Software Tool for Semantic Resources Allocation in Humanitarian Crises

Aladdin Shamoug
United Nations Development Programme
aladdin.shamoug@undp.org

Radmila Juric
HSN University College, Norway
rju@hbv.no

Abstract

Resource Allocation (RAlloc) is one of the most important tasks in organizing humanitarian response to humanitarian crises. It is not only that adequate and efficient RAlloc save lives and reduce damages caused by humanitarian crises, but RAlloc must be fast and efficient to save time and resources. Given that RAlloc is a type of a decision making process, it is expected that decision on RAlloc are based on accurate and relevant information generated at various stages of humanitarian response. In this paper we promote Semantic Resource Allocation (SemRAlloc) tool which a) collects and interprets the semantics of an environment where RAlloc is required and b) the reasons upon the semantics of that environment in order to make appropriate RAlloc. The tool is built with computations based on SWRL enabled OWL ontologies. The prototype has been implemented as a desk-top application which can also run in mobile/wireless environments, including Android smart phones.

1. Introduction

Hazardous situations in human environments, with which we are not prepared to deal, may end in humanitarian crises (HC), ranging from drought, flooding, famine and hurricanes to wars, earthquakes and volcanic eruptions [1]. The correct information, which is either collected or created during HCs, is essential in managing humanitarian response (HR), saving lives and bringing such environments back to their normal state [2]. Many countries cannot deal with HC. They appeal to the international community to intervene and help in mitigating the catastrophe [3]. There are various ways of addressing HC, but making a correct and prompt decision on WHO is going to intervene in HC, WHY, WHEN and HOW they can help in HC, remains one of the most crucial and complex tasks in HR. RAlloc often takes longer than expected [4, 5, 6, 7] and its efficacy depends on information availability and reliability [8, 9]. One of the most difficult tasks for decision makers in HR is to find appropriate humanitarian organizations to respond to HC. For every HC a list of requirements is different and needs in HC might not suit all humanitarian organization profiles.

The role of coordinating body during HR is very important because they act as a hub connecting:

- Actors, such as Donors (D), Agencies (A) and Implementing Partners (P)), which provide help,
- Sectors (S) which refer to the technical services provided by actors to affected populations, such as food, water, health, shelter, education and
- Locations (L) which refer to the place where the affected population live.

In efficient RAlloc, the coordinating body uses its accumulated human experience to reason upon Actors, Sector and Location and perform RAlloc by selecting the best possible combination of A and S to work with each other in a particular location L. Given that RAlloc is a complex process, where decisions are taken under enormous pressure of time and resources, HR must be tailored to a particular HC and consider cultural, political, organizational practices and regulations. Having information systems in place, with well-structured repositories and the exchange of their contents in place, is one of the prerequisites for informed RAlloc [10,11,12,13,14,15]. Constant changes in HC, which might happen on hourly basis, are also important to capture. Decision makers in RAlloc rely on such changes rather than waiting for accurate and exact information, which may come late, or not at all [16]. RAlloc must be progressive, iterative, interpret *what is going on* rather than *what should be done* [17]. It should collect data from people/machines, support data planning/tasking and provide data analysis [18, 19, 20, 21].

1.1. The Problem to Resolve: Creating an Automated Tool for SemRAlloc

In this paper we propose a software tool with slightly different characteristics compared to other tools in decision making. The tool enables:

a) continuous support in RAlloc from day 1 of HC and assistance in decision making, regardless how much information is available;

HICSS

b) the collection and interpretation of the semantics of an environment where RAlloc is required

c) the reasoning upon the semantics of that environment to make appropriate RAlloc.

The emphasis is on the understanding the meaning of available (and constantly changing) information and very quick decision making based on it. It is obvious that SemRAlloc sits between a range of data repositories with various information relevant to the HR co-ordination body. Urgent decisions on RAlloc vary from day to day and always focus on urgent issues on a particular day / time of HC [15]. The tool also allows the input of various types of information from heterogeneous repositories and gives provision for accepting data from media and social media.

The idea of building SemRAlloc which manages the semantic of environment struck by HC, as itemized in a)-c), requires a new computational model, which powers it. If we used solutions from the Artificial Intelligence (AI) filed for managing the semantics of RAlloc, we will build too complex software product. It will be difficult to scale run it in constantly changing environments of HC. We need a sleek and easy to install/operate software tool, which addresses these changes by focusing on various moments in HR where decisions on the RAlloc are to be made, sometimes on an ad-hoc basis. Instead of building a demanding expert system to manage the semantics of RAlloc, we deliver the same functionality by creating a modern application, which run on mobile and smart devices in HR.

The computational model which can satisfy a)-c) should be built with Semantic Web Technologies (SWT) [22] and their languages OWL and SWRL [23,24]. SWT still evolves, but has been used in applications for manipulating the semantics of the Internet. Computational models based on SWT are rather new. There are not so many developed outside semantic web, but they are efficient from two perspectives: (i) they can address constant changes in environments where such SE solutions are needed [25,26,27,28] (ii) they can perform reasoning without using computationally demanding AI algorithms. The computational model, which powers SemRAlloc, enabled us to: manipulate the semantics of RAlloc and create a portable/scalable software product.

