.10: Compatibility of Ushahidi against Sahana and HARE

Fig. 5.10 depicts the compatibility percentage of Ushahidi against Sahana and HARE. For information diversity, the HARE ontology has overcome Ushahidi. There are five categorized terms, e.g., city, country, event, location, and person. The black area points out the compatibility percentage of Ushahidi against HARE. The purple area depicts the compatibility percentage of Ushahidi against Sahana. The black area includes the purple area, i.e., the compatibility of Ushahidi against HARE is higher than that of Ushahidi against Sahana. Usability of Ushahidi also increases as the compatibility of Ushahidi against HARE expands.

We notice that there are some common information in both schemata. There are two types of integrated information, i.e., the direct integrated information (common information and extended information), and the indirect integrated information (explicit knowledge). In this experiment, we focus only the direct integrated information. The indirect integrated information is out of scope in this research. Fig. 5.11 is represented the direct integrated information between Sahana and Ushahidi through HARE ontology. There are three zones in the figure, i.e., Sahana, Ushahidi, and common zones between

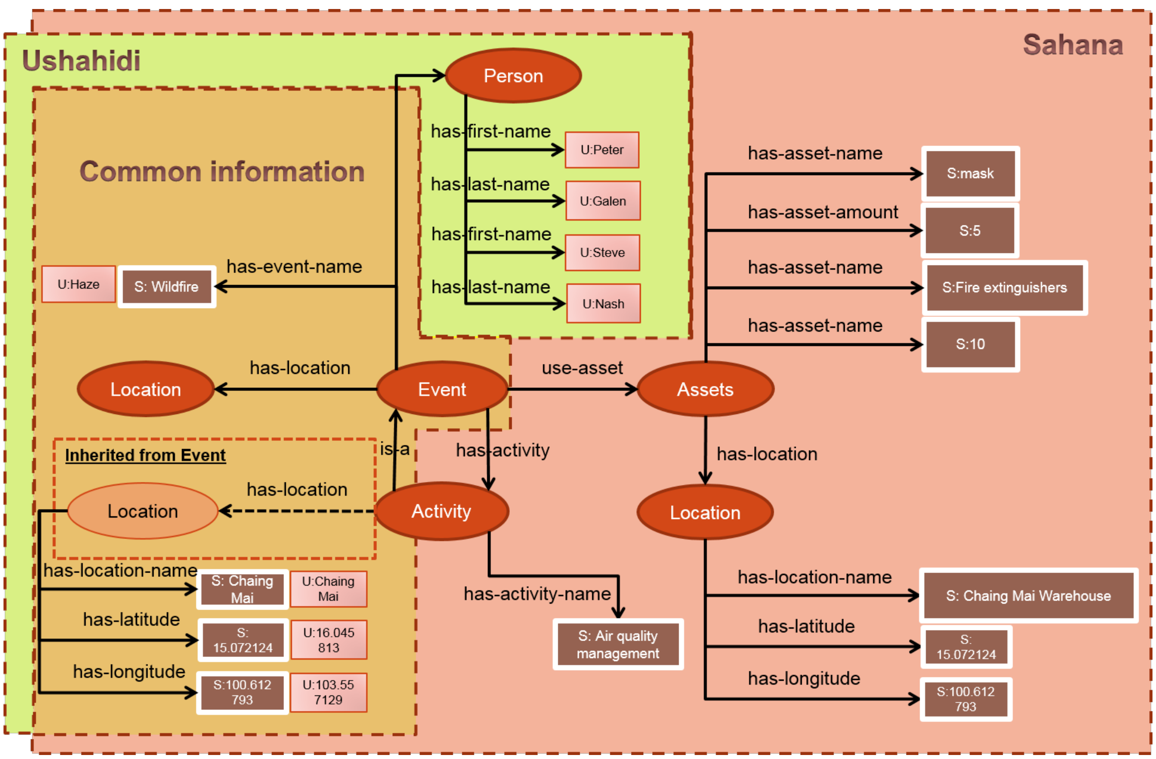



Figure 5.11: Information Sharing between Sahana and Ushahidi through HARE

Sahana and Ushahidi.

**Direct integrated information**, it combines two groups of information, i.e. common information, and extended information. The common information is derived from common correspondences between Sahana and Ushahidi through HARE ontology (Table.5.5). A

Table 5.5: Common correspondences between Sahana Eden and Ushahidi through HARE

| **HARE Concept** | $\mathcal{S} \leftrightarrow \mathcal{H}$ | $\mathcal{U} \leftrightarrow \mathcal{H}$ |
|---|---|---|
| Event | <1.1, event_incident, Event, hyponym><br><1.2, event_event, Event, equality><br><1.3, event_incident.name, Event.has-event-name, hyponym><br><1.4, event_event.zero_hour, Event.has-start-date, equality><br><1.5, event_event.name, Event.has-event-name, equality> | <2.1, incident, Event, hyponym><br><2.2, incident.incident_title, Event.has-event-name, hyponym><br><2.3, incident.date, Event.has-start-date, hyponym> |
| Location | <1.5, gis_location, Location, hyponym><br><1.6, gis_location.name, Location.has-name, hyponym><br><1.7, gis_location.lat, Location.has-latitude, hyponym><br><1.8, gis_location.lon, Location.has-longitude, hyponym> | <2.4, location, Location, equality><br><2.5, location.location_name, Location.has-name, equality><br><2.6, location.latitude, Location.has-latitude, equality><br><2.7, location.longitude, Location.has-longitude, equality> |
| Person | <1.9, pr_physical_description, Person, hyponym><br><1.10, pr_person, Person, equality><br><1.11, pr_person.first_name, Person.has-first-name, equality><br><1.12, pr_person.last_name, Person.has-last-name, equality> | <2.8, incident_person, Person, hyponym><br><2.9, incident_person.person_first, Person.has-first-name, hyponym><br><2.10, incident_person.person_last, Person.has-last-name, hyponym> |

correspondence between an element is represented by a 4-tuple as we mention in Chapter 5;

$$\langle id, e_1, e_2, CT \rangle,$$

- $e_1$ is an element, e.g., a table or a column, of the Sahana.
- $e_2$ is an element, e.g., a class or a property, of HARE.
- $\mathcal{S} \leftrightarrow \mathcal{H}$ is correspondences between Sahana and HARE.
- $\mathcal{U} \leftrightarrow \mathcal{H}$ is correspondences between Ushahidi and HARE.

There are three concepts of HARE ontology that can be matched from both systems, i.e., Event, Person, and Location. The common correspondences are making an initial connection between both systems. For example, Fig. 5.11, the common correspondences

$$\langle 1.3, event\_incident.name, Event.has-event-name, hyponym\rangle,$$

$$\langle 2.2, incident.incident\_title, Event.has-event-name, hyponym\rangle$$

are both matched with Event.has-event-name from HARE and we can assume that S:Chiang Mai is same as U:Chiang Mai, and S:Wildfire and U:Haze possibly be a same event with their matching name 'Chiang Mai' and their location name, approximate latitude and longitude. As extended information, the HARE ontology indicates that not only 'location' concept has relation to 'event' concept, but also 'person', 'activity', and 'asset' concept have relations to 'event' concept as well. Eventually, extended information has been linked, such as the result of a typical integration is an event name S:Wildfire or U:Haze has possibly two participants (U:Peter, U:Steve) that information comes from Ushahidi. The event has possibly activity named S:Air quality management that be extended from Sahana.

**Indirect integrated information** is to elicit information in the HARE ontology. The HARE ontology builds the semantic model for both systems. Domain knowledge from the existing system is incrementally externalized by the HARE ontology. The HARE ontology does not intentionally have to perfectly overlap the database schema of existing systems, but even so the compatibility results reveal that the derived correspondences scatter satisfactorily all over the schema.

## 5.2 Semi-automatic Matching

The evaluation described above is based on manual matching. Next, we exploit semi-automatic matching in the two groups previously used for manual matching (cf. Fig 5.1 in this Chapter) and investigate appropriate matching strategies. We measure the precision, recall, F-measure, and the number of correspondences to determine the effectiveness of the HARE ontology and combinations of matching strategies for database integration through HARE.

### 5.2.1 Design Intention of Experiment

We construct two trials. Fig. 5.12 is represented trail 1. The results are measured by a number of correspondences compared to direct matching results in order to measure accuracy of pivot ontology model. Fig. 5.13 is represented trail 2. In this experiment, the precision, recall, and f-measure are analyzed. There are three data (i.e. true positive, false positive, and false negative) would be collected in the experiment to calculate the precision, recall, and f-measure.

- True positive is the correct correspondences from the correspondences derived by semi-automatic matching techniques.
- False positive the wrong correspondences from the correspondences derived by semi-automatic matching techniques.
- False negative is the correct correspondences that cannot be derived by semi-automatic matching techniques.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- The precision is the proportion of selected correspondences that are correct.
- The recall is proportion of correct correspondences that are selected.
- The f-measure is a combined measure that assesses the precision and recall trade off.

The range of precision, recall, and f-measure values is between 0-1. The results are measured by f-measure value compared to direct matching results in order to find the efficacy combining matcher for pivot model.

The results from both trails are able to verify the interoperability of HARE ontology.

| Scenario | Matching without HARE (Number of correspondences) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔U | ER1.1 | ER2.1 | ER3.1 | ER4.1 |

| Scenario | Matching S↔U through HARE (Number of correspondences) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔H | TC4.1 | TC5.1 | TC6.1 | TC7.1 |
| U↔H | TC8.1 | TC9.1 | TC10.1 | TC11.1 |
| S↔H↔U | TC12.1 | TC13.1 | TC14.1 | TC15.1 |

Figure 5.12: Measure by the Number of correspondences (Trial 1)

| Scenario | Matching without HARE (F-Measure: Max =1, Min=0) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔U (Medium DB and Small DB) | ER1.2 | ER2.2 | ER3.2 | ER4.2 |

| Scenario | Matching S↔U through HARE (F-Measure: Max =1, Min=0) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔H (Medium DB and Ontology) | TC4.2 | TC5.2 | TC6.2 | TC7.2 |
| U↔H (Small DB and Ontology) | TC8.2 | TC9.2 | TC10.2 | TC11.2 |

Figure 5.13: Measure by f-measure value (Trial 2)

- $\mathcal{S}$ is a database schema from Sahana system.
- $\mathcal{U}$ is a database schema from Ushahidi system.
- $\mathcal{H}$ is the HARE ontology.
- $\mathcal{S} \leftrightarrow \mathcal{U}$ is correspondences between Sahana and Ushahidi.
- $\mathcal{S} \leftrightarrow \mathcal{H}$ is correspondences between Sahana and HARE ontology.
- $\mathcal{U} \leftrightarrow \mathcal{H}$ is correspondences between Ushahidi and HARE ontology.
- $\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$ is correspondences between Sahana and Ushahidi through HARE ontology.

- *TC* stands for test case.

We are using COMA++ [hDR02, ADMR05, DR07] as a matching tool. COMA++ is a schema matching infrastructure to provide matching algorithm. Fig.5.14 shows the graphical user interface of COMA++. It imports source and target data to match processing. By experiments, we use 0.4 as a low-threshold similarity value for finding correspondences.

