Commonsense Knowledge for Human-Robot Collaboration in Hybrid Search and Rescue Teams*

Kurt Geihs, Stefan Jakob, Stephan Opfer, and Yasin Alhamwy

Abstract— **Collaborative autonomous rescue robots have become an increasingly important integral part of rescue teams in large-scale emergency missions. In order for such robots to act effectively, they need to collaborate with human rescuers of their own rescue team as well as with robots of other rescue teams. This raises a number of specific challenges: rescue teams from different organizations and their rescue robots have different skills, capabilities, and control facilities; robots cannot be pre-programmed completely at design time considering the unknown disaster environment; human rescuers have general commonsense knowledge that robots do not have per se. Our specific focus in this paper is on the latter subject. Our main contribution is a comprehensive methodology to automatically translate commonsense knowledge from general knowledge sources into an easily applicable and flexible representation, and thus to provide a bridge language for the human-robot interaction in heterogeneous hybrid rescue teams. Clearly, service robots in other application domains will also benefit from our solution.**

I. Introduction

Critical infrastructures for electricity, communications, water, traffic, etc. are endangered by large-scale natural disasters such as storms, floods, wildfires, and earthquakes. Unfortunately, it seems that the number of such incidents is increasing [1]. In case of such life-threatening disasters, typically many rescue teams from different organizations arrive at the disaster area and collaborate in order to provide quick response and mitigate the effects of the disaster. The use of autonomous robots for search and rescue missions has increased in recent years due to their ability to perform exploration and rescue tasks in areas that are too dangerous for human rescuers or in areas that are inaccessible by the human rescuers.

Rescue teams specialize in different activities and thus bring different kinds of robots with different properties and capabilities to the disaster area. This raises a wide range of technical challenges related to the organization, planning, and execution of the collaborative rescue activities, in particular when humans and autonomous robots are involved. That is what we call *hybrid teams*.

*Research supported by the emergenCITY project as part of the LOEWE program of the state Hessen in Germany.

K. Geihs is with the Scientific Center for Information System Design (ITeG) at University of Kassel, Kassel, Germany
(e-mail: geihs@uni-kassel.de).

S. Jakob was with the University of Kassel, Kassel, Germany. He is now with Micromata GmbH, Kassel, Germany (e-mail: s.jakob872@gmail.com)

S. Opfer was with the University of Kassel, Kassel, Germany. He is now with Micromata GmbH, Kassel, Germany (stephan.opfer@posteo.net)

Y. Alhamwy is a PhD candidate at the University of Kassel, Kassel, Germany (e-mail: alhamwy@uni-kassel.de)

An important aspect of such hybrid teamwork is the management of the joint knowledge which is needed for the purposeful interactions of humans and autonomous robots. This applies in particular to equipping the heterogeneous autonomous robots with commonsense knowledge that naturally underlies the reasoning and actions of every human rescuer but cannot be assumed to be available to the robots.

Thus, in this paper our main focus is on the following general research question: How can we improve the human-robot-interaction by extending a joint knowledge base with commonsense knowledge in order to provide a better basis for the collaboration of human rescuers and heterogeneous autonomous robots? Such formalized commonsense knowledge should also facilitate the collaborative teamwork of the heterogeneous multi-robot system.

The research reported here is performed under the roof of a large interdisciplinary research center called emergenCITY[1] that is operated by three German universities together with several application partners. The overall goal is to improve the resilience of the future smart city: How can the functionality of cities with digitally networked critical infra-structures be ensured in extreme crisis situations and disasters? To achieve this goal emergenCITY explores four research threads: *1) Cyber-Physical Systems* focusing on (semi-)autonomous robotic systems for emergency response and recovery support. *2) Communication* focusing on the design of resilient communication networks. 3*) Information* focusing on the provision of information and communication services targeted at reaction, recovery, and prevention of crisis situations. 4*) City and Society* focusing on the historical, political, social, and legal aspects relevant for the design of resilient technologies.