This paper is organized as follows. Section 2 introduces SemRAlloc's software architectural model, gives a scenario where the tool can be used and sets up an environment where we can perform RAlloc. Section 3 explains computations which power the tool in terms of describing main OWL concepts and a detailed reasoning process which is performed upon them. Section 4 illustrates the functionality of the tool through an example of HC in

the Scenario and user interfaces, which are essential for understanding SemRAlloc. In Conclusions we debate the commercialization of the tool and its reusability in any type of HR.

2. The SemRAlloc

2.1. Software Architecture for SemRAlloc

The reader should note three important aspects of **SemRAlloc.** Firstly, the inputs to the tool are available from any source of information, which may include existing data from structured repositories such as databases, to information generated by media, social media and anyone involved in HR. Secondly, the tool converts these inputs into OWL concepts and, through them, it interprets the semantics of the environment where RAlloc is required. Information on A,S,L has been created and it is ready for performing reasoning upon them in order to support decision making for RAlloc. Thirdly, the tool uses a set of interfaces for human intervention and involvement in the process of reasoning performed by the tool. The user of the tool is in a position to control inputs to the tool and manipulate the reasoning process for RAlloc decisions - if necessary.
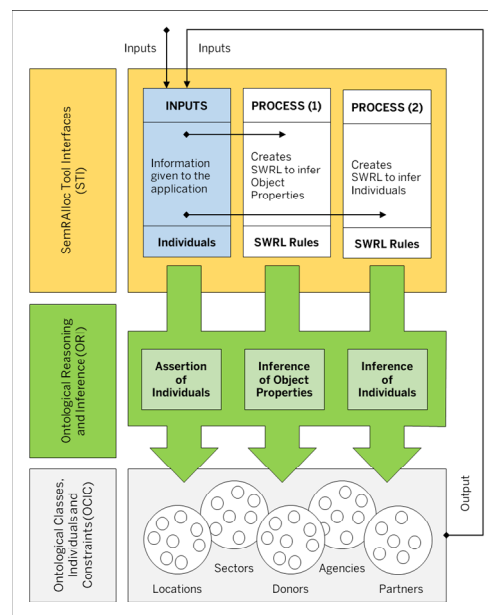

Figure 1. Software architecture of SemRAlloc

Fig. 1. shows the main layers of SemRAlloc architectural model. The tool uses data repositories, from the Ontological Classes, Individuals, and Constraints (OCIC) layer, which are SWRL enabled OWL ontologies. They store the semantic of the

environment affected by HC in OWL concepts: classes, individuals and constraints. Constraints have a format of either data type or object properties.

Ontological Reasoning and Inference Layer (ORI) stores SWRL reasoning rules which manipulate the semantic from the OCIC layer. These SWRL rules are one part of the computational model, which powers SemRAlloc. Their role is to assert individuals into the OWL ontologies, infer object properties upon OWL concepts defined in the ontologies and infer individuals as the final result of reasoning. These inferred individuals are the main source of information for decision making and RAlloc.

SemRAlloc Tool Interfaces (STI) layer illustrates the user interaction with the tool. We show that (1) certain types of data must be inserted into the tool (INPUTS), (2) inference should define object properties (PROCESS 1) and (3) reasoning should be performed in order to have efficient decision making for RAlloc (PROCESS 2).

It is obvious that decision making is performed through steps of reasoning with SWRL rules. They may run either individually or as a chain, one after another. The output of one reasoning step may become an input to the next one, which is illustrated as a line connecting OCIC and STI layers in Figure 1.

2.1. The Scenario

In April 2015, a sudden earth quake hit Nepal (location L1). Media reports show massive destructions, deaths, causalities and damages in the country's infrastructure. Some humanitarian agencies (A1, ..., An) have become aware of the problem and analyzed the emergency of the situation in Nepal. At the same time various donors (D1, ..., Dm), have also become aware of the crises in Nepal and are assessing the possibilities of getting involved. The media information is not sufficient for organizing any HR. Agencies may try to approach other sources of information in Nepal; donors may make their own investigations on HC, but would prefer to wait for agencies to appeal for assistance. A coordination body, responsible for organizing HR, is in the same situation: they have to start their own investigations before they make decisions. They can liaise with the government of Nepal, its bodies, UN agencies, and local NGOs, that have constant presence in Nepal. The information needed is not necessarily historical. It may be the result of the HC itself. This information might be known to Nepalese government, local authorities, UN agencies in Nepal, and NGOs, and must be made available to the coordination body.

2.2. Competency Questions for the Scenario

The scenario raises a number of questions, which should be answered when managing RAlloc. These answers find out who are the most suitable Donor(s), Agencie(s) and Implementing Partner(s) to provide the needed sectors in crisis location(s). They will make RAlloc feasible and more accurate.

We use term Competency Questions (CQ) according to the vocabulary of the SWT. However, these CQ are identical to questions the coordinating body may have in HR before an appropriate RAlloc is decided. Therefore the tool must answer the questions. On the other side, in our computational model, CQs trigger the definition OWL concepts and creation of SWRL rules, which perform reasoning and answer the CQ. Table 1 lists the CQ from the Scenario. In our implementation, these CQ may either be chosen from the drop down menu, available in the tool, or entered by the user.

Table 1. Competency questions

1	Which donors and agencies are interested in the crisis location L1 (Nepal)?
2	Which sectors appear in the location crises L1?
3	Which donors from Q1 are willing to fund which agencies from Q1 and become possible donors and agencies in this crisis?
4	Which sectors are now confirmed sectors in crisis location L1?
5	Which donors from Q3 are confirmed that they will fund agencies form Q3?
6	Which agencies are confirmed to be involved in a combination of Location/Sector (L/S)?
7	Which implementing partners will receive resources from agencies in order to allocate these resources to a combination of L/S?