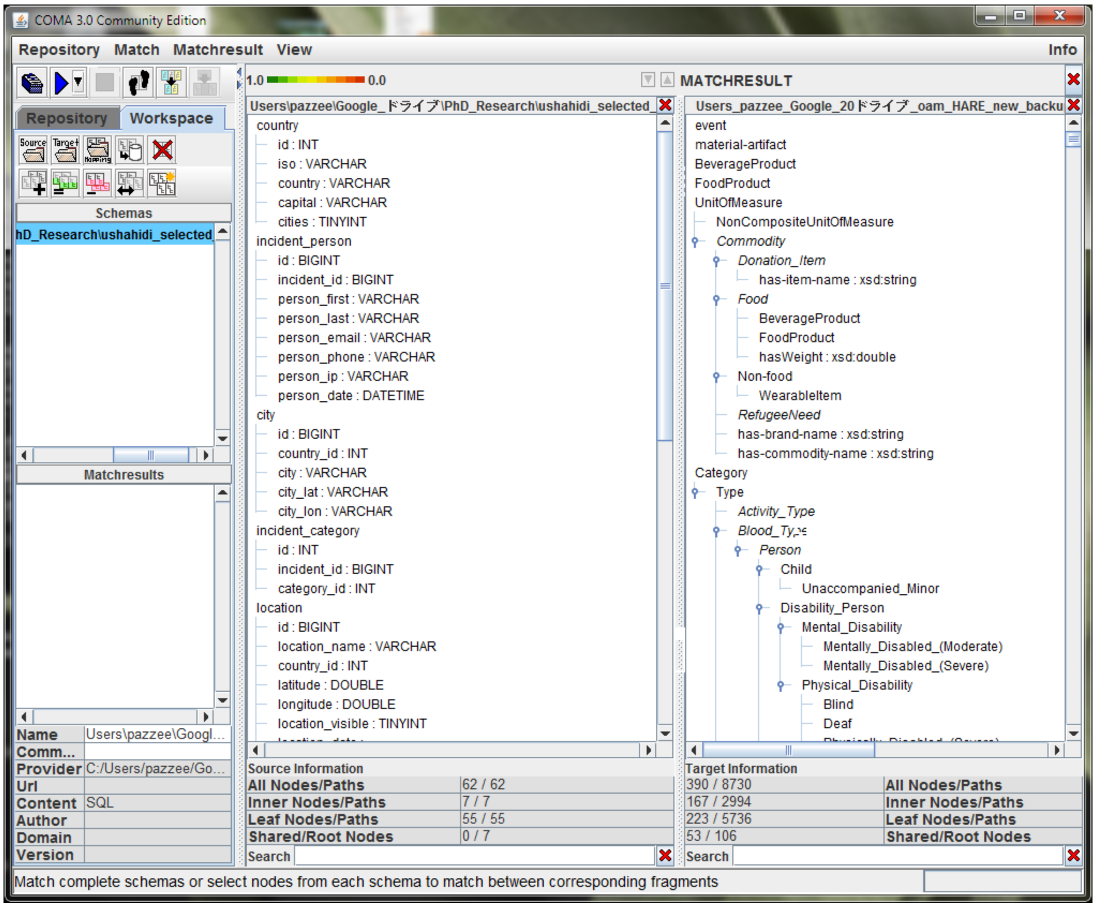



Figure 5.14: User interface of COMA++

### 5.2.2 Trial 1: Measure by the Number of correspondences

We do the evaluation according to following steps;

1. Matching Sahana and Ushahidi without HARE ontology

   (a) Expected Result 1.1, 2.1, 3.1, and 4.1: we try to find the correspondences of $\mathcal{S} \leftrightarrow \mathcal{U}$. Ushahidi and Sahana schema are imported to COMA++ to be a source and target, respectively. The four cases have been the variables to tune up. The correspondence results are collected as final results.

2. Matching Sahana and Ushahidi through HARE ontology

   (a) Test case 4.1, 5.1, 6.1, and 7.1: we try to find the correspondences of $\mathcal{S} \leftrightarrow \mathcal{H}$ first. Sahana schema is imported to COMA++ as a source and HARE schema is imported to COMA++ as a target. The four cases have been the variables to tune up. The correspondence results are collected to be used as the input data in test case 12.1, 13.1, 14.1, and 15.1.

   (b) Test case 8.1, 9.1, 10.1, and 11.1: we try to find the correspondences of $\mathcal{U} \leftrightarrow \mathcal{H}$ first. Ushahidi schema is imported to COMA++ as a source and HARE schema is imported to COMA++ as a target. The four cases have been the variables to tune up. The correspondence results are collected to be used as the input data in test case 12.1, 13.1, 14.1, and 15.1.

   (c) Test case 12.1, 13.1, 14.1, and 15.1: we try to manually combine all correspondence results from test case 4.1-11.1 for finding the correspondence results of $\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$.

### 5.2.3 Trial 2: Measure by F-measure value

We do the evaluation according to following steps;

1. Matching Sahana and Ushahidi without HARE ontology

   (a) Expected Result 1.2, 2.2, 3.2, and 4.2: we try to find the correspondences of $\mathcal{S} \leftrightarrow \mathcal{U}$. Ushahidi and Sahana schema are imported to COMA++ to be a source and a target, respectively. The four cases have been the variables to tune up. Precision, Recall, and f-measure are analyzed.

2. Matching Sahana and Ushahidi through HARE ontology

(a) Test case 4.2, 5.2, 6.2, and 7.2: we try to find the correspondences of $\mathcal{S} \leftrightarrow \mathcal{H}$ first. Sahana schema is imported to COMA++ as a source and HARE schema is imported to COMA++ as a target. The four cases have been the variables to tune up. Precision, Recall, and f-measure are analyzed.

(b) Test case 8.2, 9.2, 10.2, and 11.2: we try to find the correspondences of $\mathcal{U} \leftrightarrow \mathcal{H}$ first. Ushahidi schema is imported to COMA++ as a source and HARE schema is imported to COMA++ as a target. The four cases have been the variables to tune up. Precision, Recall, and f-measure are analyzed.

| Matching approaches | Strategies | | | | | |
|---|---|---|---|---|---|---|
| | Name | Name and Synonym | Path | Leave | Parent | Datatype |
| **Case1: Simple** | Trigram | - | Trigram | Trigram | Trigram | - |
| **Case2: Datatype** | Trigram | - | Trigram | Trigram | Trigram | Datatype similarity |
| **Case3: Synonym** | Trigram | Trigram | Trigram | Trigram | Trigram | - |
| **Case4: Synonym + Datatype** | Trigram | Trigram | Trigram | Trigram | Trigram | Datatype similarity |

Figure 5.15: Semi-automatic matcher composition

The experiments of trail 1 and 2 are arranged in four cases. Fig. 5.15 shows the detail of each case. We describe the idea of each strategy that is used in each case as follows[Do06];

1. Name: this strategy consider only the name of element. It performs some pre-processing steps and the similarity is obtained from string matcher (Trigram).

2. Name and Synonym: this strategy consider as the same as Name strategy, but it performs additional matcher that is Synonym matcher and the result is combined with the similarity result from Trigram.

3. Path: this strategy uses paths of data as inputs. The similarity is computed by combining the similarity values of elements on the paths.

4. Leave: this strategy will combine the leaf-level that performs bottom-up similarity. This strategy aims at more stable similarity in cases of structural conflicts.

5. Parent: this strategy performs top-down similarity to derive the similarity between elements from their parents.

6. Data type: this strategy matches elements based on data type similarity that will be working well in relational schemata.

Above strategies are randomly combined for four cases. Trigram is mostly used in entire strategies. The trigram similarity between two strings is determined by the number of matching letter triples in both strings[Rod]. We describe the idea of each case as follows;

1. Case 1 is a simple case which is a base case for compare other cases. It will combine four strategies for matching, i.e., name, path, leave, parent strategies.

2. Case 2 is data type case. The simple case will be combined with the data type strategy.

3. Case 3 is synonym case. The simple case will be combined with the name and synonym strategies.

4. Case 4 is synonym and data type case. This case will combine all six strategies, i.e., name, synonym, path, leave, parent, and data type strategies.

### 5.2.4 Matching Results of Semi-Automatic Matching

The precision, recall, F-measure, and also numbers of correspondences have been analyzed. Precision is the proportion of selected correspondences that are correct. Recall is the proportion of correct correspondences that are selected. F-measure is a combined measure that assesses the precision and recall trade off. All values of precision, recall, F-measure is between 0-1. Maximum result is 1 and minimum result is 0. The following list is the matching results:

**Trial 1: Measure by the Number of correspondences**

| Scenario | Matching without HARE (Number of correspondences) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔U ($ER_m$=13) | ER1.1 **3** | ER2.1 **3** | ER3.1 **1** | ER4.1 **5** |

| Scenario | Matching S↔U through HARE (Number of correspondences) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔H ($ER_m$=107) | TC:4.1 46 | TC:5.1 29 | TC:6.1 63 | TC:7.1 71 |
| U↔H ($ER_m$=24) | TC:8.1 5 | TC:9.1 4 | TC:10.1 4 | TC:11.1 7 |
| S↔H↔U ($ER_m$=13) | TC:12.1 **2** | TC:13.1 **1** | TC:14.1 **2** | TC:15.1 **5** |

Figure 5.16: Trial 1: Experiment Result of Semi-Automatic Matching (Number of correspondences)

| Scenario | Matching without HARE (% of manual matching) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔U ($ER_m$=13) | ER1.1 **23.08%** | ER2.1 **23.08%** | ER3.1 **7.69%** | ER4.1 **38.46%** |

| Scenario | Matching S↔U through HARE (% of manual matching) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔H ($ER_m$=107) | TC:4.1 43% | TC:5.1 27.1% | TC:6.1 58.88% | TC:7.1 66.36% |
| U↔H ($ER_m$=24) | TC:8.1 20.83% | TC:9.1 16.67% | TC:10.1 16.67% | TC:11.1 29.17% |
| S↔H↔U ($ER_m$=13) | TC:12.1 **15.38%** | TC:13.1 **7.69%** | TC:14.1 **15.38%** | TC:15.1 **38.46%** |

Figure 5.17: Trial 1: Experiment Result of Semi-Automatic Matching (% of manual matching)

Fig. 5.17 is the results that compare the results from Fig. 5.16 to the percentage of manual matching. $ER_m$ means the expected result from manual matching.

1. Matching $\mathcal{S} \leftrightarrow \mathcal{U}$ without HARE ontology: Expected Result 1.1, 2.1, 3.1, and 4.1, which are in Fig.5.16, have results 3, 3, 1, and 5, respectively. As Fig.5.17, they have results 23.08%, 23.08%, 7.69%, and 38.46%, respectively. Case 4 has the highest value.

2. Matching $\mathcal{S} \leftrightarrow \mathcal{U}$ through HARE ontology.

(a) Test case 4.1, 5.1, 6.1, and 7.1, which are in Fig.5.16, have results 46, 29, 63, and 71, respectively. As Fig.5.17, they have results 43%, 27.1%, 58.88%, and 66.36%, respectively. Case 4 has the highest value.

(b) Test case 8.1, 9.1, 10.1, and 11.1, which are in Fig.5.16, have results 5, 4, 4, and 7, respectively. As Fig.5.17, they have results 20.83%, 16.67%, 16.67%, and 29.17%, respectively. Case 4 has the highest value.

(c) Test case 12.1, 13.1, 14.1, and 15.1, which are in Fig.5.16, have results 2, 1, 2, and 5, respectively. As Fig.5.17, they have results 15.38%, 7.69%, 15.38%, and 38.46%, respectively. Case 4 has the highest value.

All results are less than manual matching results. The effective cases have been compared by ranking the different value ($DV$) from Fig.5.17 between expected results 1.1, 2.1, 3.1, and 4.1 ($ER$) and test cases 12.1, 13.1, 14.1, and 15.1 ($TC$).

$$DV = \mid ER - TC \mid$$

The ranking has been ordered as case 4: data type and synonym ($\mid 38.46 - 38.46 \mid = 0$), case 3: synonym ($\mid 7.69 - 15.38 \mid = 7.69$), case 1: simple ($\mid 23.08 - 15.38 \mid = 7.7$), and case 2: data type ($\mid 23.08 - 7.69 \mid = 15.39$). Case 4 shows the highest correspondence value.

**Trial 2: Measure by F-measure**

1. Matching $\mathcal{S} \leftrightarrow \mathcal{U}$ without HARE ontology: the matching results of Sahana and Ushahidi have been shown in Fig.5.18. The bar graph depicts the four cases. Case 1, 2, 3, and 4 are the expected result 1.2, 2.2, 3.2, and 4.2, respectively. All test cases have been presented that high recall and the expected result 4.2 is the highest value of F-measure, that is 0.28. The high recall means the highly correct correspondences that are derived from each case are selected.

2. Matching Sahana and Ushahidi through HARE ontology

(a) Test case 4.2, 5.2, 6.2, and 7.2: the matching results of Sahana and HARE have been shown in Fig.5.19. The bar graph depicts the four cases. Case 1, 2, 3, and 4 are test case 4.2, 5.2, 6.2, and 7.2, respectively. All test cases have been presented that high precision and test case 7.2 is the highest value of F-measure, that is 0.65. The high precision means the selected correspondences that are derived from each case is highly correct.