In the remainder of this paper we present our main contribution, i.e. a comprehensive method to provide commonsense knowledge for hybrid multi-robot teams in dynamic and volatile disaster scenarios. The paper is organized as follows: In Section II we discuss the goals of our work and the research-guiding requirements. Section III presents our solution particularly focusing on the aspect of commonsense knowledge. In Section IV we show results of evaluation experiments. Section V discusses related work. Section VI concludes the paper and points to future work.

II. Requirements and Problem Statement

Experiences with large scale disasters such as the recent floods in Germany have shown that the management of the relevant information for the more or less spontaneous coordination and allocation of official and voluntary rescue

[1] Emergency Responsive Digital Cities, https://www.emergencity.de/

activities as well as the distribution of aid supplies – among many other tasks – is key for an effective crisis response.

Generally, in such scenarios the challenges for effective help activities by heterogeneous teams consisting of human rescuers and autonomous rescue robots include:

(1) Inherent heterogeneity. Human rescue forces and their autonomous rescue robots are specialized for different tasks and thus have special characteristics, properties, and capabilities. Their collaboration as a hybrid team requires a common knowledge representation and management platform for joint reasoning, planning, and execution of actions.

(2) Commonsense knowledge. While we may assume that human rescue teams have a largely overlapping general commonsense knowledge of search and rescue-related services and beyond, such general knowledge cannot be assumed for the robots involved. Thus, we need a general method to convert accessible commonsense knowledge from knowledge sources such as ConceptNet [2] and Search & Rescue ontologies [3] to a common knowledge representation, as well as a transformation of the common representation into the local representations of the autonomous robots.

Besides the knowledge-related challenges, we must assume:

(3) Dynamic run-time environment. The crisis area may be changing dynamically, e.g., in terms of the state of the terrain and its objects.

(4) Harsh run-time environment. Communication between the involved agents, i.e. humans and robots, may be unreliable. Robots may move out of reach, break down temporarily or fail permanently. Thus, the number of communicating participants fluctuates, and a centralized knowledge management is not a good choice in order to prevent a single point of failure.

III. SOLUTION

A. Foundations

Before we present our solution, we give a brief overview of the key techniques whereupon our solution is based.

1) Knowledge representation

To efficiently work with knowledge, an appropriate representation is needed. In general, such a representation can be seen as a relation between two domains [4], where the first domain is the representation, i.e. a symbol, and the second an object in the domain. In general, knowledge representation focuses on symbols defining or representing propositions that are known or believed by agents.

We use Answer Set Programming (ASP) [5] which is a declarative and non-monotonic logic programming formalism tailored for NP-hard search problems. Furthermore, ASP combines the results of research in the fields of knowledge representation, logic programming, and constraint satisfaction [6]. The complete ASP-Core-2 Input Language Format is presented in detail in [7].

Key to the reasoning based on ASP is the solving process that generates solutions for an ASP program. There are several ASP solvers available. ASP solvers like Clingo [8] typically divide the solving process into two distinct steps. The first step is the grounding, which replaces variables with the corresponding part of the Herbrand Universe [9] of the ASP program. This results in a variable free ASP program, which is then given to the second step, i.e. solving the ASP program. It determines the solution of the ASP program denoted as an Answer Set or Stable Model. Generally speaking, the Answer Set consists of all facts as well as all derived rule heads.

For solving ASP programs, we selected the Clingo solver [8] mainly for its capabilities to handle dynamically changing knowledge specifications, which is related in particular to our requirements (1) and (3) above. Clingo's features called External Statements and Program Sections support the required dynamic extension of the knowledge base and multi-shot solving. See [8] for further details.

2) ConceptNet5

In order to demonstrate the viability of equipping robots with commonsense knowledge, we chose ConceptNet 5 (CN5) [2] as an example. CN5 is a multilingual commonsense knowledge base. It aims at supporting machines in understanding the meaning of concepts, which appear in the daily life of humans. The basis of CN5 is a semantic hypergraph that represents commonsense knowledge, which has been extracted from various knowledge sources like the Open Mind Common Sense project [10], Wiktionary21[2], WordNet [11], DBPedia [12], and Wikipedia[3]. In version 5.7, the CN5 hypergraph consists of approximately 34 million edges and supports 304 languages.