3. Computations in the SemRAlloc

This section describes how the computational model works. The ontological classes are derived from the Scenario and the semantic of the information found in a particular stage of the HC. They are shown in Figure 2. They store the semantics of all actors involved: Donors, Agencies and Implementing Partners, and include locations and sectors affected by the HC. RESULTS class stores the results of the reasoning process.

When making decisions in HC, as a part of RAlloc, we may categorize donors/agencies/ implementing partners as: (i) possible, because they have either shown their interest for or are suitable as actors in HR and (ii) confirmed because they are chosen and agreed to be involved in HR. The same logic applies to locations and sectors: certain locations have been affected and sectors might have been identified, but they should also be confirmed.

3.1. Ontological Classes and Individuals

SemRAlloc ontological model from Figure 2 shows that the tool stores information on Location, Sectors, Agencies, Donors and Partners. Some of them can be "possible" or "interested" and some of them will become "confirmed". Figure 2 also shows two important characteristics of the model.

Firstly, it allows us to have all possible donors / agencies / partners / locations / sectors as ontological individuals, and leave their descriptions and relationships to become constraints (properties). Properties can be dynamically defined and inferred as the situation in HC changes from day to day. Therefore the relationships between actors, sectors and locations (A,S,L) is changeable The model from Figure 2 is generic, reusable for HC and relatively simple to develop from the software engineering point of view. It produces a stable and flexible software tool, which will address changes in HC through OWL constraints and not through the complexity of OWL taxonomical structure.
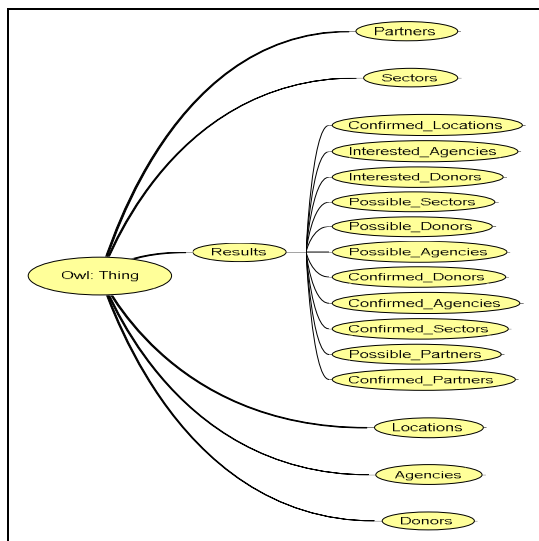

Figure 2. The ontological classes

Secondly, the model from Figure 2 allows to address the volatility of situations in HC, where *everything might be possible*. Sectors affected by HC might appear suddenly, and might not be allocated to a specific location. We might know that there is a need for water, health and shelter, but which one of them is needed at which location might not be clear. More and more agencies will appear on a daily basis, showing interest in current locations and sectors, but they need to have committed donors to support their interest. However strong their desire to get involved in HR is, some agencies will never get an initial

commitment from some donors. They may not become "confirmed". Some donors may change policies and fund agencies even if it is against their regulations. The volatile relationship between actors/ locations/sectors is not a part of OWL model. It is INFERRED through constraints as situations in HC change. This is one of the most important characteristics of the tool.

3.2. Ontological Constraints

Table 2 shows a possible set of constraints inferred on the OWL concepts from Figure 2. The first column in the table identifies a reasoning rule which is used for inferring the constraint, and the last shows where our results of reasoning are stored. These constraints are derived from the scenario and CQs. All of them are **object proprieties** except for the first rule, which uses **data property** "has_crisis" for Locations class (it should meets "Yes" condition).

Not all individuals from a domain class In Table 2 are related through object properties to individuals of another class. The choices of individuals which participate in relationships through constraints may be defined manually (assertion) and automatically (inference), depending on a situation in HC.

Table 2. The object properties

	Domain Class	Data/Object Property	Range Class	Result Class
1	Location	hasCrisis	"Yes, No"	Confirmed Location
2	Agency	Interested In	Confirmed Location	Interested Agency
3	Donor	Interested In	Confirmed Location	Interested Donor
4	Sector	Needed In	Confirmed Location	Possible Sector
5	Interested Donor	Willing To Fund	Interested Agency	Possible Donor
				Possible Agency
6	Possible Sector	Provided By	Possible Agency	Confirmed Sector
				Confirmed Agency
7	Possible Donor	Has To Fund	Confirmed Agency	Confirmed Donor
8	Partner	Interested In	Confirmed Location	Interested Partner
9	Interested Partner	Has Experience In	Confirmed Sector	Possible Partner
10	Possible Partner	Is Hired By	Confirmed Agency	Confirmed Partner

Readers should also note that object properties given in Table 2 are important for ontological

matching. Without their correct inference we cannot perform reasoning. In other words, object properties from Table 2 are basic constraints for running reasoning rules upon constantly changing individuals of actors/ location/ sectors in the ontology.

