UDC 004.82:519.17

Fuzzy Conceptual Graphs for Knowledge Representation in Process-Oriented Organizations

E. J. Azofeifa, G. M. Novikova

Department of Information Technologies
Peoples' Friendship University of Russia
6, Miklukho-Maklaya str., Moscow, Russia, 117198

The use of fuzzy conceptual graphs for representing knowledge in process-oriented organizations is considered. Two types of knowledge, procedural and declarative, are discussed; the difference between them is shown along with representation and usage details in knowledge bases. A formal definition of fuzzy conceptual graphs is given, and their ability to represent declarative and procedural domain knowledge in a simple and understandable way is shown. The structure of the knowledge base includes three levels: an ontological layer, which contains concepts and integrates declarative and procedural knowledge; a middle or interface layer, which describes business processes based on real data; and the ground layer, the layer of real (historical) data, which collects primary information about the current state of objects and the relationships between them. The difference between the types of information on each of the layers of the knowledge base is shown, as well as the application method of fuzzy conceptual graphs on the interface layer. A description of the mechanisms of interaction and rules for reflecting data from the ground layer to the interface layer is provided. Mathematical methods for analysis of primary data and fuzzy knowledge indicators are described, which aid in decision-making, optimization and refinement of procedural knowledge systems.

Key words and phrases: Fuzzy conceptual graphs, knowledge representation, business process, knowledge base, ontology, intelligent agent.

1. Introduction

The span of Business Intelligence (BI) tools and similar approaches is due to grow steadily in the next years, following the current expansion of information in organizations. Many companies already consider BI as a necessity, more than a competitive advantage. Consequently, specializations or branches of BI technologies have been emerging during the last years. One of them is Business Process Intelligence (BPI), or the application of BI techniques to business processes [1]. BPI relates to the emerging data inside of a company; namely, the information concerning processes and the resources associated to them. However, BPI is prone to encounter issues that, most likely, have been dragged through the lifecycle of the company. Some of these problems are related to technological limitations, like absence of event logging, while others are related to Knowledge Representation (KR), like incompatibilities in the ontological knowledge among business units or lack of standardized process documentation.

Conceptual Graphs (CGs) are an intuitive and easily understandable means to represent knowledge [2, p. v], and they can be combined with Fuzzy Logic (FL) to create Fuzzy Conceptual Graphs (FCG), which are focused on the approximate representation of imprecise knowledge [3, p. 1]. The aim of this work is to present FCGs as a suitable formalism for KR and reasoning in process-oriented organizations. For this purpose, a Knowledge Base (KB) structure is proposed based on the FCG formalism. Mapping between declarative and procedural knowledge is discussed, and an interface between ontological and factual knowledge is presented. Fuzzy approaches to quantify the characteristics of the KB are proposed, including a pseudometric based on the flow of information through entities.

Received 26th February, 2016.

2. Background

Knowledge representation can be thought of as an internal representation of reality inside an intelligent agent. It consists of symbols that represent a limited collection of propositions about the world, and as such, is only able to approximate reality. Choosing a specific type of KR will determine how and what to perceive from reality.

Ontologies are abstract models consisting of concepts that are relevant for describing the real world [4, p. 30]. They provide a set of terms used to represent reality, and for this reason, they are considered the heart of every knowledge representation. They are typically built on top of a taxonomy, which is a hierarchical structure of concepts using the relation "is a kind of" among them. However, they vary according to which kind of knowledge they hold. Two types of knowledge can be distinguished [5]. Declarative knowledge describes facts or the understanding that something is true. Procedural knowledge, on the other hand, refers to the ability to perform a task in an efficient way [4, p. 48].

To establish a basis for reasoning, ontologies must be embedded into a suitable framework, which includes a formal system of logic and an efficient computational environment. Conceptual graphs are a logically precise formalism which can represent knowledge in a humanly readable, and computationally manageable form [6]. Fuzzy logic is an extension of traditional logical systems, which offers the framework for dealing with uncertainty and imprecision. In contrast to exact KR, FL is a much more similar way to represent the human mind [7].

A CG is composed of concept nodes representing entities and relation nodes representing relationships between these entities. A concept node refers to a specific entity by adding an individual marker to its label. Otherwise, the concept node refers to an unspecified entity. In fuzzy logic, an object is said to be of a type with an uncertainty and/or truth degree. The main difference between CGs and FCGs is only that CGs are based on basic concept types and basic relation types, whereas FCGs are based on fuzzy concept types and fuzzy relation types [3, p. 36].

