Future and Emerging Technologies
FET-Open

Phoenix

665347

Open template ontologies for knowledge representation

Deliverable D3.2

V0.2

Document history – List of changes

Version	Date	Author name	Scope
v0.2	30/09/2016	Anil Yaman, Matt Coler, Giovanni Iacca	Final version
v0.1	27/09/2016	Anil Yaman, Matt Coler, Giovanni Iacca	First complete draft

Contents

Funded by the European Union

1 Introduction

This document includes the following deliverable:

- D3.2: Open template for ontologies for knowledge representation – An extendable framework for knowledge representation;

This deliverable is necessary for WP3, and WP4 insofar as they are critical to:

- identify the requirements and goals of an exploration task obtained from a user in the human interface layer (WP3);
- suggest the settings of the real world experiments in the reincarnation cycle (WP3);
- expand the knowledge in the Knowledge Base by incorporating the outcome of the results gained in the co-evolution and reincarnation cycles (WP3);
- present the results back to the user in the human interface layer (WP3);
- establish the evolutionary settings for adapting the sensor agents (motes) to the environments, depending on the exploration goals in co-evolutionary cycle (WP4).

This deliverable provides the components necessary for the implementation of the Phoenix Knowledge Base. The Knowledge Base is one of two core contributions of WP3:

- **The Phoenix Knowledge Base (from D3.1, D3.2, and D3.3)** is a component of Phoenix where formal knowledge is stored. Different types of knowledge are represented depending on the requirement. Developing the Phoenix Knowledge Base requires knowledge elicitation (D3.1) and knowledge representation. These processes are usually implemented by the *knowledge engineers*. Knowledge engineers are individuals involved in the processes of building and maintaining knowledge bases. In Phoenix, they perform knowledge elicitation and representation tasks.
- **The human interface layer (HIL) (from D3.4 and D3.5)** is a medium for the knowledge exchange between the user and the Phoenix system. This will be described in more detail with the delivery of the two associated deliverables in M24.

Figure 1 shows the components that WP3 develops (see for 4.2 an elaborate view of the components shown as a class diagram). The definitions of these components and their interactions can be found in (Yaman, Coler, & Iacca, 2015, 2016), and the specification of the work packages can be found in deliverable 2.2 (Phoenix, 2016).

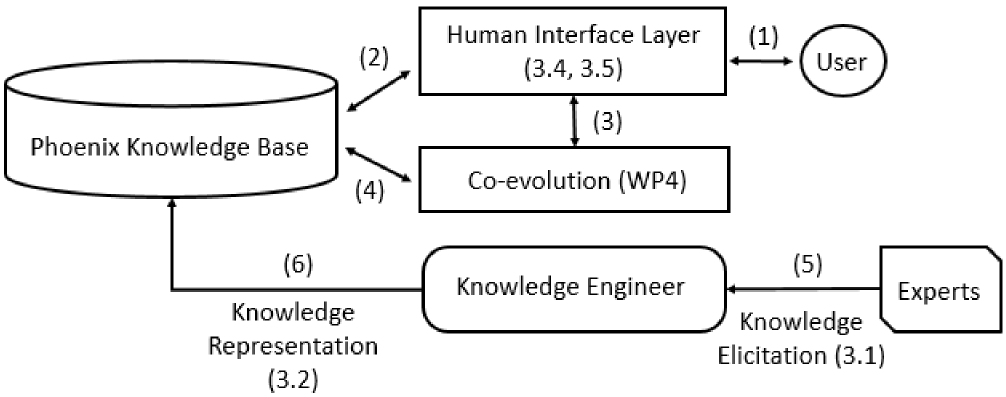

Figure 1: Interaction of the human interface layer and the Phoenix Knowledge Base with the other components of the project.

The definitions of the special terms we use in this document are provided in glossary 4.2.

2 D3.2: Open template for ontologies for knowledge representation

This deliverable uses the outcome of task 3.2 (that is, the knowledge targeted in knowledge elicitation tasks) to provide an open, extendable template for ontologies for knowledge representation (KR) in Phoenix, which is necessary for the Knowledge Base. This deliverable explicitly references an ontological KR which aims to support an extendible framework. Accordingly, we provide an uncertain graph representation which adheres to the ontology development guidelines (Noy & McGuinness, 2001), (Roussey, Pinet, Kang, & Corcho, 2011) for building an extendible framework for Phoenix Knowledge Base. Furthermore, uncertain graph representation is key to Phoenix especially with regards to the capacity of representing uncertain knowledge.