3.3. Reasoning within the SemRAlloc

We illustrate the ontological reasoning through 10 SWRL rules which give answers to the CQ listed in Table 1. These 10 rules infer individuals in the ontology. The order of their execution is shown in Figure 3, which illustrates which classes of the OWL model from Figure 1 are involved in which reasoning rule(s). It is important to note that these 10 SWRL rules are supported by additional rules, which infer object properties given in Table 2, before each of the main 10 rules are run. These supporting rules do not define the reasoning process because they infer constraints and thus we do not show them in Fig. 3.
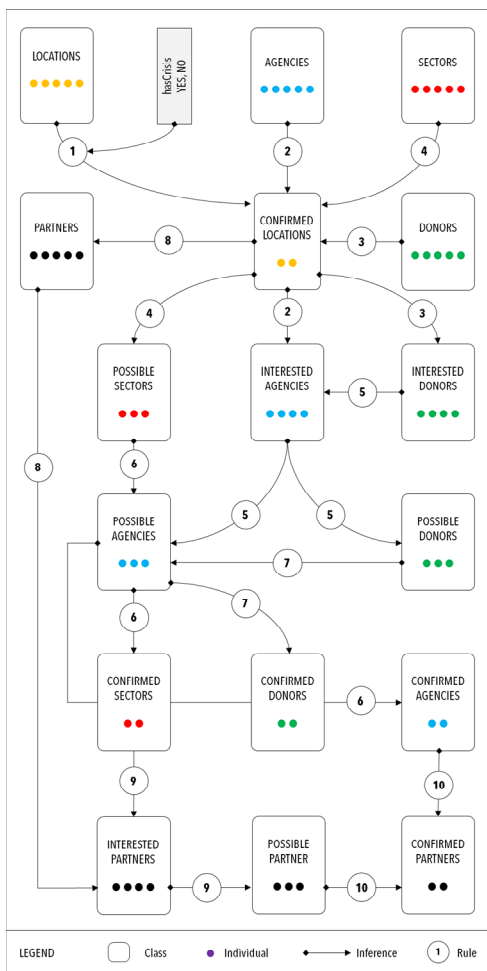

Figure 3. The reasoning process

In the next section, we illustrate each of these 10 SWRL rules with (a) its syntax, (b) a set of OWL classes involved in the reasoning and classes which store its results and (c) screen shots of individuals of RESULT subclasses which have answers to the CQ.

We may perform reasoning upon classes, which store inferred individuals from previous reasoning, which are subclasses of the RESULT classes. Therefore, we follow the principles of tool's software architectural model from Figure 1.

4. Running SemRAlloc

The way SemRAlloc answers CQ, and assists in making appropriate RAlloc in our scenario context is shown through a set of 10 User Interfaces (UI), which are software components from the STI layer in Fig. 1. These UI, named as STI1 – STI10 are in Fig. 4-13. We describe the content of each of STI separately in order to show the functionality of the SemRAlloc, the way reasoning is performed and how CQs are answered.

STI1 in Figure 4 captures inputs from the users of SemRAlloc. The user enters "Nepal" for *crisis location* and "Nepal" becomes an ontological individual of LOCATION class.

Figure 4. Reasoning upon crisis locations (STI1).

As mentioned in 3.3 before we run the rule, which infers individuals, we have to run a supporting rule which infers properties. In STI1, where we run the first SWRL rule R1 for establishing that Nepal is a confirmed location for the HC, we have to run a supporting rule R1A which infers data type property hasCrisis. This means that within the ontology we will store the information that "location Nepal has crises" (YES for data type hasCrisis property values). The result of inference o individuals is that Nepal becomes a confirmed location for this HC. Rule R1 corresponds to software components stored within PROCESS (2) and Rule R1A corresponds to software components stored within PROCESS (1) of the architectural model in Figure 1. At this moment,

SemRAlloc has not answered any CQ yet. It has equipped the OWL ontology with initial information inputted by the user: "Nepal becomes a confirmed location for this HC".

STI2 from Figure 5 shows that (a) The user makes a selection of Agencies, from the dropdown menu, which might be interested in intervention in Nepal. This information is either known to the user or collected from the agencies. In both cases the information can be automatically entered into SemRAlloc. (b) The selection of these agencies trigger the inference of Interested_In object property (SRWL Rule 2A). (c) SWRL Rule 2, available in the lower part of STI2, reasons upon individuals of Agency class and infers its individuals into the Interested_Agency class, if they are connected through Interested_In object property with Nepal individual of Confirmed_Location class.
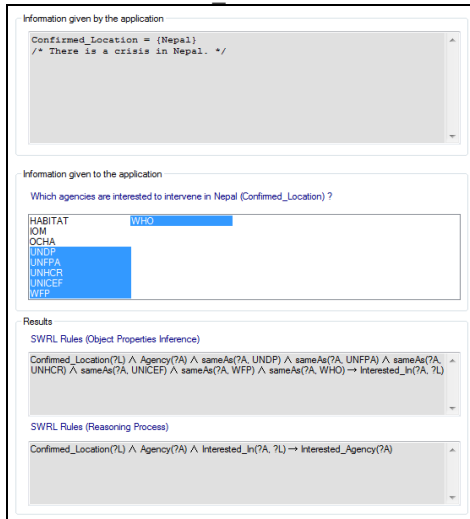

Figure 5. Reasoning upon confirmed locations and agencies (STI2)

Therefore, Rule R2 stores UNDP, UNFPA, UNHCR, UNICEF, WFP and WHO in the Interested_Agency class. This is a partial answer to CQ1. R2 corresponds to software components stored within PROCESS (2) and Rule R2A corresponds to software components stored within PROCESS (1) of the architectural model in Figure 1.