(b) Test case 8.2, 9.2, 10.2, and 11.2: the matching results of Sahana and HARE have been shown in Fig.5.20. The bar graph depicts the four cases. Case 1, 2, 3, and 4 are Test case 8.2, 9.2, 10.2, and 11.2, respectively. All test cases have been presented that high precision and test case 11.2 is the highest value of F-measure, that is 0.45. The high precision means the selected correspondences that are derived from each case is highly correct.

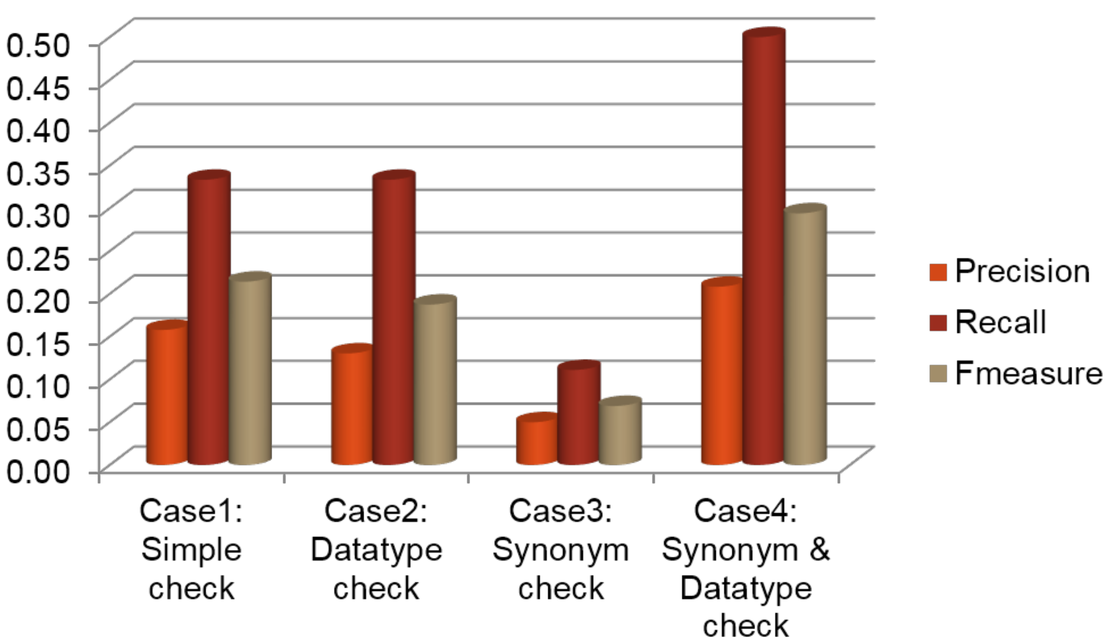



Figure 5.18: Expected Result 1.2, 2.2, 3.2, and 4.2

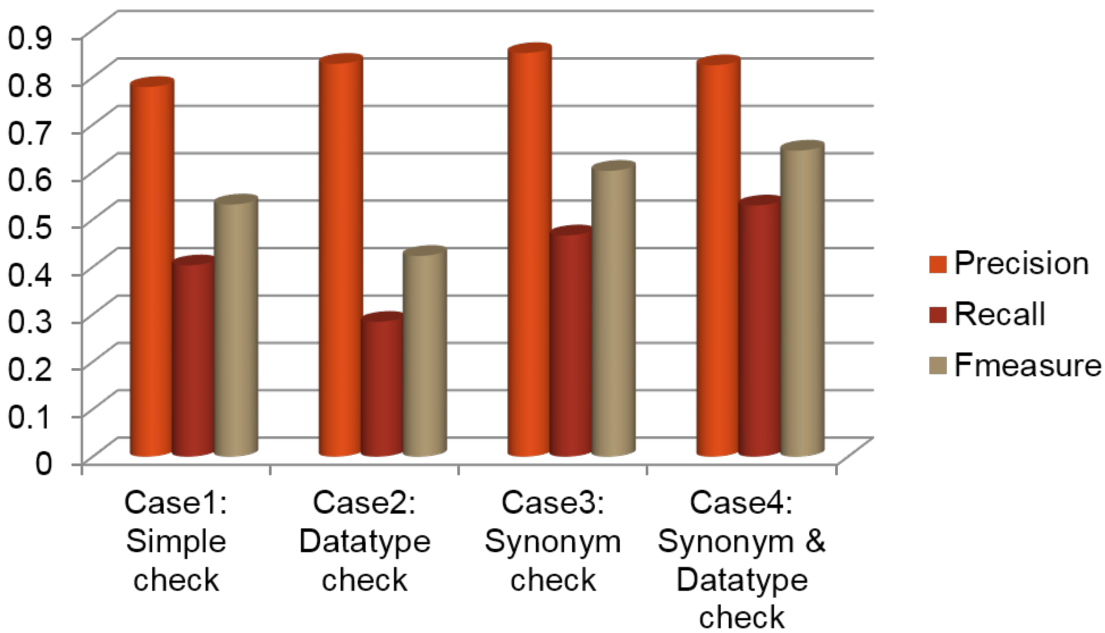



Figure 5.19: test case 4.2, 5.2, 6.2, and 7.2

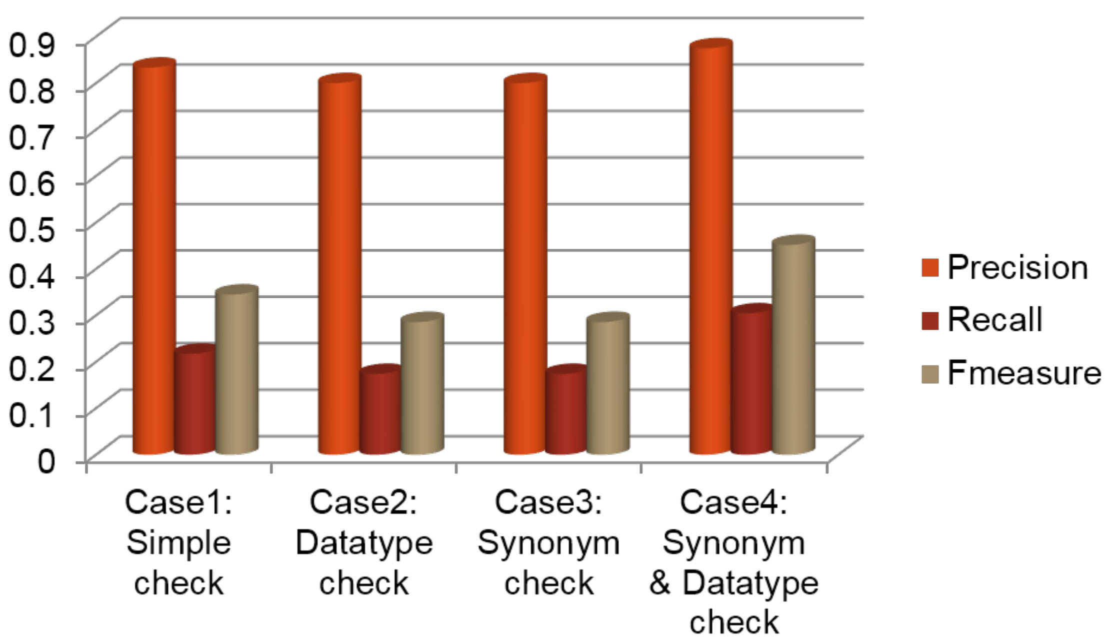



Figure 5.20: test case 8.2, 9.2, 10.2, and 11.2

| Scenario | Matching without HARE (F-Measure: Max =1, Min=0) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔U (Medium DB and Small DB) | ER1.2 0.2 | ER2.2 0.18 | ER3.2 0.06 | ER4.2 **0.28** |

| Scenario | Matching S↔U through HARE (F-Measure: Max =1, Min=0) | | | |
|---|---|---|---|---|
| | case1 Simple | case2 DataT | case3 Syn | case4 D+S |
| S↔H (Medium DB and Ontology) | TC:4.2 0.53 | TC:5.2 0.42 | TC:6.2 0.60 | TC:7.2 **0.65** |
| U↔H (Small DB and Ontology) | TC:8.2 0.34 | TC:9.2 0.29 | TC:10.2 0.29 | TC:11.2 **0.45** |

Figure 5.21: Semi-automatic matching result (F-measure)

The effective matching strategy composition is shown in Fig. 5.21 that is case 4: the synonym and data type strategy combining. The aim of this study is to learn appropriated strategies. The improved performance is out of scope for the research, the results are derived from the basic schema matching strategies. The result of $S1$ represents the schema

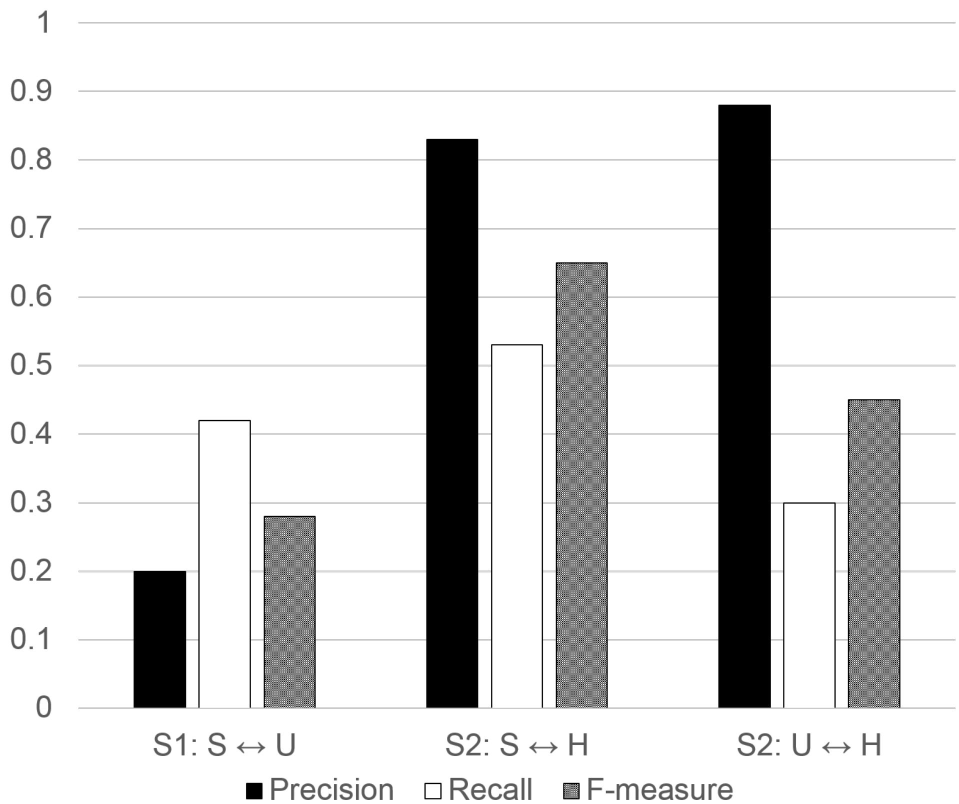



Figure 5.22: Matching comparison between with/without HARE ontology by semi-automatic matching

overlap between Sahana and Ushahidi. The last two results ($S2$) represent the schema overlap with the HARE ontology. The best semi-automatic matching is in $S2$: $\mathcal{S} \leftrightarrow \mathcal{H}$

has high recall, high precision. The precision is high for all $S2$ scenarios. The F-measure of $S2$ is higher than $S1$. The results are lower than manual matching. However, both results from the manual and semi-automatic matching have the same direction.

However, both results from the manual and semi-automatic matching are expected to be close or equal. The PivotOntology-to-Database matching is computed in both element and structure levels, e.g., lexical matching and data type matching. Thus, an assumption of evaluation is that Case 4, i.e., combination of synonym and data type strategies, can be an appropriate matching strategy for both groups of experiments.