This deliverable is structured as follows: In section 2.1, we provide a summary of the state-of-the-art in KR techniques (with particular attention to KR of for uncertain knowledge in 2.1), which provides the "intellectual scaffolding" for our contribution. In section 2.2, where we present our KR technology. Finally, in section 2.3 we outline future work.

2.1 State-of-the-art survey in knowledge representation

Knowledge Representation (KR) is one of the subfields of artificial intelligence (AI) dedicated to structuring and representing knowledge about the world in a machine-processible format to solve complex problems. We follow the criteria posited by Davis (Davis, Shrobe, & Szolovits, 1993) regarding adequate KR, which makes explicit that any KR:

- is a substitution for objects in the real world;
- is a set of ontological commitments that focus on certain aspects of a phenomena and ignore/obscure others;
- is a means for drawing inferences; thus, it can be crucial to intelligent reasoning;
- is a basis for efficient computation where reasoning processes are performed;
- is a medium of human expression.

Broadly speaking, there are six approaches to KR, as outlined in the following table:

Table 1: Main KR technologies and their origins.

Main traditions in KR	Examples KR technologies	Suitable for representing	Origin
Knowledge-based systems	Production systems (Klahr, David and Langley, 1987), Frames & Scripts (Minsky, 1974)(Schank & Abelson, 1975), Graph models (Semantic Networks (Sowa, 1991), Ontologies (Gruber, 1995)(Roussey et al., 2011), Hierarchical Task Networks(Erol, Hendler, & Nau, 1994)),	Domain and Procedural Knowledge	Psychology
Logic	First order and Predicate (Hurley, 2014)	Procedural and Domain knowledge	Mathematics
Logic	Fuzzy (Zadeh, 1988), Probabilistic (De Raedt & Kimmig, 2015)	Procedural, Domain and Uncertain Knowledge	Mathematics
Connectionism (McClelland, Rumelhart, & McClelland, 1986)	Artificial Neural Networks	Procedural Knowledge	Biology
Casual Networks	Bayesian Networks (Jensen, 1996)	Procedural and Uncertain Knowledge	Statistics
Rational Agents	Belief-desire-intension (Georgeff, Pell, Pollack, Tambe, & Wooldridge, 1998)	Domain and Procedural Knowledge	Economics

The optimal KR strategy used depends on the task for which the representation is required. Since a core component of Phoenix has to do with the roles of human knowledge, and involves a high level of reasoning, we elected to rely on strategies from knowledge-based systems. Below we briefly summarize each of the KR technologies for knowledge-based systems in the table above.

Production systems (Klahr, David and Langley, 1987)**:** A production rule consists of two components, condition and action: "IF <condition> THEN <action>". Production systems use an inference engine to execute the rules. The inference engine reads the current state of the world from the memory, and retrieves the rule that matches to the memory. Execution of a rule changes the state of the world in the memory.

Frames & Scripts: Frames and scripts (Minsky, 1974), (Schank & Abelson, 1975) provide convenient representations of objects or events that are typical to a stereotypical situation. In frames, objects are represented by table-like structures called frames. Frames have slots that describe the aspects (properties) of objects. One slot of a frame can be related to another frame; thus may have relations. The main idea is that when an object is encountered, the “frame” that describes that object is retrieved from the memory. New frames can be defined when a new object is encountered, and existing frames can be modified if the belief of an object changes. Scripts are ideal for representing stereotyped sequence of events; thus, they best suited for representing procedural knowledge. Each script defines a script, track, props, roles, entry, result, and scenes.

Graph models (semantic networks, ontologies): Graph models are one of the fundamental data structures in computer science. Fundamentally, graphs are mathematical models studied in a field of graph theory (Dickson, 2006). Graph representations are ideal for representing concepts (as vertices) and their relations (as edges) in a domain. One of the most useful aspects of graph representation is the notion of inheritance where properties of concepts can be transferred to others without the need of defining all for each. This forms the basis for reasoning over graphs. However, in order to have inheritance, the concepts should be organized as hierarchies. Semantic networks (Sowa, 1991), ontologies (Gruber, 1995; Roussey et al., 2011), and hierarchical task networks (Erol et al., 1994) are the main knowledge representation technologies that use a graph structure. Semantic networks can be used for representing domain and procedural knowledge; but, are mainly suitable for domain knowledge. They can be specific to the type of relation they use. For example, *is-a* relation indicates that a child concept is a sub-set/member of a parent concept. *Is-partof* relation indicates that a child concept is a component of a parent concept. Mixed semantic networks allow many different relations between the concepts. Ontologies use similar representation with semantic networks. They usually consist of concepts, properties, relations, rules and logical axioms. Ontologies refer to a special set of technologies in the semantic web community. They use standardized frameworks and languages to facilitate easy integration and information exchange between different applications in the web (Krötzsch, Vrandečić, & Völkel, 2006).