STI3 from Figure 6 shows reasoning upon Donors and Confirmed Locations. This is very similar to STI2, but instead of finding Agencies interested in Nepalese crises, this reasoning will find Donors who will be interested to fund Nepalese HC. Therefore (a)-(c) above can be repeated for STI3. Rule 3 stores DFID, ECHO, EU and USAID individuals in the Interested_Donor class. Rule R3 corresponds to software components stored within PROCESS (2) and R3A corresponds to software components stored

within PROCESS (1) of the architectural model in Figure 1. Rules R2 and R3 give a full answer to CQ1.
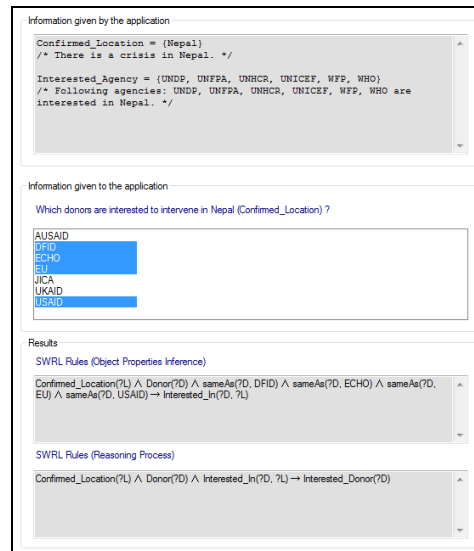

Figure 6. Reasoning upon confirmed locations and donors (STI3).

In Figure 7 we have STI4 which deals with another type of user's input. At this stage we would like to know if the user has any information on sectors needed for HR in Nepal. Therefore, a dropdown menu appears and the user chooses sectors which he/she knows that is needed. This information can change and we can run the rules associated with STI4 as many times as needed, is the situation in HC changes. In other words if sectors change then we can come back to STI4 and chose different sectors.
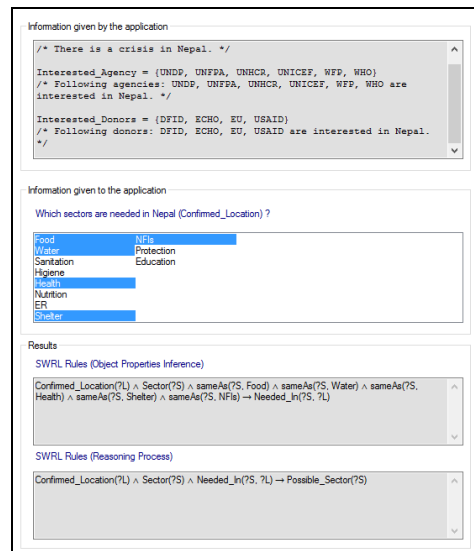

Figure 7. Reasoning upon confirmed locations and sectors (STI4).

The choice of sectors triggers two rules as in STI2 and STI3. Firstly, we infer object property Needed_In between sectors and conformed location classes through Rule R4A. Secondly Rule R4 infers that sectors needed in Nepalese HC are Food, Water, Health, Shelter and NFI and therefore these individuals are inferred into Possible_Sector class. Rule R4 corresponds to software stored within PROCESS (2) and Rule R4A corresponds to software stored within PROCESS (1) of the architectural model in Figure 1. Rule R4 answers CQ2.

In STI5, given in Figure 8, we show how interested agencies and donors are becoming possible agencies and possible donors using Rule (R6) available in the lower part of Figure 8. However, before we can run the rule we need to infer a couple of additional object properties named "Willing_To_Fund". They are automatically generated after the user connects Interested_Agency with Interested_Donor using a dropdown menu in the upper part of STI5. This information is supplied using the same principles as in STI1 and STI3. The user may either have a viable information at the time or the information may come from donors' and agencies' or their historical data. In STI5 we have 6 inference rules labelled as R5A. All of them could be seen by scrolling down the scroll bar. Rule R5 answers CQ 3.
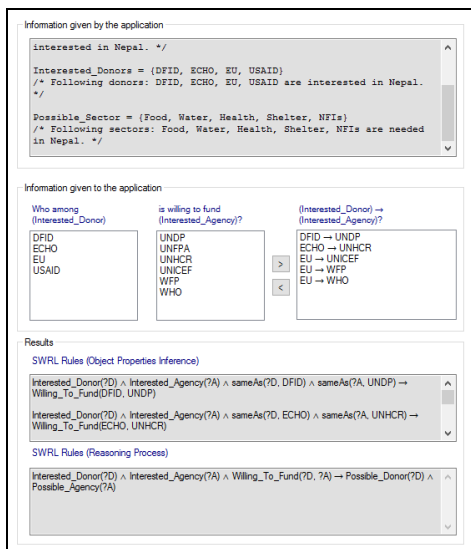

Figure 8. Reasoning upon interested donors and interested agencies (STI5).