Table 5.6: Semi-automatic matching results

| Group | Scenario | % of manual matching | | | |
|---|---|---|---|---|---|
| | | Case 1 | Case 2 | Case 3 | Case 4 |
| 1 | $\mathcal{S} \leftrightarrow \mathcal{U}$ ($ER_m$=13) | 23.08% | 23.08% | 7.69% | 38.46% |
| 2 | Test Case 1: $\mathcal{S} \leftrightarrow \mathcal{H}$ ($ER_m$=107) | 43% | 27.1% | 58.88% | 66.36% |
| | Test Case 2: $\mathcal{U} \leftrightarrow \mathcal{H}$ ($ER_m$=24) | 20.83% | 16.67% | 16.67% | 29.17% |
| | Test Case 3: $\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$ ($ER_m$=13) | 15.38% | 7.69% | 15.38% | 38.46% |

Table 5.6 shows the results of semi-automatic matching compared with the results of manual matching (Fig 5.8), where $ER_m$ is the number of correspondences obtained from manual matching.

The most effective matching strategy composition is Case 4 (Combination of synonym and data type strategies). The aim of this empirical study is to investigate appropriate semi-automatic matching strategies. Improvement of the performance of semi-automatic matching techniques is beyond the scope of this research. However, according to Table 5.6, the results of semi-automatic matching are in line with those of manual matching. In particular, using the strategy combination of Case 4, the result of Test Case 3 ($\mathcal{S} \leftrightarrow \mathcal{H} \leftrightarrow \mathcal{U}$), i.e., 38.46%, is the same as that of manual matching without HARE ontology ($\mathcal{S} \leftrightarrow \mathcal{U}$).

## 5.3 Conclusion

Our experiments show that two existing humanitarian aid information systems, i.e., Sahana and Ushahidi, can be extensively integrated by using the HARE ontology, with the mainstream of their schemata being well covered. Our case study not only demonstrates PivotOntology-to-Database schema matching, but also typically presents the integrated information extended by using the HARE ontology. In addition, several semi-automatic matching strategies are investigated.

# Chapter 6

# Situation awareness in humanitarian aid information systems

This chapter discusses the interoperability in the future humanitarian aid information systems. What is the challenging, and how the Pivot ontology approach makes interoperability more explicit.

## 6.1 Disaster Management

Disaster management or emergency management is the managerial function charged with creating the framework within which communities reduce vulnerability to hazard and cope with disaster. Simple definition of emergency management is a discipline that deals with risk and risk avoidance. It is an essential role of government [HBC13]. The disaster cycle consists of the steps that emergency managers take in planning for and responding to the disaster. Each step in the disaster cycle correlates to the part of the ongoing cycle that is the emergency management (Fig. 6.1). The emergency management can be defined as the process as disaster management phases [Pet85]:

- Preparedness: Developing a response plan and training first responders to save lives and reduce disaster damage. This includes the identification of critical resources and the development of necessary agreements among responding agencies.

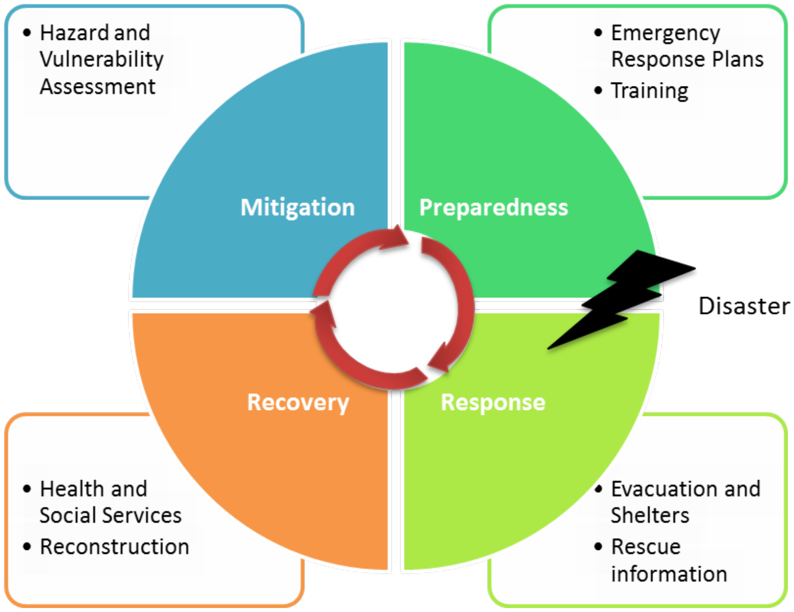



Figure 6.1: Emergency Management Phases

- Response: Providing the emergency aid and assistance, reducing the probability of secondary damage, and minimizing the problems for recovery operations. The focus will quickly turn to fulfilling the basic humanitarian needs of the affected population. Donations are often sought this period, especially for large disasters that overwhelm local capacity.

- Recovery: Providing the immediate support during the early recovery necessary to return vital life support systems to minimum operation levels, and continuing to provide the support until the community returns to normal. During the recovery phase, lessons learned are collected and shared within the emergency response community.

- Mitigation: It is almost concurrent with the recovery phase. The activities in this phase, e.g., health care, safety, and welfare of society management, have been determined to exist and implemented by a risk reduction program.

Finally, the lessons learned from the response, recovery, and mitigation phases have been returned for revising the plans, their understanding of the material and human resources' needs in the preparedness phase.

## 6.2 Emergency Response and Recovery

During and in the aftermath of disasters, the emergency response and recovery is the organizing, coordinating, and directing of available resources in order to respond to the emergency event and bring the affected area back to normal situation as quickly as possible. The humanitarian aid is the action designed to save lives, alleviate suffering and maintain and protect human dignity during and in the aftermath of man-made crises and natural disasters [Dev]. The humanitarian organizations or individuals require for data gathering on the disaster situations, such as population, housing, transportation, facility system. These data are useless if it is not organized, synthesized and prepared for valuable decision making. Data have been collected and analyzed by several systems, such as computer-aided dispatch, situational reporting, environmental monitoring stations, advanced weather prediction systems. To make the data useful, the information from the various fields of humanitarian aid needs to be tied together in a way that makes it semantically possible to plan for different scenarios.

### 6.2.1 Interoperability of Humanitarian Aid Systems

Section 2 discusses the need for the interoperability in the humanitarian aid domain. To being interoperable in the humanitarian aid, the interoperability needs to be achieved in semantic understanding to ensure that the integrated information is semantically merged from the source of information and the destination of information. The interoperability also needs to be achieved in the domain knowledge to ensure that the providers, recipients, and implementers can be understand the situation. They are able to project the future by analyzing the integrated information in making a decision. It directly leads to the situation awareness and the situation assessment for information sharing and collaboration among heterogeneous systems.

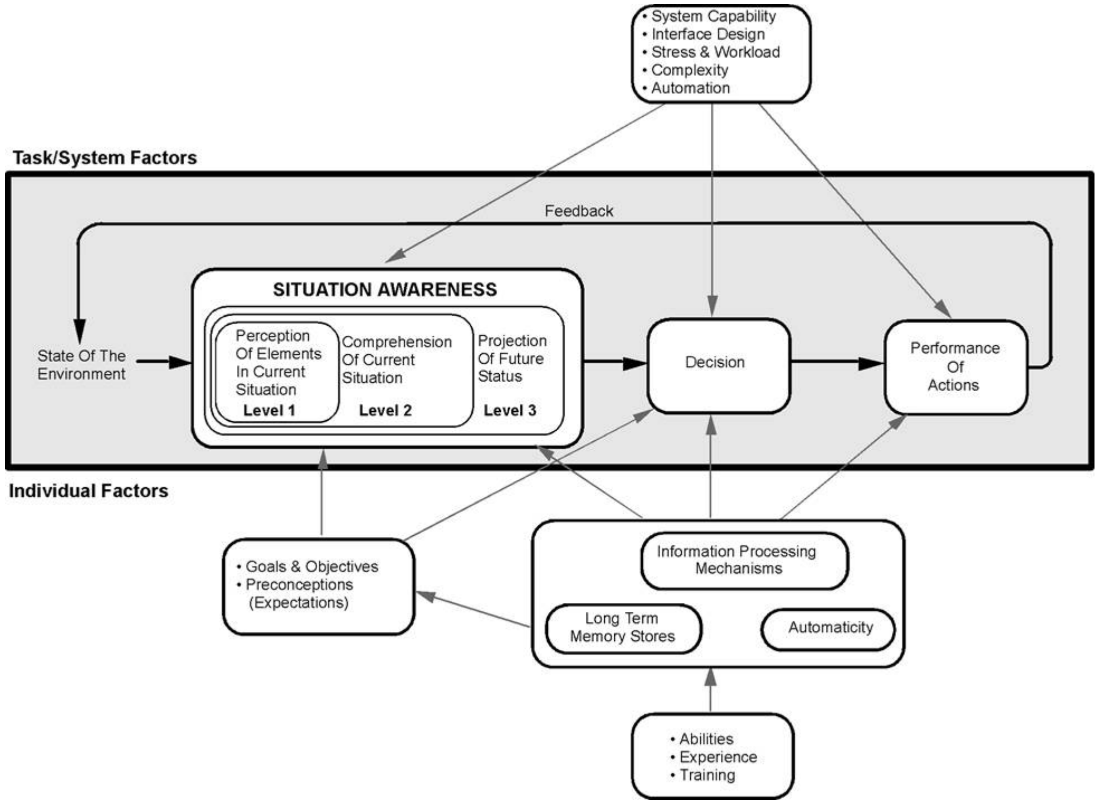



Figure 6.2: Situation Awareness in dynamic decision making (adapted from [End95])

## 6.3 Situation Awareness and Assessment

Situation awareness is the real-world changing knowledge that is critical for effective decision making and action. The situation awareness breaks down into three separated levels [EG00]:

1. Level 1: Perception of the elements in the environment
2. Level 2: Comprehension of the current situation
3. Level 3: Projection of future status

The relationships between these levels and the internal and external factors affect the development of a complete understanding and lead to be an effective decision making as illustrated in Fig. 6.2.

The situation awareness is conceptualized as state of knowledge that can help to explain the ability of experts to handle and overtake the complex situations. The situation assessment is an active process of seeking information from the environment [End95, EG00]. To achieve the state of knowledge, the situation assessment for humanitarian aid domain should be designed.

### 6.3.1 Situation Assessment for Humanitarian Aid

The intermediate processes for the humanitarian aid decision making are identified. The process model is shown in Fig. 6.3. It is briefly described details as follows:

1. Level 1: Information Gathering- it begins with an information gathering that explores and collects data from the heterogeneous sources as in the form of RDBs. The relevant data will be stored in internal database management system (DMS).

2. Level 2: Situation Refinement- relationships are established between elements from DMS and the HARE ontology that is a common ontology for the humanitarian aid domain.

3. Level 3: Information Identification- information is integrated by matching schema through the HARE ontology. The information needs can be estimated and predicted, but it depends on goals of the decision making system.

A certain degree of relationship can be viewed between aspects of the HAIP model (Fig. 6.3) and Situation Awareness model (Fig. 6.2). The Information Gathering in HAIP model is related to domain information directly that helps to gain perception (Level 1 of the situation awareness model) of the information related in the situation. The Situation Refinement is a synthesis process of domain information from Information Gathering and includes comprehension (Level 2 of the situation awareness model) of the current situation through HARE ontology. In order to forecast the future situation (Level 3 of the situation awareness model), Information Identification helps to get the information needs for the decision making.