A concept in CN5 is a natural language term, which is annotated with a language tag and a sense label denoting its word class. In order to build a hypergraph, these concepts are connected by edges. The meaning of an edge is given by its assigned relation. CN5 defines 34 different relations. Besides their sources and a relation, edges contain a weight. This weight is the sum of weights assigned to the sources. A weight greater than or equal to 1.0 denotes that the edge has been extracted from at least one verified source like WordNet. The higher the weight, the more reliable is the knowledge represented by the edge.

3) Architecture

The Multi-Agent-System (MAS) paradigm is well suited to model the interactions between autonomous devices, i.e. intelligent autonomous agents, as well as between such agents and human actors [13]. For example, an agent can be an autonomous robot, a computer program, or a human being. According to the widely accepted MAPE-K agent architecture [14], an agent monitors its environment via sensors, analyses the acquired information, reasons about reactive action plans, and potentially executes interactions with the environment including, in case of a MAS, collaborative activities with other team members, i.e. robots and human operators, using its actuators and communication facilities. An important component in this MAPE-K cycle is a knowledge base where the agent stores static and dynamic knowledge about the environment.

[2] Wiktionary, The Free Dictionary, https://en.wiktionary.org
[3] Wikipedia, The Free Encyclopedia, https://en.wikipedia.org

Rescue robots may encounter situations where other agents are out of reach, or communication has broken down temporarily. Thus, agents need to decide autonomously about their activity based only on their own local knowledge base. Clearly, as stated above, a centralized knowledge management architecture for the hybrid agent team would not be appropriate because it would create a central point of failure as well as a potential performance bottleneck.

B. Providing Commonsense Knowledge for Hybrid Teams

In contrast to pure machine-to-machine communication, the interaction with humans benefits greatly from the incorporation of commonsense knowledge since humans make use of it while solving everyday tasks [15]. Thus, robots working with humans should have knowledge provided by commonsense knowledge sources, such as CN5. Obviously, incorporating and reasoning about such commonsense knowledge requires a suitable logic formalism. Based on the general approach described in [16], we provide a methodology tailored to the requirements of hybrid search and rescue teams.

A typical approach to model knowledge and provide a common vocabulary is the use of an ontology. However, current ontology frameworks do not support dynamic adaptation of the represented knowledge and non-monotonic reasoning, do not provide a formalism to express negation, and cannot handle huge amounts of data [17]. These issues demand a new way to model ontologies. Krötzsch suggests in [18] that a declarative and symbolic knowledge representation is needed to overcome these issues. Furthermore, we require not only a knowledge representation formalism, but also a computational reasoning paradigm. ASP is particularly suited for this. It provides non-monotonic reasoning capabilities and allows to model default knowledge by applying negation-as-failure and classical negation. It relies on symbolic knowledge representation and supports the application of the closed world assumption.

Besides the selection of a suitable representation, several challenges arise. The first challenge is the vast amount of commonsense knowledge itself. For example, CN5 has several million edges which are almost impossible to handle manually. Hence, an automated approach is needed. Storing a complete commonsense knowledge base could limit its usage to robots with high computational power and thus would limit its applicability. Therefore, the ontology generation has to be configurable in order to select only relevant parts for the given application scenario. Another challenge is the adaptability of the ontology itself. Since the necessary commonsense knowledge can vary depending on the environment of a robot, parts of the ontology have to be adapted or expanded during run-time.

By relying on the presented features of ASP, we have developed a novel general method for generating a dynamic and non-monotonic ontology. The resulting ontology is derived from a hypergraph-based knowledge source, such as CN5. It can be adapted during run-time and does not require a rebuild of the ontology if further knowledge is added. Thus, our focus is set on the creation of dynamic ontologies that enable classification of individuals, the derivation of super- and subclasses, as well as the incorporation of facets. Facets provide means for a detailed specification of properties in an ontology.