Figure 9 shows STI6 and illustrates that (i) The user makes a pairing between individuals in Possible_Agency and Possible_Sector classes, which appear in dropdown menus. This information is either known to the user or collected from the agencies. In both cases the information can be automatically fed into SemRAlloc. (ii) The pairing of these agencies and sectors trigger the inference of Provided_By object property (SRWL Rule R6A). In STI6 we have 6 inference rules labelled as R6A. All of them could be visible if we scroll down the scroll bar. (iii) SWRL Rule R6, available in the lower part of STI6, reasons upon individuals of Possible_Agency and Possible_Sector classes and infers individuals of both classes into the Confirmed _Agency and Confirmed_Sector classes, if they are connected through Provided_By object property. Rule R6 answers both CQ 4 and CQ 6.
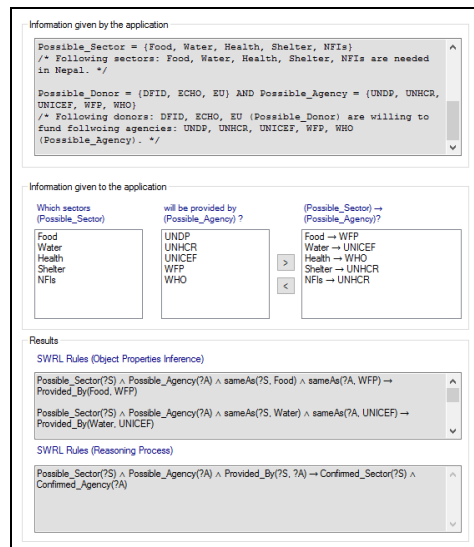

Figure 9. Reasoning upon possible sectors and possible agencies (STI6).

STI7 in Figure 10 has similar reasoning to STI6. The user pairs individuals in Possible_Donor and Confirmed_Agency classes, which triggers a few properties Has_To_Fund (visible when using a scroll bar). These donors become "confirmed" as a result of pairing through R7: they will have to fund confirmed agencies for chosen sectors. Rule R7 answers CQ5.

STI8 in Figure 11 is similar to STI2 and STI3. It shows that (a) The user makes a selection of Implementing Partners, from the dropdown menu, which might be interested in intervention in Nepal. This is either known to the user or collected from the partners and can be automatically fed into SemRAlloc. (b) The selection of partners triggers the inference of Interested_In object property (SRWL Rule R8A). (c) SWRL Rule R8, available in the lower part of STI8, reasons upon individuals of Partner class and infers individuals into the Interested_Partner class. Rule R8 answers partially CQ7.
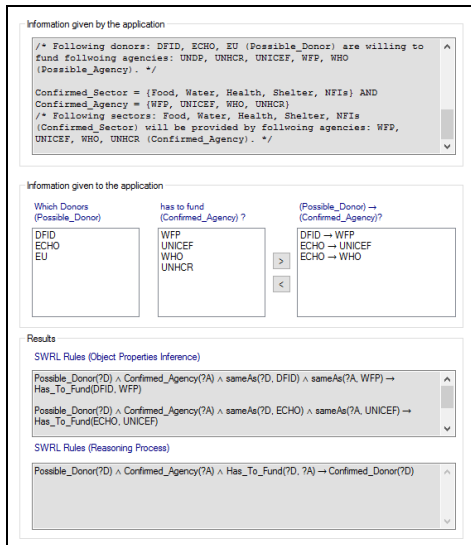
Figure 10. Reasoning upon possible donors and possible agencies (STI7).

STI8 in Figure 11 is very similar to STI2 and STI3. It shows that (a) The user makes a selection of Implementing Partners, from the dropdown menu, which might be interested in intervention in Nepal. This information is either known to the user or collected from the partners and can be automatically fed into SemRAlloc. (b) The selection of these partners triggers the inference of Interested_In object property (SRWL Rule R8A). (c) SWRL Rule R8, available in the lower part of STI8, reasons upon individuals of Partner class and infers individuals into the Interested_Partner class, if they are connected through Interested_In object property. Rule R8 answers partially CQ7.

STI9 in Figure 12 is very similar STI6. The user makes a pairing between Interested_Partner and Confirmed_Sector, which appear in the dropdown menus. The pairing of these partners and sectors triggers the inference of Has_Experience_In object property (SRWL Rule R9A). In STI9 we have 5 inference rules labelled as R9A. All of them could be visible if we scroll down the scroll bar. SWRL Rule R9, available in the lower part of STI9, reasons upon individuals of Interested_Partner and Confirmed_Sector and infers individuals of Interested partners into Possible partners, if they are connected through Has_Experience_In object property.

STI10 in Figure 13 is similar to STI9. The user makes a pairing between Possible_Partner and Confirmed_Agency, which appear in the dropdown menus. The pairing of these partners and agencies triggers the inference of Is_Hired_By object property (SRWL Rule R10A). In STI10 we have 5 inference

rules labelled as R10A. All of them could be visible if we scroll down the scroll bar. SWRL Rule R10, available in the lower part of STI10, reasons upon individuals of Possible_Partner and Confirmed_ Agency and infers individuals of Possible_Partner into Confirmed_Partner. Rule R10 answers CQ 7.
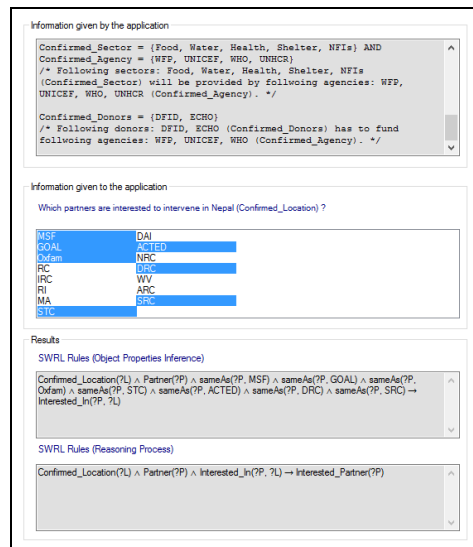