2.1.1 Uncertainty in knowledge representation

Uncertain knowledge representation and reasoning is very active topic in the AI community (Stuart & Peter, 2003). This is, in part, because uncertainties are common in KR (Aggarwal & Yu, 2009) due to expert elicitation and formalization (O’Hagan, 2005), (Mitchell et al., 2015), or integration (Dong, Halevy, & Yu, 2009). Reasoning processes involve uncertainties that arise from unknown parameters, default values with range or probabilities, and noisy measurements (Aggarwal & Yu, 2009). Reasoning over a graph is often performed by querying algorithms, such as: reachability, shortest path, and pattern matching (Khan & Chen, 2015). Uncertain knowledge graphs and probabilistic query matching (Huang & Liu, 2009), probabilistic logic programming (De Raedt & Kimmig, 2015), and Markov logic networks (Richardson & Domingos, 2006) are some of the main approaches to knowledge representation and reasoning under uncertainty. Uncertain graph representation has been applied to many problems in fields that including knowledge bases (Dong et al., 2014; Udrea, Subrahmanian, & Majkić, 2006), sensor and communication networks (Colbourn, 1987; Ghosh, Ngo, Yoon, & Qiao, 2007), traffic networks (Hua & Pei, 2010)(Hua & Pei, 2010; J. Wang, Zhu, & Yang, 2013) and social networks (Deshpande & Getoor, 2009; Kempe, Kleinberg, & Tardos, 2003).

2.2 Phoenix open template for KR

The Knowledge Base needs an approach to KR to satisfy the requirements of applications. The Phoenix KR strategy must be extendable because, (1) the problem solving method of Phoenix should easily be applicable to different environments that are unknown, and (2) it should make it easy to update the system with any technological developments, including the integration of new/existing W3C compliant ontologies. Therefore, we adopted ontology development guidelines that can support modular and extendible KR design (Noy & McGuinness, 2001; Roussey et al., 2011).

During the knowledge elicitation process (D3.1), knowledge is structured based mainly on a hierarchical taxonomy of the concept meanings. This is a powerful design principle to allow reuse of existing knowledge across applications. Example superclass/subclass hierarchy on environment types is shown in D3.1. Similarly, we have different domains, such as: environments, motes, co-evolution, and reincarnation that are defined separately from each other. The relations between these separate domains are then established. These main domains and their relations form the skeleton of the Phoenix approach. Any extension is possible within these separate domains, even within the same domain. Our KR technique based on uncertain graph reflects this design principle and facilitates future development.

Uncertain graph representation also makes it possible for extensions of already existing knowledge represented as ontologies. For example, there are open ontologies that define the general framework for representing environments (“SWEET Ontology”, 2016) and sensor networks (Compton, Barnaghi, & Bermudez, 2011). These ontologies can be integrated into the representation. The capability of extension and reuse of knowledge is not limited to the domain knowledge. Procedural, inference and empirical knowledge types also support for extendibility in our framework.

The open template for KR ontologies we propose is an uncertain graph. Uncertain graphs incorporate features of the graph based knowledge representation technologies mentioned in 2.1. They are basis for ontological representations.

They can model uncertainty on an existence, or-set or attribute levels (Y. Wang, Li, Li, & Wang, 2013). Existence uncertainty indicates whether graph components exist or not with a certain probability value, or-set uncertainty indicates the probabilities of several alternatives that a graph component can take, and attribute uncertainty models continues random variables by their probability density functions. It is straight-forward to relate each of these aspects of uncertainty to Phoenix.