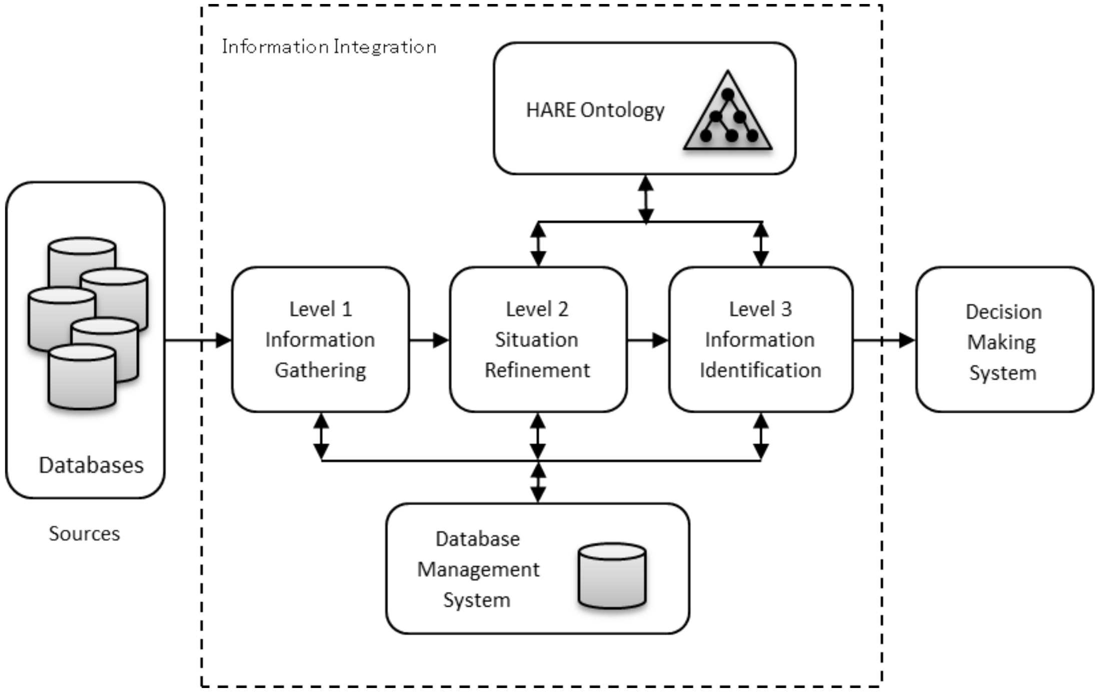



Figure 6.3: Humanitarian Aid Information Processing (HAIP) Model

## 6.4 Conclusion

The purpose of this chapter deals with how to achieve the domain knowledge understanding. We decide the HAIP model with respect to the situation awareness model. The HAIP model is combined the humanitarian aid information integration processes to achieve their knowledge. The HARE ontology construction model and methodology are explained in section 4 and level 1, 2, and 3 are described in detail in section 5 and section 6. This model can be applied for the system that uses the information integration for the decision making.

# Chapter 7

# Conclusion and Future works

## 7.1 Conclusion

The problem of semantic interoperability between three groups of actors including providers (donors), recipients (affected populations), and implementers (e.g., government, foundation, Red Cross, NGOs, UN agencies) is a critical issue in the humanitarian aid domain. Humanitarian aid information, including information on the occurrences of disaster situations, victims, shelters, resources, facilities, etc., is usually heterogeneous, rapidly changeable, ambiguous, and large. It is widely distributed and owned by different organizations, and as such, it is stored diversely in distinct heterogeneous data sources in different locations. Because of the increasing of humanitarian aid information, various systems are individually developed by organizations. The collaboration among individual systems leads to be the ambiguous collaboration. Successful and innovative collaboration solutions are limited by a large number of humanitarian aid actors and incompatible information. An important challenge of information integration in the humanitarian aid domain is to identify the correlation of data from multiple sources.

An ontology enables one to reuse and share application domain knowledge using a common vocabulary across heterogeneous application. An ontology provides a promising approach to deal with semantic heterogeneity problems. Although there are existing ontologies for disaster management, these ontologies are application-dependent, i.e., they are developed for specific applications.

Our objective is to provide a basis for common understanding of terms related to humanitarian aid in the disaster management domain in order to facilitate information interoperability. In order to avoid terminology alignment between every pair of different systems, we design a pivot ontology framework, and present a pivot construction methodology and a PivotOntology-to-Database schema matching methodology. The pivot construction methodology is adopted from an ontology engineering technique, and the PivotOntology-to-Database schema matching methodology is based on a linguistic relation approach. We focus on the techniques for reusing existing ontologies and semantic hierarchical conceptual model to design a common ontology for all humanitarian aid information systems. To integrate humanitarian aid in emergency information from several databases, the Humanitarian Aid for Refugee in Emergencies (HARE) ontology has been proposed. On the content level, the HARE ontology is able be a common understanding of humanitarian aid. The concepts in HARE ontology are possible to communicate between humanitarian aid systems. The HARE ontology is proposed based on pivot ontology construction methodology. The sources of knowledge for HARE ontology building is derived from the standard handbooks for the emergency rescue.

Coverage of the HARE ontology is evaluated with respect to comparison against knowledge source, and matching with existing systems. The evaluations demonstrate that the HARE ontology is broadly compatible with existing database schemata. Two existing humanitarian aid information systems, i.e., Sahana, and Ushahidi, are extensively integrated covering the mainstream of their schema by HARE ontology. A case study is not only demonstrated the PivotOntology-to-Database schema matching, but also typically presented the information after sharing as the direct integrated information. In addition, the semi-automatic matching strategies are exploited in this research to be analyzed the appropriated combining Matcher for schema matching between HARE ontology and databases.

## 7.2 Open Problems

We provide some directions in which, in our opinion, research on humanitarian aid management is necessary to make existing databases available for humanitarian aid applications. In this section, we point out current needs that help the collaboration in disaster management system as follows;

1. Information gathering - In real world, humanitarian aid information has not usually been collected for humanitarian aid systems. Information gathering on humanitarian aid is a disposable information. The officers do not store dynamic information in systems, such as donation history and assistance tracking. They keep only static information, such as disaster information. We can expect that the more data collection in this field, the more information for integration.

2. Information integration application - we present PivotOntology-to-Database matching techniques in order to integrate humanitarian aid information. In real world, information is not only integrated. It has to be validated the duplicated information before it is disseminated to users. On the other hand, new types of input are expected. The pivot ontology needs to be matched with several types of input.

## 7.3 Future works

In this research, we designed a situation assessment model called the Humanitarian Aid Information Processing (HAIP) model for the future humanitarian aid decision making systems. The HAIP model consists of Information Gathering Process, Situation Refinement Process, and Information Identification Process. This model relies on the HARE ontology as a common conceptualization of humanitarian aid domain. Future work involves improving the extensible HARE ontology in the broader scope to handle the broader humanitarian aid knowledge. We aim to offer the HARE ontology to others and to implement the emergency response service of ontology-based decisions support system. The service will address issues of expert knowledge interoperability. The provider or implementer needs explicit information about recipient for making decision to rescue people. An ontological model is considered as a pivot for interoperating the emergency response applications in order to share information during all phases of the disaster management.

# Chapter 8

# Appendix

## 8.1 Appendix A: Lower Level of HARE Ontology

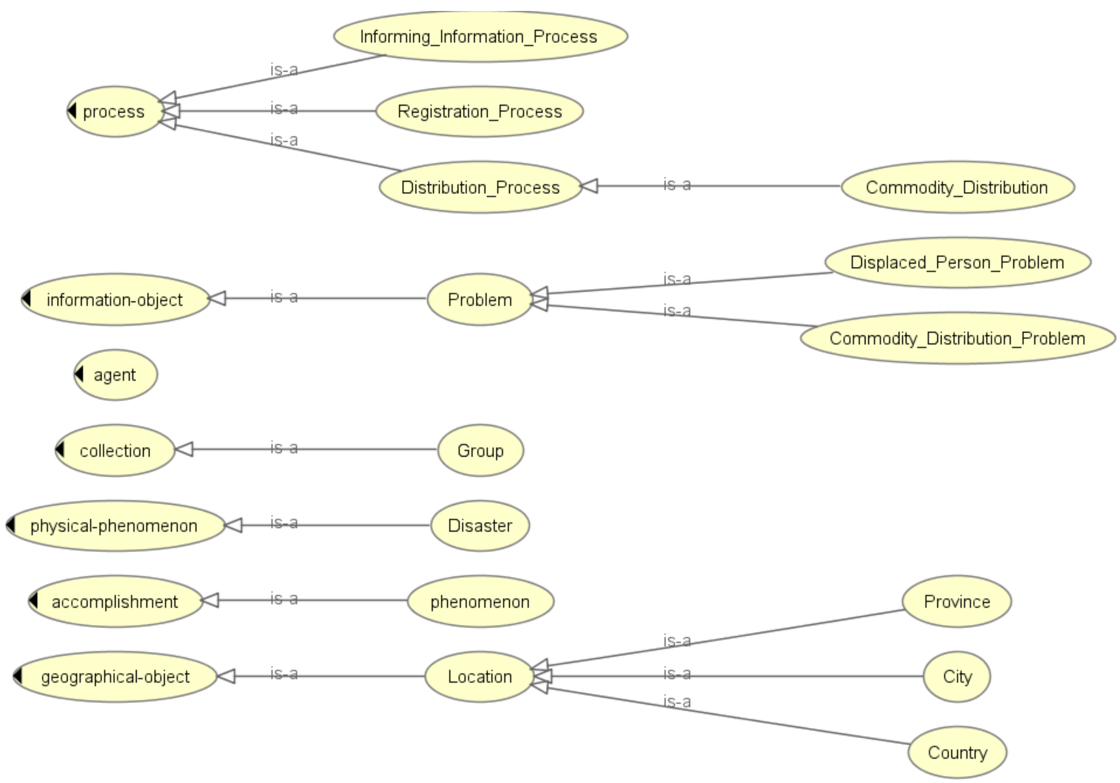



Figure 8.1: HARE Ontology (part 1)

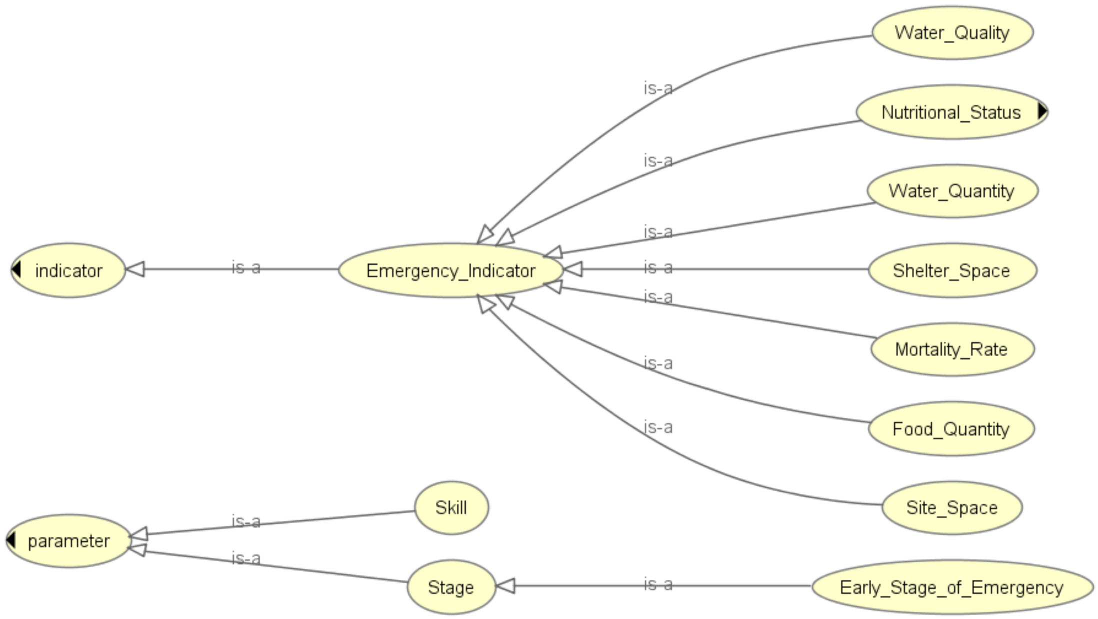



Figure 8.2: HARE Ontology (part 2)


Figure 8.3: HARE Ontology (part 3)