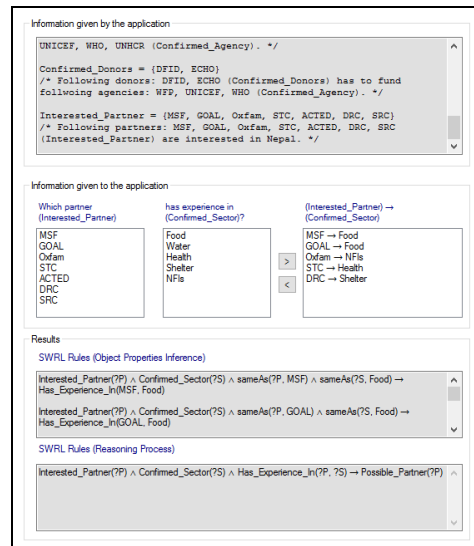**Figure 11. Reasoning upon confirmed locations and partners (STI8).**

Figure 12. Reasoning upon partners and confirmed sectors (STI9).

5. Conclusions

We propose SemRAlloc tool, which automates RAlloc during HC, by exploiting the semantics of the environment where HC occurs and reasoning upon it in order to assist in decision making for HR. At the

time of writing this paper we could not find any tool which can be used dynamically for RAlloc, utilize the power of SWT, manipulate the semantics found in various stages of HR for decision making in RAlloc, address constant volatility and changes in these environments and involve users in decision making at the same time. Therefore we were not able to find any related work for this paper. Software tools used in HR for RAlloc offer only a fraction of SemRAlloc's functionality and none of them use reasoning. The novelty of this work is fourfold.
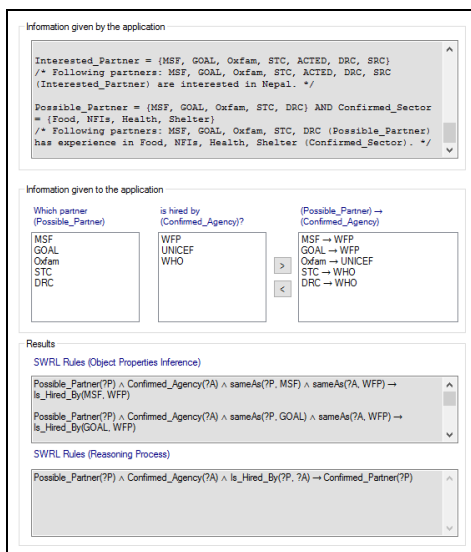

Figure 13. Reasoning upon possible partners and possible agencies (STI10).

Firstly, SemRAlloc captures every moment of HC from day one, makes decisions based on the information we may have at the moment we run the tool, i.e. answers to the questions we may have in RAlloc are applicable to the moment we collect semantics and reason upon it. Therefore the tool will not create any historical data in decision making because it will have to manipulate semantic which appear at the certain moment in HC.

Secondly, the changing nature of HC requires that the tool "reacts" to these changes. We addressed this in the computational model. As situations change in HC, we are able to infer constraints in OWL ontology dynamically, according to each situation. The reusable ontological model form Fig. 2, gives a sound and stable software solution for building SemRAlloc.

Thirdly, SemRAlloc offers improvements in decision making though reasoning. The selection of rules we ran in section 4 is not the only viable set of reasoning steps. Users of the tool can tailor their way of reasoning according to CQs they may have. Our rules are influenced by the selection of object properties from Table 2 and different inference of

object properties might result in different types of SWRL rules being available. We can write them in advance, and even on an ad-hoc basis. SemRAlloc runs any number of SWRL rules required by a situation in HC and may repeat any rule at any time. The user chooses SWRL rules and possible order we run them. UIs and its drop down menus within STI components guide the user how to use the tool.

Finally, the rigid exclusion logic the tool follows is very important for tailoring the tool for a particular situation in HR. We start with a specific HC in a geographical location (L1 is Nepal) and exclude all other HC: The same logic is used in all STI modules of the software tool. Initially we exclude all agencies, donors and partners and keep only *interested* ones. On the next round we excluded all except *possible* agencies, donors, sectors and partners. Finally we excluded all except *confirmed* agencies, donors, sectors and partners. We kept only individuals which are inferred to "confirmed" result classes. The confirmed results will be used by a coordination body for RAlloc. The results of all computation processes, from STI1 to STI10, is shared with concerned parties to better utilize their resources and speed up decision making process.

SemRAlloc has been implemented as a web application in J2E and .NET. Figure 1 was used as the starting point in the implementation and thus the commercialization of this tool is feasible. If we wish to run it on hand held smart devices, then it should be Android environments because we can reuse our architectural solution from Fig. 1 in Android Studio and the App generated from SemRAlloc will have good performance results.

Our future work includes inputting relevant unstructured data into the tool, such as live data generated in social media. We have done experiments for feeding Twitter data into Hadoop and used its SQL queries to feed our OWL model. In spite of having abilities to process enormous amount of individuals in OWL, SemRAlloc is not a replacement for tools which manipulate Big Data in HC, because it was created for a different purpose: It does not hold historical data and it deals with a particular moment in HC where decisions on RAlloc are taken. Bridges between structured/semi-structured data and OWL ontologies were developed a decade ago and would not required the restructuring of SemRAlloc model from Fig. 1, if we wish to add Twitter data to SemRAlloc. Our SemRalloc has a stable software model and generic architecture and does not run complex computations: 10 SWRL rules in our example run efficiently and give fast response time. The number of individuals in OWL has no impact on SemRAlloc performance, but the number of inferred

properties might have, which may affect SemRalloc for Android devices.