The first step of our methodology for the generation of a commonsense knowledge ontology is the extraction of edges from the hypergraph-based knowledge source. During this process, the initial classes of the ontology and their relations are formed. We start from a given root concept and traverse the hypergraph until a set of stopping criteria is met. For example, to create a simple taxonomy using CN5, the relation *IsA* and an edge weight of 2.0 as the stopping criterion could be used. Selecting such a high edge weight as a stopping criterion will only consider edges with at least two verified sources to become part of the generated ontology. Thus, a small ontology consisting of trusted and reliable edges is created. In the case of unweighted knowledge sources like WordNet [11], a maximum number of hops can be used as a stopping criterion. Therefore, the edges directly connected to the root concept are annotated with the maximum hop number. Subsequently, each layer of edges connected to the previous layer is annotated with a decremented hop count.

After successfully extracting relevant edges in the first processing step, the second step translates the resulting set of edges into ASP by applying a second algorithm. Its input is a set of edges E, and it returns a corresponding ASP program. First, the algorithm creates an empty ASP program. Subsequently, each edge in E is translated into an ASP fact. By annotating facts with the keyword *#external* denotes them as *External Statements* and, thus, their truth value can be dynamically and flexibly altered at run-time, which is a key requirement in our application domain. Further details can be found in [19] [20].

C. Speech-based HRI

The most natural ways for human-robot interaction is to talk to a robot. Therefore, a speech recognition software such as Sphinx[4] coupled with a part-of-speech tagging system such as spaCy[5] would allow humans to communicate with the robots quite naturally.

We consider the gap between the output of such a speech tagging algorithm and an ASP rule to be small compared to other knowledge representations. However, there is still a considerable amount of research to be done before we can automatically transform natural speech into ASP rules. We do not directly address this research area, but Schwitter et al. [21] [22] present proper solutions for the case of controlled natural language. Based on their work, we assume that it is possible to convert parts of natural language into ASP rules directly. However, since our initial research focus is primarily on the knowledge representation and reasoning support for hybrid rescue teams, we have not yet included a natural language interface. This is future work. In the current version of our system, we provide a graphical user interface that expects already well-formed ASP rules as input [19]. With this interface, it is possible to retrieve, create and change knowledge content in order to affect the robots' MAPE-K cycles.

[4] CMUSphinx, https://cmusphinx.github.io/wiki/
[5] Industrial-Strength Natural Language Processing, https://spacy.io/

IV. EVALUATION

The transformation of commonsense knowledge from CN5 to ASP expressions has been evaluated in a range of experiments. As explained in the previous section, the selected stopping criteria significantly influence the size of the generated commonsense ontologies and the runtime of the transformation. Thus, we first evaluate their impact on the size of the resulting ontologies and then the corresponding runtimes. More details can be found in [20].

A. Ontology Size

Two parameters primarily influence the automatic extraction of commonsense ontologies. The first parameter is the selected root concept, which determines the starting position of the extraction in the selected hypergraph. The second parameter is the set of stopping criteria for the extraction.

The following evaluation uses CN5 as the commonsense knowledge source. The considered root concepts are Animal, Car, Person, and Thing. Furthermore, minimum edge weights are used as stopping criteria since they express the reliability of the extracted edges. For the creation of the ontology, edges labelled with the relations IsA, FormOf, Synonym, and HasProperty are used. Since the highest number of edges is given for the IsA relation, the adaption of its minimum weight will have the highest impact on the size of the resulting ontology. Therefore, we use the minimum weights 2.5, 2.0, and 1.0 for the IsA relation. The other relations have a fixed value of 2.0 for the minimum weights. TABLE I. shows the resulting ontology sizes.