There are various formalisms of uncertain graphs. The optimal choice depends entirely on the application. Our current research focuses on the uncertain graph models that eventually incorporate all three aspects of uncertainty listed in the preceding paragraph. For now, we define an uncertain graph model with an existence level uncertainty. In **Definition 1,** an uncertain graph model with uncertain vertices and uncertain directed edges is provided. The vertices represent the concepts, and directed edges represent their relations. The labeling function L provides a label for each vertex and edge. Labels of vertices refer to concept names and labels of edges refer to the semantic of relation. This uncertain graph model is capable of representing the knowledge elicitation result provided in D3.1 but adding an existence probability to each concept and edge. If an existence of an element is known to be certain, then we can simply assign the probability of 1.

Definition 1 (Uncertain Graph): Let *G = (V, E, P, Σ, L)* be an uncertain graph where *V* is set of vertices, *E* is set of directed edges, *Σ* is set of labels, *P: V∪E → (0,1]* is a function that assigns an existence probability to each vertex and edge, and *L: V∪E → Σ* is a function that assigns labels to each vertex and edge.

Consider a variation of the example case we have developed in D3.1 where there is a 0.8 probability that the **T-Junction** environment has a junction location, and **Acceleration** and **Localization** can map **FlowVelocityProfile** with a probability value of 0.9 due to the measurements errors or some other factors (the rest of the knowledge assumed to be certain).

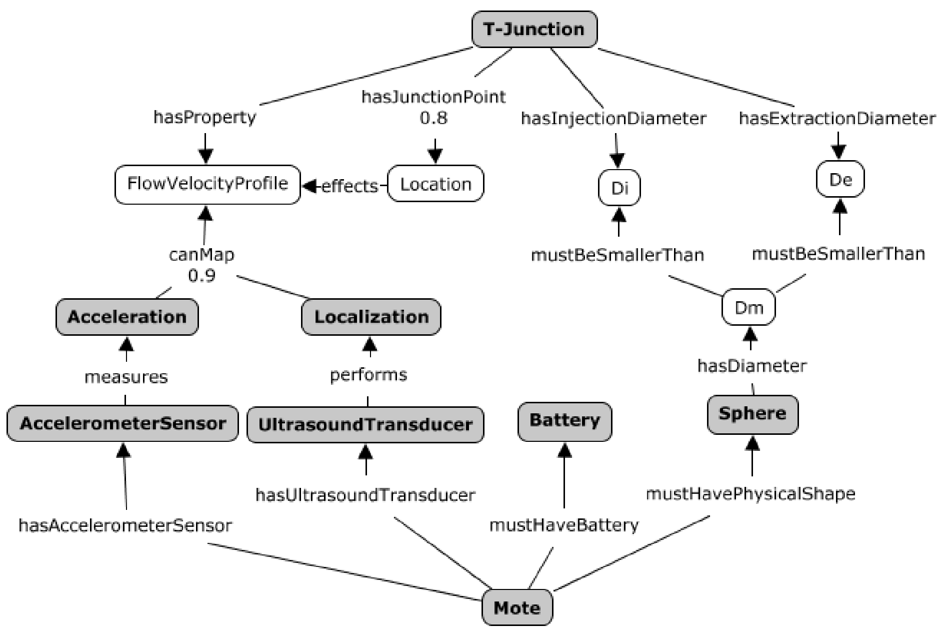

Figure 2: Uncertain graph representation of the example developed in D3.1.

Figure 2 represents the scenario where the uncertain knowledge is involved. Since we work on the existence level uncertainty, the edges labelled **hasJunctionPoint** and **canMap** can exist with probabilities 0.8 and 0.9. We assume possible world semantics without dependence (Abiteboul, Kanellakis, & Grahne, 1991), which interprets the uncertain graph *G* as a $2^{|V \cup E|}$ number of deterministic graphs called possible world graphs *g = ((V_g, E_g), Pr)*. Thus, the example graph in **Figure 2** can instantiate 4 possible world graphs. The probability of each of these possible world graphs can be calculated using Equation 1.

$$\Pr(g) = \prod_{c \in C_g} P(c) \prod_{c \notin C_g} (1 - P(c))$$

Equation 1: The probability of a possible world graph

wher$C_g \subseteq E_G \cup V_G$ and $c \subseteq C_g$ denotes union of uncertain vertices and edges that exist in possible world graph *g*. Table 2 shows the probabilities of all possible world graphs, that can be instantiated from

Figure 2. The notation "state(canMap)" can be equal to 0 or 1 that indicate the existence of the edge labelled as **canMap**. The notation "Pr(state(hasJunctionPoint) = 0| state(canMap) = 1)" defines the probability of the possible world where the edge hasJunctionPoint does not exist and canMap exists. In respect to knowledge, this possible world refers to the scenario where there is a junction in the environment, however, we cannot map the **FlowVelocityProfile** using **Acceleration** and **Localization**. This possible world can occur with a probability of 0.18. Perhaps in these kinds of situations, we have to find other ways to map **FlowVelocityProfile** and show the location of the junction.