12. References

[1] Sphere Project, Sphere Handbook (2010), Humanitarian Charter and Minimum Standards in Disaster Response, Sphere Project, Geneva, Switzerland.

[2] UN, United Nations, Global Symposium+5: Information for Humanitarian Action, Geneva, Switzerland 22-26 October 2007. Geneva: United Nations.

[3] HPG, Humanitarian Policy Group (2006) "Humanitarian Response to Natural Disasters" HPG Briefing Paper, May 2006, ODI: London.

[4] FEWSNET, Famine Early Warning Network (2001) "Worsening Crisis in Sudan" Food Security Update, USAID Publications.

[5] Charny, J. R. (2004) "Upholding Humanitarian Principles in an Effective Integrated Response" Ethics International Affairs, Vol. 18, pp. 13-20.

[6] HPG, Humanitarian Policy Group (2005) "Humanitarian Issues in Niger" HPG Briefing Note, Overseas Development Institute, ODI: London.

[7] Margesson R. and Taft-Morales M. (2010) "Haiti Earthquake: Crisis and Response" CRS Report for Congress, Congressional Research Serv.: Washington D. C.

[8] Morrison, C. T., and Cohen, P. R., (2005) "Noisy information value in utility-based decision making" Proc. of the 1st Int. workshop on Utility-based data mining, pp.34-38, August 21-21, Chicago, US

[9] Bharosa, N. and Janssen, M., (2009) "Reconsidering information management roles and capabilities in disaster response decision-making units" Proc. of the 6th Intern. ISCRAM Conference, May, Gothenburg, Sweden 2009.

[10] Shamoug, A. and Juric, R., (2012) "Framework for Knowledge Inference in Decision Making during Humanitarian Crises", Workshop on Knowledge Manag. Capacity in Africa, KMCA2012, Khartoum, Sudan

[11] Shamoug, A., and Juric, R., (2911) "Addressing Interoperability Through the Semantics of Information Highway in Managing Responses in Humanitarian Crises", Proc. of the 8th Int. ISCRAM conf., Lisbon, Portugal.

[12] Shamoug, A., Juric, R., and Paurobally, S. (2012) "Ontological Reasoning as a Tool for Humanitarian Decision Making", Proc. of the 9th International ISCRAM, conference, Vancouver, Canada, 2012.

[13] Shamoug, A., and Juric, R. (2011) "Decision Making in Humanitarian Crises Based on OWL/SWRL Enabled Ontologies", Proc. of SDPS 2011 Conference, Korea.

[14] Shamoug, A., Juric, R., Paurobally, S. (2012) "Semantic Modelling of Decision Making for HR to Crises", Proc. of SDPS 2012, Berlin, Germany.

[15] Shamoug, A., Juric, R., and Paurobally, S. (2014) "Semantic Representations of Actors and Resource Allocation through Reasoning in Humanitarian Crises", Proceedings of the 47th HICSS, Hawaii, USA.

[16] Kari, J., (1998) "Making Sense of Sense-Making: From Metatheory to Substantive Theory in the Context of Paranormal Information Seeking", Workshop on Theoretical Stands in Studying Library and Inform. Institutions, Oslo, Norway.

[17] Pollack-Johnson, B., and Liberatore, M.J. (2006) "Incorporating Quality Considerations into Project Time/Cost Tradeoff Analysis and Decision Making", IEEE Trans. on Engineering Managm. 53(4), 2006, pp. 534-542.

[18] Kozlowski, S.W.J., and Klein, K.J. (2000) A Multilevel Approach to Theory and Research in Organizations, Jossey-Bass, pp. 3-90, San Francisco, CA.

[19] Mackenzie, M.L. (2004) "A Theoretical Model of Managerial Decision Making and the Accumulation of Information", Proc. of the 33rd Annual Meeting of the Northeast Decision Science Inst., NY, USA, pp. 261-265.

[20] Lee, M.D., and Cummins, T.D.R. (2004) "Evidence Accumulation in Decision Making: Unifying the 'Take the Best' and the 'Rational' Models', Psychologic Bulletin & Review, 11(2), 2004, pp. 343-352.

[21] Balci, F., Simen, P., Niyogi, R., Saxe, A., Hughs, J., Holmes, P., and Cohen, J.D. (2011) "Acquisition of Decision Making Criteria: Reward Rate Ultimately Beats Accuracy", Attention, Perception & Psychophysics, 73(2), 2011, pp. 640-657.

[22] SWT https://www.w3.org/standards/semanticweb/

[23] OWL https://www.w3.org/TR/owl-features/

[24] SWRL https://www.w3.org/Submission/2004/03/

[25] Shojanoori, R., Juric, R., Lohi, M., Computationally Significant Semantics in Pervasive Healthcare, Journal of IDPT, IOS Press, Volume 16, No. 1, 2012, pp. 43-62.

[26] Shojanoori, R. (2013) "Towards formalization of situation-specific computations in pervasive computing environments", PhD Thesis, Univ. of Westminster, UK

[27] Macfie, A., W. (2014) "Semantic role-based access control", PhD Thesis, University of Westminster, 2014.

[28] Kataria, P. (2011) "Resolving semantic conflicts through ontological layering", PhD Thesis, Univ. of Westminster, UK.