TABLE I. ONTOLOGY SIZES (NUMBER OF GENBERATED EDGES)

Min. Weight	Root Concept			
	Animal	*Car*	*Person*	*Thing*
2.5	522	34	17	196
2.0	95353	95353	95353	95353
1.0	201148	201148	201148	201148

The minimum edge weight of 2.5 is the most restrictive criterion, and the resulting ontologies have the lowest number of edges. Lowering the minimum weight increases the size of the ontology. In the case of a minimum weight of 2.0, the influence of the root concept is no longer present and a big part of the CN5 hypergraph is explored, resulting in ontologies of same size for all root concepts. Similarly, lowering the minimum weight to 1.0 roughly doubles the ontology size. Thus, as expected, higher minimum edge weights create smaller domain-specific ontologies, while lower minimum weights lead to bigger, more general commonsense ontologies.

B. Runtime

Let us have a look at the runtime of ontology generation.[6] Again, CN5 is used as the commonsense knowledge source, and edges annotated with the relations IsA, FormOf, Synonym, and HasProperty are considered. Furthermore, the minimum edge weight is used as the stopping criterion. The runtime of the ontology generation has been measured 20 times for each root concept and minimum weight. TABLE II. shows the mean values of the measured runtimes. The corresponding standard deviations are given in TABLE III. Considering the ontology sizes presented in TABLE I. we observe that the generation runtime scales linearly with the ontology size.

TABLE II. RUNTIME (IN SECONDS)

Min. Weight	Root Concept			
	Animal	*Car*	*Person*	*Thing*
2.5	142.05	5.01	3.65	50.75
2.0	8097.45	8139.04	8199.49	8142.40
1.0	13505.68	13402.36	13341.25	13134.26

TABLE III. STANDARD DEVIATION (IN SECONDS)

Min. Weight	Root Concept			
	Animal	*Car*	*Person*	*Thing*
2.5	0.61	0.22	0.20	0.43
2.0	51.54	20.95	66.44	34.64
1.0	103.87	83.11	54.40	84.25

Another essential aspect besides the ontology generation is reasoning. This process consists of seven steps. The first four steps ensure that the ASP solver Clingo can use the ontology during subsequent queries. This includes the adding and grounding of the edges of the ontology, which are modelled by External Statements. Clingo assumes newly added External Statements as false. They have to be set to True in the third step, thus including them in the reasoning process. After this process is finished, the ontology can be used to classify individuals via queries. Queries are separate ASP programs. They require three steps, i.e. adding, grounding, and solving. The measured runtimes show similar behavior to the runtimes of the ontology generation. The highest impact on the total reasoning runtime is caused by the introduction of the ontology to Clingo, i.e. adding and grounding of the ontology have the biggest impact on the runtime, while the other steps take much less time.

V. RELATED WORK

The handling of commonsense knowledge has been studied in various application domains. Here, we provide a brief selective overview and discuss differences to our work.

A well-known framework using knowledge to enhance the capabilities of service robots is KnowRob [23]. The primary objective of this framework is the improvement and execution of plans by applying domain-specific as well as commonsense knowledge. A central aspect of KnowRob is its knowledge base. The available information is stored in Prolog predicates. In addition to the classical knowledge base, a virtual knowledge base is created. It is an extension that creates abstract representations of data once it is queried to improve its overall quality. The knowledge base is based on OWL ontologies. These ontologies adhere to the Open-World assumption and are undecidable in the full OWL

[6] Experiments performed on a Lenovo T570 workstation with Intel® Core™ i7-7500U @ 2.70 GHz Dual-Core, 16GB RAM, Ubuntu 18.04.4.

specification. Since using the decidable subset of OWL is limited in its expressiveness [16], in cases where the Closed-World assumption is needed, Prolog and a corresponding reasoner are used. While the OWL ontologies form the first layer of the knowledge base, Prolog predicates in the second layer link the OWL classes to corresponding reasoning methods. A complete specification of the KnowRob knowledge base and its knowledge representation is given in [23].