Table 2: Probabilities of all possible world graphs.

1	Pr(state(hasJunctionPoint) = 1| state(canMap) = 1) = 0.72
2	Pr(state(hasJunctionPoint) = 0| state(canMap) = 0) = 0.02
3	Pr(state(hasJunctionPoint) = 1| state(canMap) = 0) = 0.08
4	Pr(state(hasJunctionPoint) = 0| state(canMap) = 1) = 0.18

The process of accessing the knowledge in the Knowledge Base is carried out by query evaluation. Query evaluation process is used in Human Interface Layer for interpreting the user's question, finding suitable experiments, co-evolutionary cycle for providing required parameters to evolve. In respect to these uses, query evaluation functions as a reasoning mechanism where questions can be asked to the Knowledge Base to determine their truth (or, in this case, the probability of being true). Consider the example where we want to figure out if we can use **Accelerometer** and **Localization** for mapping **FlowVelocityProfile.** In this case, we can try to find if the graph pattern given in Figure 3 matches to the graph given in Figure 2. The query evaluation process returns a probability value of 0.9. Complex query results can be evaluated by breaking them down to a simpler queries and combining the results. Usually, reachability (Zhu, Zhang, Zhu, Zhang, & Lin, 2011), shortest path (Cheng, Yuan, Wang, Qiao, & Wang, 2014), and pattern matching (Yuan, Wang, Chen, & Ning, 2016) queries are some of the most fundamental types of queries.

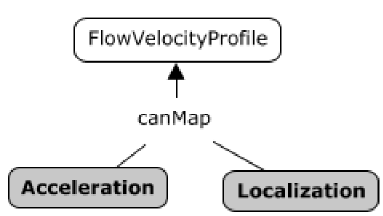

Figure 3: Pattern graph representation of a query.

Finding exact query evaluation requires exponential time complexity for uncertainty graphs. Therefore, we cannot afford to use the naïve enumeration method. Instead, we use efficient query evaluation strategies that include Monte-Carlo algorithms (Zhu et al., 2011), indexing (Khan, Bonchi, Gionis, & Gullo,

2014) and search space reduction (Zhu et al., 2011). These methods aim to provide fast and reliable query evaluation results.

2.3 Future work

We plan to extend the uncertain graph representation to include the results we obtain from the knowledge elicitation process. This knowledge will eventually involve or-set, and attribute level uncertainty representation. Therefore, we will include these levels of uncertainty into our uncertain graph model.

Our representation model assumes independent probabilities. We aim to include the cases where there are dependencies between the components of the Knowledge Base in our model. This is modeled under conditional probability models.

We will continue to work on the knowledge elicitation and representation processes for procedural knowledge. Reasoning by query evaluation is a suitable strategy for domain knowledge; however, it may not be the ideal way for representing and reasoning about procedural knowledge. Therefore, we identify the possibility of making use of alternative KR technologies, such as frames and scripts, that are suitable for procedural knowledge. Additionally, we will invest in the research where we investigate query processing for procedural knowledge. Furthermore, we will consider using an AI planner, and reasoning (case-based and analogical) methods to generate automated exploration procedures for answering the user's question.

3 References

Abiteboul, S., Kanellakis, P., & Grahne, G. (1991). On the representation and querying of sets of possible worlds. *Theoretical Computer Science*, *78*(1), 159–187. http://doi.org/10.1016/0304-3975(51)90007-2

Aggarwal, C. C., & Yu, P. S. (2009). A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, *21*(5), 609–623. http://doi.org/10.1109/TKDE.2008.190

Cheng, Y., Yuan, Y., Wang, G., Qiao, B., & Wang, Z. (2014). Efficient Sampling Methods for Shortest Path Query over Uncertain Graphs, 124–140. article.