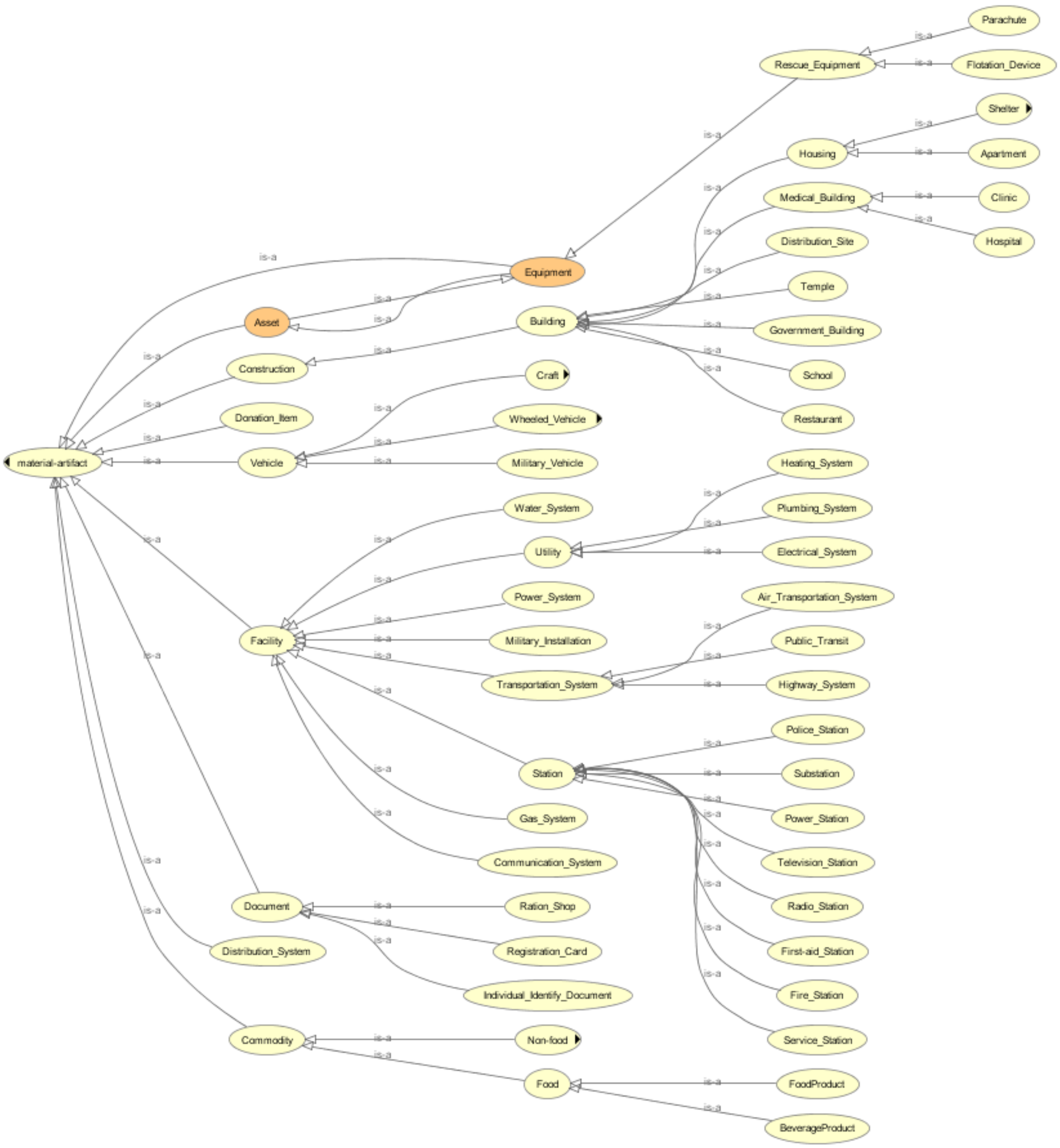



Figure 8.4: HARE Ontology (part 4)

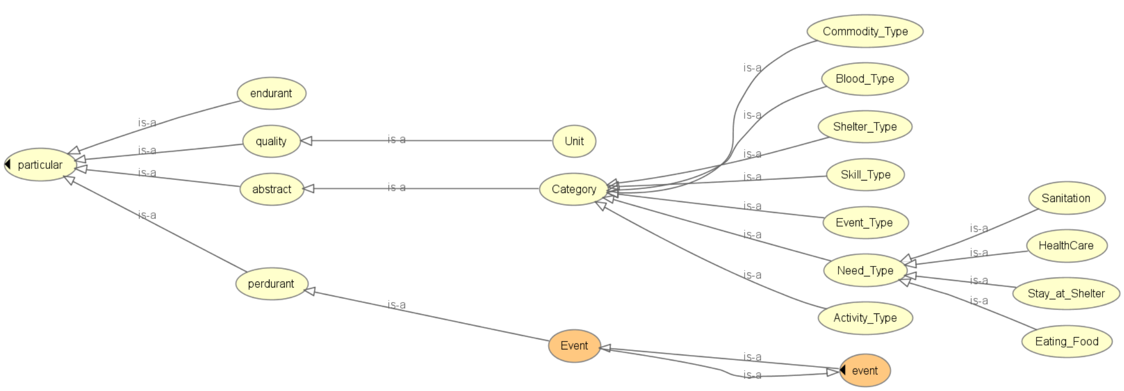



Figure 8.5: HARE Ontology (part 5)


Figure 8.6: HARE Ontology (part 6)

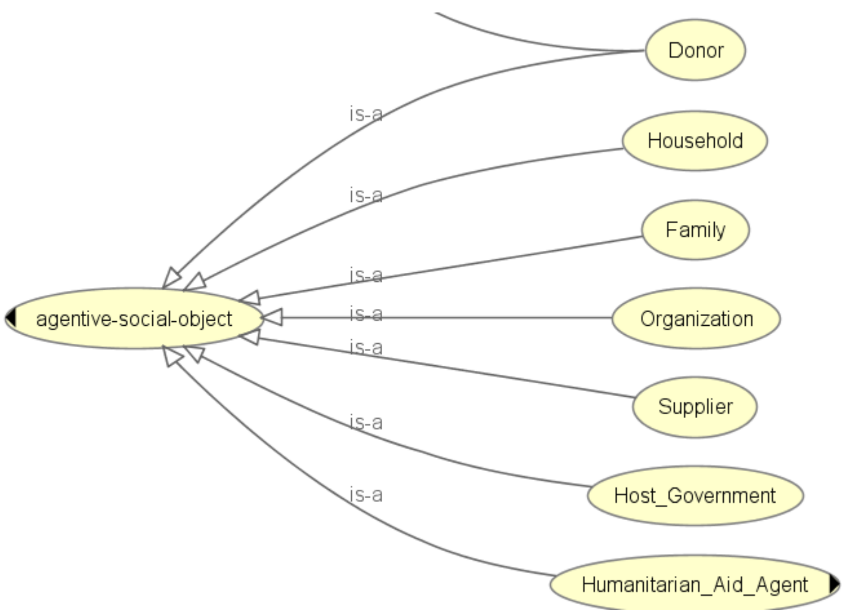



Figure 8.7: HARE Ontology (part 7)

## 8.2 Appendix B: Database schemas of existing systems

### 8.2.1 B.1 Sahana Eden Database Schema

Table 8.1: Table: asset_asset

| Column name | Type |
|---|---|
| id | int(11) |
| track_id | int(11) |
| doc_id | int(11) |
| number | varchar(512) |
| item_entity_id | int(11) |
| item_id | int(11) |
| type | int(11) |
| sn | varchar(512) |
| supplier | varchar(512) |
| purchase_date | date |
| purchase_price | double |
| purchase_currency | varchar(3) |
| location_id | int(11) |
| assigned_to_id | int(11) |
| comments | text |
| building_name | varchar(512) |
| address | varchar(512) |
| postcode | varchar(512) |

Table 8.2: Table: cr_shelter

| Column name | Type |
|---|---|
| id | int(11) |
| site_id | int(11) |
| name | varchar(64) |
| organisation_id | int(11) |
| shelter_type_id | int(11) |
| shelter_service_id | int(11) |
| location_id | int(11) |
| phone | varchar(512) |
| person_id | int(11) |
| capacity | int(11) |
| population | int(11) |
| source | varchar(512) |
| comments | text |
| building_name | varchar(512) |
| address | varchar(512) |
| postcode | varchar(512) |

Table 8.3: Table: cr_shelter_service

| Column name | Type |
|---|---|
| id | int(11) |
| name | varchar(512) |

Table 8.4: Table: cr_shelter_type

| Column name | Type |
|---|---|
| id | int(11) |
| name | varchar(512) |

Table 8.5: Table: doc_document

| Column name | Type |
|---|---|
| id | int(11) |
| site_id | int(11) |
| doc_id | int(11) |
| name | varchar(128) |
| file | varchar(512) |
| url | varchar(512) |
| person_id | int(11) |
| organisation_id | int(11) |
| date | date |
| location_id | int(11) |

Table 8.6: Table: event_activity

| Column name | Type |
|---|---|
| id | int(11) |
| event_id | int(11) |
| activity_id | int(11) |

Table 8.7: Table: event_asset

| Column name | Type |
|---|---|
| id | int(11) |
| event_id | int(11) |
| asset_id | int(11) |

Table 8.8: Table: event_event

| Column name | Type |
|---|---|
| id | int(11) |
| scenario_id | int(11) |
| name | varchar(64) |
| exercise | varchar(1) |
| zero_hour | timestamp |
| closed | varchar(1) |

Table 8.9: Table: event_human_resource

| Column name | Type |
|---|---|
| id | int(11) |
| event_id | int(11) |
| human_resource_id | int(11) |

Table 8.10: Table: event_incident

| Column name | Type |
|---|---|
| id | int(11) |
| event_id | int(11) |
| name | varchar(64) |

Table 8.11: Table: gis_location

| Column name | Type |
|---|---|
| id | int(11) |
| name | varchar(128) |
| code | varchar(512) |
| code2 | varchar(512) |
| level | varchar(2) |
| parent | int(11) |
| path | varchar(256) |
| members | text |
| addr_street | text |
| addr_postcode | varchar(128) |
| gis_feature_type | int(11) |
| lat | double |
| lon | double |
| wkt | text |
| url | varchar(512) |
| geonames_id | int(11) |
| osm_id | int(11) |
| lat_min | double |
| lat_max | double |
| lon_min | double |
| lon_max | double |
| elevation | double |
| area | double |
| source | varchar(32) |

Table 8.12: Table: hms_hospital

| Column name | Type |
|---|---|
| id | int(11) |
| site_id | int(11) |
| paho_uuid | varchar(128) |
| gov_uuid | varchar(128) |
| other_ids | varchar(128) |
| name | varchar(64) |
| aka1 | varchar(512) |
| aka2 | varchar(512) |
| facility_type | int(11) |
| organisation_id | int(11) |
| location_id | int(11) |
| address | varchar(512) |
| postcode | varchar(512) |
| city | varchar(512) |
| phone_exchange | varchar(512) |
| phone_business | varchar(512) |
| phone_emergency | varchar(512) |
| website | varchar(512) |
| email | varchar(512) |
| fax | varchar(512) |
| total_beds | int(11) |
| available_beds | int(11) |
| ems_status | int(11) |
| ems_reason | varchar(128) |
| or_status | int(11) |
| or_reason | varchar(128) |

Table 8.13: Table: hms_hospital (cont.)

| Column name | Type |
|---|---|
| facility_status | int(11) |
| clinical_status | int(11) |
| morgue_status | int(11) |
| morgue_units | int(11) |
| security_status | int(11) |
| doctors | int(11) |
| nurses | int(11) |
| non_medical_staff | int(11) |
| staffing | int(11) |
| facility_operations | int(11) |
| clinical_operations | int(11) |
| access_status | varchar(512) |

Table 8.14: Table: hrm_human_resource

| Column name | Type |
|---|---|
| id | int(11) |
| track_id | int(11) |
| organisation_id | int(11) |
| person_id | int(11) |
| type | int(11) |
| job_title | varchar(512) |
| status | int(11) |
| start_date | date |
| end_date | date |
| location_id | int(11) |
| site_id | int(11) |

Table 8.15: Table: hrm_skill

| Column name | Type |
|---|---|
| id | int(11) |
| skill_type_id | int(11) |
| name | varchar(64) |