In its second version [24], besides Prolog predicates, KnowRob introduces the concept of *computables*. Their truth value is initially unknown and later set by external sources. Hence, they are comparable to the External Statements provided by Clingo. In this respect, KnowRob and our framework are based on a similar kind of symbolic knowledge representation. A major difference is the number of involved agents. KnowRob is designed to be used on a single intelligent agent. In contrast, our multi-agent knowledge base distributes its knowledge management on a dynamically changing team of hybrid agents. The second major difference is given in the dynamics of the application scenarios. Robots equipped with KnowRob achieve astonishing results for application scenarios that are rather static and do not require fast reasoning, for example, a scenario for preparing breakfast, making it unsuitable for our target environments. In contrast to this, our knowledge base is tailored specifically with the dynamically changing environments in mind.

In [25], Erdem et al. employ a hybrid planning approach to model the task of tidying a household. It is a combination of ASP, Prolog, and a motion planner. The task of tidying the household as well as the actions of a robot are formulated in ASP. ConceptNet 4, the predecessor of ConceptNet 5, is used as a commonsense knowledge source. Two of its relations are extracted and translated into ASP predicates for further use: The AtLocation relation provides typical places of objects, while the HasProperty relation is used to describe if an object is fragile. The resulting ASP predicates are then used as external input to the Prolog program, which performs the actual task planning. Finally, the resulting task enriched with commonsense knowledge is given to the motion planning.

The approach of Erdem et al. is similar to our integration of commonsense knowledge into the knowledge base of autonomous agents. Besides using ConceptNet 5 instead of ConceptNet 4, we translate the extracted knowledge directly into ASP instead of using Prolog. This reduces the overall complexity of the system since no additional formalism has to be applied. Additionally, the use of Clingo's External Statements enables the retraction of commonsense knowledge if contrary knowledge is received. A further difference is the amount of employed commonsense knowledge. Erdem et al. only use the two relations AtLocation and HasProperty in order to prevent possible inconsistencies. In contrast, we support the full set of base relations in CN5. To prevent semantic inconsistencies in the knowledge base of a robot, we introduce an automatic inconsistency prevention in [26], which relies on commonsense knowledge to find contradictions in the properties of an object.

Lemaignan et al. present in [27] a framework for autonomous robots with advanced human-robot interaction skills. They use OWL in combination with the Pellet reasoner [28] as their knowledge representation and reasoning formalism. This limits the addition of further knowledge during runtime, since the complete knowledge base has to be classified again if new classes are to be added dynamically. The knowledge base of a robot is supplemented with databases such as WordNet [11] and DBPedia [12], that are part of CN5. The communication and dialogues between robots are modelled by the *Dialogs* component [27]. It heavily relies on the knowledge base to resolve the relationships between the perceived words.

The usage of commonsense knowledge shown by Lemaignan et al. is similar to our usage of commonsense knowledge and our detection of inconsistencies introduced in [26]. Both enrich their knowledge bases with commonsense knowledge extracted from external sources. A difference is given in the used reasoning formalism. Lemaignan et al. rely on OWL, which requires the reclassification of the complete knowledge base to introduce additional classes at runtime. We avoid this disadvantage by using ASP and Clingo.

Ayari et al. present in [29] a system for ambient assisted living, which applies commonsense knowledge and reasoning. Two ontologies form the basis of their framework. The first ontology provides a hierarchy of classes and individuals using a generalization / specification relation like *IsA*. It is used to describe static commonsense knowledge. The second ontology is used to represent dynamic terms. Both ontologies can be enriched by commonsense knowledge extracted from external sources like WordNet [11]. Instead of using OWL for the ontologies, the Narrative Knowledge Representation Language (NKRL) [30] is employed to form the ontologies. In addition to classical ontology features, NKRL supports the definition of events, which serve as templates for actions.

The framework of Ayari et al. and our approach both use commonsense knowledge to support agents in solving their tasks, for example, in search and rescue missions or service robots in household scenarios. The major difference is the knowledge representation language. Ayari et al. use NKRL, which focuses on the narratives of text. It is not fully declarative and adheres to the Open-World assumption. In contrast, our approach uses the fully declarative language ASP, which supports the dynamic adaption of the knowledge and enables the usage of the Closed-World assumption.