Colbourn, C. J. (1987). The Combinatorics of Network Reliability. *OXFORD UNIVERSITY PRESS, 200 MADISON AVENUE, NEW YORK, NY 10016 (USA) , 1987, 176 Pp*. Retrieved from http://search.proquest.com/docview/24726015?accountid=14116\nhttp://ensor.lib.strath.ac.uk/sfxlcl41?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&genre=article&sid=ProQ:ProQ%3Acomputerinfo&atitle=The+Combinatorics+of+Network+Reliability.&ti

Compton, M., Barnaghi, P., & Bermudez, L. (2011). The SSN Ontology of the Semantic Sensor Networks Incubator Group. *Journal of Web …*, 1–6. Retrieved from http://www.w3.org/2005/Incubator/ssn/wiki/images/f/f3/SSN-XG_SensorOntology.pdf

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a knowledge representation? *AI Magazine*, *14*(1), 17–33.

De Raedt, L., & Kimmig, A. (2015). Probabilistic (logic) programming concepts. *Machine Learning*, *100*(1), 5–47. http://doi.org/10.1007/s10994-015-5494-z

Deshpande, A., & Getoor, L. (2009). Graphical models for uncertain data. *Managing and Mining Uncertain Data*, 1–36. http://doi.org/10.1007/978-0-387-09690-2_4

Dickson, A. (2006). Introduction to Graph Theory. *October*, (October), 1–15.

Dong, X. L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., … Zhang, W. (2014). Knowledge Vault : A Web-Scale Approach to Probabilistic Knowledge Fusion, 601–610.

Dong, X. L., Halevy, A., & Yu, C. (2009). Data integration with uncertainty. *VLDB Journal*, *18*(2), 469–500. http://doi.org/10.1007/s00778-008-0119-9

Erol, K., Hendler, J., & Nau, D. (1994). UMCP: A Sound and Complete Procedure for Hierarchical Task-network Planning. *Aips*, *94*, 249–254. Retrieved from http://www.cs.umd.edu/~nau/papers/erol1994umcp.pdf

Georgeff, M., Pell, B., Pollack, M., Tambe, M., & Wooldridge, M. (1998). The belief-desire-intention model of agency. In *International Workshop on Agent Theories, Architectures, and Languages* (pp. 1--10). Springer.

Ghosh, J., Ngo, H. Q., Yoon, S., & Qiao, C. (2007). On a routing problem within probabilistic graphs and its application to intermittently connected networks. *Proceedings - IEEE INFOCOM*, 1721–1729. http://doi.org/10.1109/INFCOM.2007.201

Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, *43*(5–6), 907–928. http://doi.org/citeulike-article-id:230211

Hua, M., & Pei, J. (2010). Probabilistic Path Queries in Road Networks: Traffic Uncertainty Aware Path Selection. *Proceedings of the 13th International Conference on Extending Database Technology*, 347. http://doi.org/10.1145/1739041.1739084

Huang, H., & Liu, C. (2009). Query evaluation on probabilistic RDF databases. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5802 LNCS*, 307–320. article. http://doi.org/10.1007/978-3-642-04409-0_32

Hurley, P. (2014). *A concise introduction to logic*. Nelson Education.

Jensen, F. V. (1996). *An Introduction to Bayesian Networks*. *UCL press London*. UCL press London.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '03*, 137. http://doi.org/10.1145/956755.956769

Khan, A., Bonchi, F., Gionis, A., & Gullo, F. (2014). Fast Reliability Search in Uncertain Graphs. *Edbt*, (c), 535–546. article.

Khan, A., & Chen, L. (2015). On Uncertain Graphs Modeling and Queries. In *VLDB 2015 Tutorial*. article.

Klahr, David and Langley, P. (1987). *Production system models of learning and development*. MIT press.

Krötzsch, M., Vrandečić, D., & Völkel, M. (2006). The Semantic Web - ISWC 2006. *The Semantic Web - ISWC 2006*, *4273*(February 2016), 935–942. http://doi.org/10.1007/11926078

McClelland, J. L., Rumelhart, D. E., & McClelland, J. L. (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 2: Psychological and Biological Models. *Parallel Distributed Processing Explorations in the Microstructure of Cognition Volume 1 Foundations*, *1*, 567. Retrieved from http://www.osti.gov/energycitations/product.biblio.jsp?osti_id=5838709

Minsky, M. (1974). A Framework for Representing Knowledge. *MIT-AI Laboratory Memo 306*.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., … Gardner, M. (2015). Never-Ending Learning. *Aaai*.