Table 8.16: Table: hrm_skill_type

| Column name | Type |
|---|---|
| id | int(11) |
| name | varchar(64) |

Table 8.17: Table: org_organisation

| Column name | Type |
|---|---|
| id | int(11) |
| pe_id | int(11) |
| name | varchar(128) |
| acronym | varchar(8) |
| type | int(11) |
| sector_id | text |
| region | varchar(512) |
| country | varchar(2) |
| website | varchar(512) |
| twitter | varchar(512) |
| donation_phone | varchar(512) |

Table 8.18: Table: org_site

| Column name | Type |
|---|---|
| site_id | int(11) |
| deleted | varchar(1) |
| instance_type | varchar(512) |
| uuid | varchar(128) |
| name | varchar(64) |
| location_id | int(11) |
| organisation_id | int(11) |

Table 8.19: Table: project_activity

| Column name | Type |
|---|---|
| id | int(11) |
| doc_id | int(11) |
| project_id | int(11) |
| name | varchar(512) |
| location_id | int(11) |
| multi_activity_type_id | text |

Table 8.20: Table: project_organisation

| Column name | Type |
|---|---|
| id | int(11) |
| project_id | int(11) |
| organisation_id | int(11) |
| role | int(11) |
| amount | double |
| currency_type | varchar(3) |

Table 8.21: Table: project_project

| Column name | Type |
|---|---|
| id | int(11) |
| doc_id | int(11) |
| name | varchar(128) |
| code | varchar(512) |
| description | text |
| start_date | date |
| end_date | date |
| duration | varchar(512) |
| sector_id | text |
| countries_id | text |
| multi_hazard_id | text |
| multi_theme_id | text |
| hfa | text |
| objectives | text |

Table 8.22: Table: pr_address

| Column name | Type |
|---|---|
| id | int(11) |
| pe_id | int(11) |
| type | int(11) |
| location_id | int(11) |
| comments | text |
| building_name | varchar(512) |
| address | varchar(512) |
| postcode | varchar(512) |

Table 8.23: Table: pr_group

| Column name | Type |
| --- | --- |
| id | int(11) |
| pe_id | int(11) |
| group_type | int(11) |
| system | varchar(1) |
| name | varchar(512) |
| description | varchar(512) |

Table 8.24: Table: pr_group_membership

| Column name | Type |
| --- | --- |
| id | int(11) |
| group_id | int(11) |
| person_id | int(11) |
| group_head | varchar(1) |
| description | varchar(512) |

Table 8.25: Table: pr_pentity

| Column name | Type |
| --- | --- |
| pe_id | int(11) |
| pe_label | varchar(128) |

Table 8.26: Table: pr_person

| **Column name** | **Type** |
|---|---|
| id | int(11) |
| pe_id | int(11) |
| track_id | int(11) |
| location_id | int(11) |
| pe_label | varchar(128) |
| missing | varchar(1) |
| volunteer | varchar(1) |
| first_name | varchar(64) |
| middle_name | varchar(64) |
| last_name | varchar(64) |
| initials | varchar(8) |
| preferred_name | varchar(64) |
| local_name | varchar(512) |
| gender | int(11) |
| date_of_birth | date |
| age_group | int(11) |
| nationality | varchar(512) |
| occupation | varchar(128) |
| picture | varchar(512) |

Table 8.27: Table: pr_physical_description

| Column name | Type |
|---|---|
| id | int(11) |
| pe_id | int(11) |
| race | int(11) |
| complexion | int(11) |
| ethnicity | varchar(64) |
| height | int(11) |
| height_cm | int(11) |
| weight | int(11) |
| weight_kg | int(11) |
| blood_type | varchar(512) |
| eye_color | int(11) |
| hair_color | int(11) |
| hair_style | int(11) |
| hair_length | int(11) |
| hair_baldness | int(11) |
| hair_comment | varchar(512) |
| facial_hair_type | int(11) |
| facial_hair_color | int(11) |
| facial_hair_length | int(11) |
| facial_hair_comment | varchar(512) |
| body_hair | varchar(512) |
| skin_marks | text |
| medical_conditions | text |
| other_details | text |

Table 8.28: Table: req_commit

| Column name | Type |
|---|---|
| id | int(11) |
| site_id | int(11) |
| organisation_id | int(11) |
| req_id | int(11) |
| type | int(11) |
| date | date |
| date_available | date |
| committer_id | int(11) |

Table 8.29: Table: req_commit_item

| Column name | Type |
|---|---|
| id | int(11) |
| commit_id | int(11) |
| req_item_id | int(11) |
| item_pack_id | int(11) |
| quantity | double |

Table 8.30: Table: req_commit_person

| Column name | Type |
|---|---|
| id | int(11) |
| commit_id | int(11) |
| skill_id | text |
| person_id | int(11) |

Table 8.31: Table: req_document

| **Column name** | **Type** |
|---|---|
| id | int(11) |
| req_id | int(11) |
| document_id | int(11) |

Table 8.32: Table: req_req

| Column name | Type |
|---|---|
| id | int(11) |
| event_id | int(11) |
| type | int(11) |
| request_number | varchar(128) |
| date | timestamp |
| priority | int(11) |
| purpose | text |
| date_required | timestamp |
| date_required_until | timestamp |
| requester_id | int(11) |
| assigned_to_id | int(11) |
| approved_by_id | int(11) |
| request_for_id | int(11) |
| site_id | int(11) |
| transport_req | varchar(1) |
| security_req | varchar(1) |
| date_recv | timestamp |
| recv_by_id | int(11) |
| commit_status | int(11) |
| transit_status | int(11) |
| fulfil_status | int(11) |
| cancel | varchar(1) |

Table 8.33: Table: req_req_item

| Column name | Type |
|---|---|
| id | int(11) |
| req_id | int(11) |
| item_entity_id | int(11) |
| item_id | int(11) |
| item_pack_id | int(11) |
| quantity | double |
| pack_value | double |
| currency | varchar(3) |
| site_id | int(11) |
| quantity_commit | double |
| quantity_transit | double |
| quantity_fulfil | double |

Table 8.34: Table: req_req_skill

| Column name | Type |
|---|---|
| id | int(11) |
| req_id | int(11) |
| task | varchar(512) |
| skill_id | text |
| quantity | int(11) |
| site_id | int(11) |
| quantity_commit | int(11) |
| quantity_transit | int(11) |
| quantity_fulfil | int(11) |

Table 8.35: Table: supply_catalog

| **Column name** | **Type** |
|---|---|
| id | int(11) |
| name | varchar(128) |
| organisation_id | int(11) |

Table 8.36: Table: supply_catalog_item

| **Column name** | **Type** |
|---|---|
| id | int(11) |
| catalog_id | int(11) |
| item_category_id | int(11) |
| item_id | int(11) |

Table 8.37: Table: supply_item

| **Column name** | **Type** |
|---|---|
| id | int(11) |
| name | varchar(128) |
| code | varchar(16) |
| um | varchar(128) |
| item_category_id | int(11) |
| model | varchar(128) |
| year | int(11) |
| weight | double |
| length | double |
| width | double |
| height | double |
| volume | double |

Table 8.38: Table: supply_item_category

| Column name | Type |
| --- | --- |
| id | int(11) |
| catalog_id | int(11) |
| parent_item_category_id | int(11) |
| code | varchar(16) |
| name | varchar(128) |
| can_be_asset | varchar(1) |
| is_vehicle | varchar(1) |

### 8.2.2 B.2 Ushahidi Database Schema

Table 8.39: Table: category

| Column name | Type |
| --- | --- |
| id | int(11) |
| parent_id | int(11) |
| locale | varchar(10) |
| category_position | tinyint(4) |
| category_title | varchar(255) |
| category_description | text |
| category_color | varchar(20) |
| category_image | varchar(255) |
| category_image_thumb | varchar(255) |
| category_visible | tinyint(4) |
| category_trusted | tinyint(4) |

Table 8.40: Table: city

| **Column name** | **Type** |
|---|---|
| id | bigint(20) |
| country_id | int(11) |
| city | varchar(200) |
| city_lat | varchar(150) |
| city_lon | varchar(200) |

Table 8.41: Table: country

| **Column name** | **Type** |
|---|---|
| id | int(11) |
| iso | varchar(10) |
| country | varchar(100) |
| capital | varchar(100) |
| cities | tinyint(4) |

Table 8.42: Table: incident

| Column name | Type |
| --- | --- |
| id | bigint(20) |
| location_id | bigint(20) |
| form_id | int(11) |
| locale | varchar(10) |
| user_id | int(11) |
| incident_title | varchar(255) |
| incident_description | longtext |
| incident_date | datetime |
| incident_mode | tinyint(4) |
| incident_active | tinyint(4) |
| incident_verified | tinyint(4) |
| incident_dateadd | datetime |
| incident_dateadd_gmt | datetime |
| incident_datemodify | datetime |
| incident_alert_status | tinyint(4) |
| incident_zoom | tinyint(4) |

Table 8.43: Table: incident_category

| Column name | Type |
| --- | --- |
| id | int(11) |
| incident_id | bigint(20) |
| category_id | int(11) |

Table 8.44: Table: incident_person

| Column name | Type |
|---|---|
| id | bigint(20) |
| incident_id | bigint(20) |
| person_first | varchar(200) |
| person_last | varchar(200) |
| person_email | varchar(120) |
| person_phone | varchar(60) |
| person_ip | varchar(50) |
| person_date | datetime |

Table 8.45: Table: location

| Column name | Type |
|---|---|
| id | bigint(20) |
| location_name | varchar(255) |
| country_id | int(11) |
| latitude | double |
| longitude | double |
| location_visible | tinyint(4) |
| location_date | datetime |

# Bibliography


[18291] Ieee standard computer dictionary: A compilation of ieee standard computer glossaries. *IEEE Std 610*, pages 1–217, Jan 1991.

[ADMR05] David Aumueller, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. Schema and ontology matching with coma++. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 906–908, New York, NY, USA, 2005. ACM.

[ANBI13] Pasinee Apisakmontri, Ekawit Nantajeewarawat, Marut Buranarach, and Mitsuru Ikeda. Towards the use of upper ontologies for refugee emergencies in disaster management. In *the Second Asian Conference on Information Systems*, 2013.

[ANBI15] Pasinee Apisakmontri, Ekawit Nantajeewarawat, Marut Buranarach, and Mitsuru Ikeda. Ontology construction and schema merging using an application-independent ontology for humanitarian aid in disaster management. In *Semantic Technology*, volume 8943 of *Lecture Notes in Computer Science*, pages 281–296. Springer International Publishing, 2015.

[BFBY14] Imen Bizid, Sami Faiz, Patrice Boursier, and JawahirCheMustapha Yusuf. Integration of heterogeneous spatial databases for disaster management. In Jeffrey Parsons and Dickson Chiu, editors, *Advances in*

*Conceptual Modeling*, volume 8697 of *Lecture Notes in Computer Science*, pages 77–86. Springer International Publishing, 2014.

[BLN86] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, December 1986.

[BMR11] Philip A Bernstein, Jayant Madhavan, and Erhard Rahm. Generic schema matching, ten years later. *Proceedings of the VLDB Endowment*, 4(11):695–701, 2011.

[CBB+12] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. The {SSN} ontology of the {W3C} semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17(0):25 – 32, 2012.

[CDSDS+06] M. Careem, C. De Silva, R. De Silva, L. Raschid, and S. Weerawarana. Sahana: Overview of a disaster management system. In *Information and Automation, 2006. ICIA 2006. International Conference on*, pages 361–366, Dec 2006.

[Dev] Devinit.org. Defining humanitarian assistance - global humanitarian assistance. http://www.globalhumanitarianassistance.org/data-guides/defining- humanitarian-aid.

[Div97] Operational Support Division. *Commodity Distribution.* United Nations High Commissioner for Refugees, 1997.

[DLKQ06] Dejing Dou, P. LePendu, Shiwoong Kim, and Peishen Qi. Integrating databases into the semantic web through an ontology-based framework. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 54–54, 2006.

[DM07] P. Di Maio. An open ontology for open source emergency response system. *Open Source Research Community*, 2007.

[Do06] Hong-Hai Do. *Schema matching and mapping-based data integration*. PhD thesis, 2006.

[DOL] Dolce : a descriptive ontology for linguistic and cognitive engineering. http://www.loa.istc.cnr.it/DOLCE.html. Accessed: 21/05/2013.