VI. Conclusion and Outlook

Human-robot interaction is a crucial element of search and rescue missions in large-scale emergency situations. In contrast to other application domains, autonomous rescue robots cannot be equipped completely in advance with all the required domain knowledge of their a priori unknown runtime environments. Our general solution approach, i.e. the symbolic representation of knowledge using ASP as a bridge language, facilitates the interaction between humans and rescue robots, since the knowledge presented is much more accessible to humans than any connectionist approach such as a neural network.

In particular, our focus is on the mitigation of the burden of missing general commonsense knowledge by integrating such knowledge into the joint knowledge base of hybrid teams in an automatic fashion. Thus, the autonomous robots can even understand implicit human requests and offer intelligent assistance. As a first step, our interface proposed for humans to interact with rescue robots supports only valid ASP rules. In the future, this interface should also understand (restricted) natural language which can be achieved based on existing work in the field [21] [22].

The integration of CN5 as commonsense knowledge database improves the interaction between robots and humans substantially. However, CN5 is only one example. There exists a significant amount of ontologies represented in description-logic-based OWL. The automatic transformation of these OWL ontologies into ASP would offer more sources of commonsense knowledge for rescue robots and would also allow the application of the performance-wise superior ASP reasoning techniques to these ontologies.

Some of the knowledge encoded in CN5 is hard to translate automatically to ASP. For example, the concept *Your Own Kitchen* has the capability to *Store Cups*. Nevertheless, this does not induce the existence of the *atLocation* relation between cup and kitchen. Unifying the concepts represented in CN5, for example, by trimming unimportant parts like *Your Own* and ignoring the difference between singular and plural (e.g., *victim* vs. *victims*) would allow the robots to benefit from even more knowledge encoded in CN5.

Clearly, service robots in other dynamic application domains will also benefit greatly from our solution approach.

ACKNOWLEDGMENT

The authors thank their colleagues at the Distributed System department of the University of Kassel for insightful discussions and research contributions.

REFERENCES

[1] UN Office for Disaster Risk Reduction (UNDRR). "2020: The Non-COVID Year in Disasters". https://www.undrr.org/publication/2020-non-covid-year-disasters (April 2021).

[2] Robyn Speer, Joshua Chin and Catherine Havasi. "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge". Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) (2017), pp. 4444–4451.

[3] Stasinos Konstantopoulos, Georgios Paliouras and Symeon Chatzinotas. "SHARE-ODS: An Ontology Data Service for Search and Rescue Operations". Advances in Artificial Intelligence, Springer (2006), pp. 525–528.

[4] Ronald J. Brachman and Hector J. Levesque. "Knowledge Representation and Reasoning". Morgan Kaufmann Series in Artificial Intelligence: Morgan Kaufmann (2003).

[5] Michael Gelfond and Yulia Kahl. "Knowledge Representation, Reasoning, and the Design of Intelligent Agents, The Answer-Set Programming Approach". Cambridge, USA: Cambridge University Press (2014).

[6] Gerhard Brewka, Thomas Eiter and Mirosław Truszczynski. "Answer Set Programming at a Glance". Communications of the ACM 54.12 (2011), pp. 92–103.

[7] Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Marco Maratea, Francesco Ricca and Torsten Schaub. "ASP-Core-2 Input Language Format". Theory and Practice of Logic Programming 20.2 (2020), pp. 294–309.

[8] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub and Sven Thiele. "Engineering an Incremental ASP Solver". International Conference on Logic Programming (2008), pp. 190–205.

[9] H.-D. Ebbinghaus, Jörg Flum and Wolfgang Thomas. "Mathematical Logic", Springer Science & Business Media (2013).

[10] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins and Wan Li Zhu. "Open Mind Common Sense: Knowledge Acquisition from the General Public". OTM Confederated International Conferences. Springer (2002), pp. 1223–1237.

[11] George A. Miller. "WordNet: A Lexical Database for English". Communications of the ACM 38.11 (1995), pp. 39–41.

[12] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak and Zachary Ives. "DBPedia: A Nucleus for a Web of Open Data". The Semantic Web, Springer (2007), pp. 722–735.