Noy, N., & McGuinness, D. (2001). Ontology development 101: A guide to creating your first ontology. *Development*, *32*, 1–25. http://doi.org/10.1016/j.artmed.2004.01.014

O'Hagan, A. (2005). Research in Elicitation. In S. K. Upadhyay, U. Singh, & D. K. Dey (Eds.), *In Bayesian Statistics and its Applications* (pp. 375–382). Anamaya: New Delhi.

Ontology, S. (2016). SWEET Ontology. Retrieved from https://sweet.jpl.nasa.gov/

Phoenix. (2016). D2.2 - Specifications of the work packages. Retrieved September 1, 2016, from http://www.phoenix-project.eu/

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*(1–2), 107–136. http://doi.org/10.1007/s10994-006-5833-1

Roussey, C., Pinet, F., Kang, M. A., & Corcho, O. (2011). *An Introduction to Ontologies and Ontology Engineering*. *Ontologies in Urban Development Projects*. http://doi.org/10.1007/978-0-85729-724-2

Schank, R. C., & Abelson, R. P. (1975). Scripts, Plans, and Knowledge. *Proceedings of the 4th International Joint Conference on Artificial Intelligence*, 151–157.

Sowa, J. F. (1991). Principles of Semantic Networks: Explorations in the Representation of Knowledge. *Morgan Kaufmann Series in Representation and Reasoning*, (1987), xi, 582.

Stuart, J. R., & Peter, N. (2003). *Artificial Intelligence a Modern Approach*. *Artificial Intelligence* (Vol. 72). http://doi.org/10.1017/S0269888900007724

Udrea, O., Subrahmanian, V. S., & Majkić, Z. (2006). Probabilistic RDF. *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI-2006*, 172–177. http://doi.org/10.1109/IRI.2006.252408

Wang, J., Zhu, J., & Yang, H. (2013). Reliable path selection problem in uncertain traffic network after natural disaster. *Mathematical Problems in Engineering*, *2013*. http://doi.org/10.1155/2013/413034

Wang, Y., Li, X., Li, X., & Wang, Y. (2013). A survey of queries over uncertain data. *Knowledge and*

Information Systems, *37*(3), 485–530. http://doi.org/10.1007/s10115-013-0638-6

Yaman, A., Coler, M., & Iacca, G. (2015). PHOENIX Knowledge Base. Retrieved from http://www.phoenix-project.eu/

Yaman, A., Coler, M., & Iacca, G. (2016). Knowledge Elicitation, Representation and Reasoning [WP3 Tutorial]. Retrieved from http://www.phoenix-project.eu/

Yuan, Y., Wang, G., Chen, L., & Ning, B. (2016). Efficient Pattern Matching on Big Uncertain Graphs. *Information Sciences*, *339*, 1–54. http://doi.org/10.1016/j.ins.2015.12.034

Zadeh, L. A. (1988). Fuzzy logic. *Computer*, *21*(4), 83–93. http://doi.org/10.1109/2.53

Zhu, K., Zhang, W., Zhu, G., Zhang, Y., & Lin, X. (2011). BMC: An efficient method to evaluate probabilistic reachability queries. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6587 LNCS*(PART 1), 434–449. article. http://doi.org/10.1007/978-3-642-20149-3_32

4 Appendix

4.1 Glossary

Collaborative knowledge elicitation: the process in which experts involved in the knowledge elicitation process provide their knowledge in the same domain.

User: Anyone with an objective of exploration a difficult-to-access environment. The user may or may not have expertise in the process or the methodology of exploration.

Expert: Selected individuals for knowledge elicitation process to include their knowledge in the Phoenix Knowledge Base.

Knowledge Engineer: Experts involved in the process of building and maintaining knowledge bases.

Exploration goal: The knowledge about an unknown environment that the user wants to gain. It is identified from the user interaction starting from the user question.

Hypothesis: Proposal for a starting point of investigating an exploration goal where there is no or limited amount of evidence about its truth.

Experiment: The process of performing co-evolutionary and/or reincarnation cycles.

Exploration task/process: the process of performing experiments to answer the exploration goal. It can be decomposed into smaller exploration tasks/processes. An exploration task/process can be iterative where experiments are performed multiple times.

4.2 A Simplified Class Diagram

“A simplified class diagram designed for the Phoenix workshop 19/09/2016 (by Ahmed Hallawa, Stephan Schlupkothen, Giovanni Iacca and Anil Yaman).