[DR07] Hong-Hai Do and Erhard Rahm. Matching large schemas: Approaches and evaluation, 2007.

[EG00] M.R. Endsley and D.J. Garland. *Situation Awareness Analysis and Measurement*. CRC Press, 2000.

[End95] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[ET] E-Teams. E-teams. http://www.nc4.us/ETeam.php.

[FG03] Ian Foster and Robert L. Grossman. Data integration in a bandwidth-rich world. *Commun. ACM*, 46(11):50–57, November 2003.

[FLGPJ97] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[Fuc09] Florian Fuchs. Semantic processing of monitoring data in industrial applications. In Jorge Cardoso and Miltiadis D. Lytras. Hershey, editors, *Semantic Web Engineering in the Knowledge Society*, pages 107–134. IGI Global, 2009.

[FZ11] Z. Fan and S. Zlatanova. Exploring ontologies for semantic interoperability of data in emergency response. *Applied Geomatics*, 3(2):109–122, 2011.

[G07] Nagypal Gabor. Ontology development. In Rudi Studer, Stephan Grimm, and Andreas Abecker, editors, *Semantic Web Services*, pages 107–134. Springer Berlin Heidelberg, 2007.

[GF95] Michael Grüninger and Mark S Fox. Methodology for the design and evaluation of ontologies. 1995.

[GHKR10] Anika Gross, Michael Hartung, Toralf Kirsten, and Erhard Rahm. On matching large life science ontologies in parallel. In Patrick Lambrix and Graham Kemp, editors, *Data Integration in the Life Sciences*, volume 6254 of *Lecture Notes in Computer Science*, pages 35–49. Springer Berlin Heidelberg, 2010.

[GPFL10] A. Gómez-Pérez and M. Fernández-López. *Ontological engineering : with examples from the areas of knowledge management, e-commerce and the semantic web*. Springer, 2010.

[GPFLC04] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004.

[GR11] J. Garrido and I. Requena. Proposal of ontology for environmental impact assessment: An application with knowledge mobilization. *Expert Systems with Applications*, 38(3):2462–2472, 2011.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[Gru95] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(56):907 – 928, 1995.

[GSY09] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Semantic matching. In *Encyclopedia of Database Systems*, pages 2561–2566. Springer, 2009.

[HBC13] G. Haddow, J. Bullock, and D.P. Coppola. *Introduction to Emergency Management*. Elsevier Science, 2013.

[HCL+10] Vagelis Hristidis, Shu-Ching Chen, Tao Li, Steven Luis, and Yi Deng. Survey of data management and analysis in disaster situations. *J. Syst. Softw.*, 83(10):1701–1714, October 2010.

[hDR02] Hong hai Do and Erhard Rahm. Coma - a system for flexible combination of schema matching approaches. In *In VLDB*, pages 610–621, 2002.

[HGXR09] Wen Hui, Gu Guona, Dai Xiaoliang, and Xu Ruzhi. An owl-based knowledge representation framework for searching semantically emergency preplan. In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, volume 3, pages 128–131, Aug 2009.

[HPnV06] Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. Textual entailment recognition based on dependency analysis and wordnet. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, volume 3944 of *Lecture Notes in Computer Science*, pages 231–239. Springer Berlin Heidelberg, 2006.

[HQ07] Wei Hu and Yuzhong Qu. Discovering simple mappings between relational database schemas and ontologies. In *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 225–238. Springer Berlin Heidelberg, 2007.

[IAS] IASC. Multi-cluster/sector initial rapid assessment (mira). https://docs.unocha.org/sites/dms/Documents/mira_final_version2012.pdf. Accessed: 24/02/2014.

[KLK06] E. Klien, M. Lutz, and W. Kuhn. Ontology-based discovery of geographic information services-an application in disaster management. *Computers, Environment and Urban Systems*, 30(1):102 – 123, 2006.

[KS14] Sharifullah Khan and Muhammad Safyan. Semantic matching in hierarchical ontologies. *Journal of King Saud University - Computer and Information Sciences*, 26(3):247 – 257, 2014.

[LLL+08] Xiang Li, Gang Liu, Anhong Ling, Jian Zhan, Ning An, Lian Li, and Yongzhong Sha. Building a practical ontology for emergency response systems. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 4, pages 222–225, Dec 2008.

[Mai07] Paola Di Maio. An open ontology for open source emergency response system, 2007.

[MBR01] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[MLR10] V. Mascardi, A. Locoro, and P. Rosso. Automatic ontology matching via upper ontologies: A systematic evaluation. *Knowledge and Data Engineering, IEEE Transactions on*, 22(5):609–623, May 2010.

[NDdMRGAM09] Ismael Navas-Delgado, Maria del Mar Roldán-García, and José F. Aldana-Montes. Semantic fields: Finding ontology relationships. In *Database and Expert Systems Applications*, volume 5690 of *Lecture Notes in Computer Science*, pages 427–434. Springer Berlin Heidelberg, 2009.

[NO95] Channah F. Naiman and Aris M. Ouksel. A classification of semantic conflicts in heterogeneous database systems. *J. Organ. Comput.*, 5(2):167–193, February 1995.

[OAM14] M.F.R. Opu, M.S. Arefin, and Y. Morimoto. Developing a framework for ontology generalization. In *Informatics, Electronics Vision (ICIEV), 2014 International Conference on*, pages 1–4, May 2014.

[OS99] A. M. Ouksel and A. Sheth. Semantic interoperability in global information systems. *SIGMOD Rec.*, 28(1):5–12, March 1999.

[PBR10] Eric Peukert, Henrike Berthold, and Erhard Rahm. Rewrite techniques for performance optimization of schema matching processes. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 453–464. ACM, 2010.

[Pet85] William J. Petak. Emergency management: A challenge for public administration. In *Public Administration Review*, volume 45 of *Special Issue: Emergency Management: A Challenge for Public Administration*, pages 3–7. Wiley, January 1985.

[Pin09] Filipe Pinto. Database marketing intelligence methodology supported by ontologies and knowlegde discovery in databases. 2009.

[PKR97] Ramesh Patil, Kevin Knight, and Tom Russ. Towards distributed use of large-scale ontologies. In *In AAAI Spring Symposium on Ontological Engineering*, 1997.

[PNL02] Adam Pease, Ian Niles, and John Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Working notes of the AAAI-2002 workshop on ontologies and the semantic web*, volume 28, 2002.

[Poj99] Judith Pojda. Administrative committee on coordination sub-committee on nutrition in united nations systems forum for nutrition. http://www.unsystem.org/scn/Publications/SCNNews/scnnews19.pdf, December 1999.

[Pro11] Sphere Project. *The Sphere Handbook, Humanitarian Charter and Minimum Standards in Humanitarian Response*. Practical Action Publishing, 2011.

[PS07] The Emergency Preparedness and Response Section. *Handbook for Emergencies*. United Nations High Commissioner for Refugees, UNHCR Headquarters, Case Postale 2500, Switzerland, 3 edition, 2007.

[PtPS03] Project Profile, the Population, and Geographic Data Section. *UNHCR Handbook for Registration*. United Nations High Commissioner for Refugees, 2003.

[RB01] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[RK08] KR Ratnam and DD Karunaratne. Application of ontologies in disaster management. 2008.

[Rod] Alexandre Rodichevski. Approximate string-matching algorithms, part 2. http://www.morfoedro.it/doc.php?n=223lang=en.

[RP04] S. Ram and Jinsoo Park. Semantic conflict resolution ontology (scrol): an ontology for detecting and resolving data and schema-level semantic conflicts. *Knowledge and Data Engineering, IEEE Transactions on*, 16(2):189–202, Feb 2004.

[RP05] R.G. Raskin and M.J. Pan. Knowledge representation in the semantic web for earth and environmental terminology (sweet). *Computers & Geosciences*, 31(9):1119–1125, 2005.

[Sah] Sahana. Sahana software foundation. http://sahanafoundation.org/.

[SE05] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. In Stefano Spaccapietra, editor, *Journal on Data Semantics IV*, volume 3730 of *Lecture Notes in Computer Science*, pages 146–171. Springer Berlin Heidelberg, 2005.

[SH04] Heiner Stuckenschmidt and Frank van Harmelen. *Information Sharing on the Semantic Web*. SpringerVerlag, 2004.

[SHM13] Sebastian Scheuer, Dagmar Haase, and Volker Meyer. Towards a flood risk assessment ontology  knowledge integration into a multi-criteria risk assessment approach. *Computers, Environment and Urban Systems*, 37(0):82 – 94, 2013.

[Sot07] Mandana Sotoodeh. *Ontology-Based Semantic Interoperability in Emergency Management*. PhD thesis, The University of British Columbia, July 2007.

[SPKR96] Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ. Toward distributed use of large-scale ontologies. In *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, pages 138–148, 1996.

[SSSS01] Steffen Staab, Rudi Studer, Hans-Peter Schnurr, and York Sure. Knowledge processes and ontologies. *IEEE Intelligent systems*, 16(1):26–34, 2001.

[SUM] Suggested upper merged ontology (sumo). http://www.ontologyportal.org/. Accessed: 21/05/2013.

[TA11] Joseph H.M. Tah and Henry F. Abanda. Sustainable building technology knowledge representation: Using semantic web techniques. *Advanced Engineering Informatics*, 25(3):547 – 558, 2011. Special Section: Engineering informatics in port operations and logistics.

[TB08] Ajay Tripathi and Hassan A. Babaie. Developing a modular hydrogeology ontology by extending the {SWEET} upper-level ontologies. *Computers Geosciences*, 34(9):1022 – 1033, 2008.

[UG96] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(02):93–136, 1996.

[UJ99] Mike Uschold and Robert Jasper. A framework for understanding and classifying ontology applications. In *IJCAI'99 Workshop on Ontology and Problem Solving Methods: Lessons Learned and Future Trends*, 1999.

[UK95] Mike Uschold and Martin King. Towards a methodology for building ontologies. In *In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, 1995.

[Ush] Ushahidi. Ushahidi. http://www.ushahidi.com/.

[Web] WebEOC. Webeoc incident management. http://www.esi911.com/home.

[Wik] Wikipedia. Upper ontology. https://en.wikipedia.org/wiki/Upper$_{o}$*ntology*.

[XZ07a] W. Xu and S. Zlatanova. Ontologies for disaster management response. *Geomatics Solutions for Disaster Management*, pages 185–200, 2007.

[XZ07b] Wei Xu and Sisi Zlatanova. Ontologies for disaster management response. In Jonathan Li, Sisi Zlatanova, and AndreaG. Fabbri, editors, *Geomatics Solutions for Disaster Management*, Lecture Notes in Geoinformation and Cartography, pages 185–200. Springer Berlin Heidelberg, 2007.

[ZZNJ02] Dongsong Zhang, Lina Zhou, and Jay F. Nunamaker Jr. A knowledge management framework for the support of decision making in humanitarian assistance/disaster relief. *Knowledge and Information Systems*, 4(3):370–385, 2002.

# Publications

## Journal

[1] Pasinee Apisakmontri; Ekawit Nantajeewarawat; Mitsuru Ikeda ; and Marut Buranarach (2015). "An ontology-based framework for semantic reconciliation in humanitarian aid in emergency information systems", Journal of Information Processing (Accepted).

## Conferences

[2] Pasinee Apisakmontri; Ekawit Nantajeewarawat; Marut Buranarach; and Mitsuru Ikeda (2015). "Ontology construction and schema merging using an application-independent ontology for humanitarian aid in disaster management", Lecture Notes in Computer Science: Semantic Technology: Proceedings of the 4th Joint International Semantic Technology Conference (JIST 2014), 9-11 November 2014, Chiang Mai, Thailand, Vol. 8943 LNCS, 2015, pp. 281-296.

[3] Pasinee Apisakmontri; Ekawit Nantajeewarawat; Marut Buranarach; and Mitsuru Ikeda (2013). "Towards the use of upper ontologies for refugee emergencies in disaster management". In Proceedings of the Second Asian Conference on Information Systems (ACIS 2013) [CD-ROM], 31 October 2 November 2013, Phuket, Thailand, pp. 153-160.