[13] Stuart Russell and Peter Norvig. "Artificial Intelligence: A Modern Approach" (4th Ed.), Prentice Hall (2020).

[14] Jeffrey O. Kephart and David M. Chess. "The Vision of Autonomic Computing". Computer 36.1 (2003), pp. 41–50.

[15] Ernest Davis. "Representations of Commonsense Knowledge", Morgan Kaufmann (2014).

[16] Stefan Jakob, Alexander Jahl, Harun Baraki and Kurt Geihs. "Generating Commonsense Ontologies with Answer Set Programming". Proceedings of the ICAART 2021. Vol. 2. ICAART2021. SCITEPRESS (2021), pp. 538–545.

[17] Boris Motik, Ian Horrocks, Riccardo Rosati and Ulrike Sattler. "Can OWL and Logic Programming Live Together Happily Ever After?" The Semantic Web - ISWC 2006. Ed. by Isabel Cruz et al., Vol. 4273. Lecture Notes in Computer Science (2006), pp. 501–514.

[18] Markus Krötzsch. "Ontologies for Knowledge Graphs?" 30th Int. Workshop on Description Logics. Vol. 1879. CEUR-WS.org (2017).

[19] Stephan Opfer. "Symbolic Representation of Dynamic Knowledge for Robotic Teams". PhD thesis. Kassel, University of Kassel, Faculty of Electrical Engineering and Computer Science (Nov. 2020).

[20] Stefan Jakob, "A Self-Organising Multi-Agent Knowledge Base". PhD thesis. Kassel, University of Kassel, Faculty of Electrical Engineering and Computer Science (Dec. 2021).

[21] Stephan C. Guy and Rolf Schwitter. "The PENG ASP System: Architecture, Language and Authoring Tool". Language Resources & Evaluation 51.1 (2017), pp. 67–92.

[22] Rolf Schwitter. "Specifying and Verbalising Answer Set Programs in Controlled Natural Language". Theory and Practise of Logic Programming 3-4 (2018), pp. 691–705.

[23] Moritz Tenorth and Michael Beetz. "Representations for Robot Knowledge in the KnowRob Framework". Artificial Intelligence 247 (2015), pp. 151–169.

[24] Michael Beetz, Daniel Beßler, Andrei Haidu, Mihai Pomarlan, Asil Kaan Bozcuoglu and Georg Bartels. "Know Rob 2.0 - A 2nd Generation Knowledge Processing Framework for Cognition-Enabled Robotic Agents". 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2018), pp. 512–519.

[25] Esra Erdem, Erdi Aker and Volkan Patoglu. "Answer Set Programming for Collaborative Housekeeping Robotics: Representation, Reasoning, and Execution". Intelligent Service Robotics 5.4 (2012), pp. 275–291.

[26] Stefan Jakob, Stephan Opfer, Alexander Jahl, Harun Baraki and Kurt Geihs. "Handling Semantic Inconsistencies in Commonsense Knowledge for Autonomous Service Robots". 2020 IEEE 14th International Conference on Semantic Computing (ICSC) (2020), pp. 136–140

[27] Séverin Lemaignan, Mathieu Warnier, E. Akin Sisbot, Aurélie Clodic and Rachid Alami. "Artificial Cognition for Social Human–robot Interaction: An Implementation". Artificial Intelligence 247 (2017), pp. 45–69.

[28] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz. "Pellet: A Practical OWL-DL Reasoner". Journal of Web Semantics 5.2 (2007), pp. 51–53.

[29] Naouel Ayari, Abdelghani Chibani, Yacine Amirat and Eric T. Matson. "A Novel Approach Based on Commonsense Knowledge Representation and Reasoning in Open World for Intelligent Ambient Assisted Living Services". IROS 2015 - IEEE/RSJ International Conference on Intelligent Robots and Systems (2015), pp. 6007–6013

[30] Gian Piero Zarri. "NKRL, A Kowledge Representation Tool for Encoding the Meaning of Complex Narrative Texts". Natural Language Engineering 3.2 (1997), pp. 231–253.