A basic CG [2, p. 26] is composed of a vocabulary $V = (TC, TR, I)$ and a 4-tuple $G = (C, R, E, l)$ where:

- TC is the set of concept types.
- TR is the set of relation symbols, which is partitioned into subsets $T_{1R}, \dots, T_{kR}$ of relation symbols of arity $1, \dots, k$, respectively.
- I is the set of individual markers, $*$ denotes the generic marker and $M = I \cup \{*\}$ is the set of markers.
- (C, R, E) is a finite, undirected and bipartite multigraph denoted $graph(G)$.
- C is the concept node set.
- R is the relation node set.
- E is the family of edges.
- l is a labeling function that satisfies:
  - A concept node c is labeled by a pair $(type(c), marker(c))$, where $type(c) \in TC$ and $marker(c) \in I \cup \{*\}$.
  - A relation node r is labeled by $l(r)$ or $type(r) \in TR$.
  - Edges incident to a relation node r are totally ordered and labeled from 1 to $arity(type(r))$. An edge labeled i between a relation r and a concept c is denoted (r, i, c).

Computer systems that make use of human knowledge are referred to as knowledge-based systems (KBS). A typical KBS consists at least of a knowledge base containing organized knowledge represented by KR formalisms, and a reasoning engine with mechanisms to imitate the problem solving process of a human expert [4, p. 54]. Organized knowledge in a KB refers to ontologies, facts, rules and constraints [2, p. 2]. Factual knowledge makes assertions about a concrete situation.

The simplest KR model consists of a KB with ontologies and facts [2, p. 312]. When working with complex systems, however, other kinds of components have to be added to the model in order to correctly describe the state of the world and its evolution in time. Inference rules represent implicit or general knowledge about a domain, which helps to fully describe a world. Evolution rules transform factual knowledge: they represent possible actions leading from one world to another.

3. A KB using the FCG formalism

The following is an approach to structuring declarative and procedural knowledge in a KB using the FCG formalism. On this proposal, ontological and factual knowledge are bound together by CGs structured in layers. There is a middle layer (interface) between the ontological and factual (ground) layers. Rules and constraints are context-dependent, and they are assumed to be developed aside from the ontological structure.

3.1. Ontological layer

Consider an enterprise XYZ divided in different departments or business units. These units are organized in a hierarchical or non-hierarchical way, or a combination of both, with varying levels of interdependence. This interdependence between departments can be expressed in several ways, depending on the task at hand: for example, interdependence measures could be the number of shared resources between business units, or the distributed weight of the different channels of communication between departments (See Figure 1).

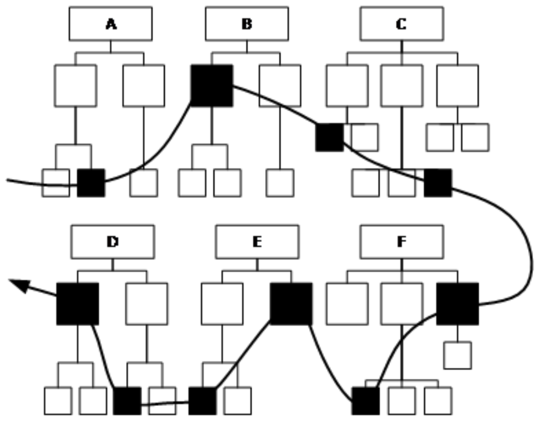

Figure 1. Interdependent process

Basically, physical or abstract entities, together with the relations between them, are commonly shared (up to a certain degree) among the units of a company. However, these dependencies are often ignored or hidden for several reasons, which contributes to problems in corporate visibility. In companies with large operations, for example, multiple users share and manipulate information or assets, even if they do not know about each other's actions. This can give rise to inconsistencies or issues, which require prompt and efficient solutions in order to resume normal operation. In the ideal scenario of a company with full documentation of its ontologies and visibility of its processes, finding solutions to this kind of problems is straightforward. In a real scenario, on the contrary, there can be flaws and inconsistencies in the form or content of the knowledge representation, which could prevent or limit problem solving.

Consider the same company XYZ, with a number n of different business units. Each one of these departments possesses a set of ontologies of type o in the form of CGs, where type o could be either declarative or procedural. The set of n departments is part of a CG representing the hierarchical structure of the company, with the top of the hierarchy being the company itself. If at least one of the business units possesses an ontology and, more specifically, its set of vertices (concepts) is not empty, then the maximal join [2, p. 218], [8] of the ontologies of each unit, together with the taxonomy, results in an upper or top-level ontology, denoted by T, which is itself a CG.

Notice that process analysis relies on the existence of documented business processes or their generation from event logs. In the proposed KB, existing business processes have to be embedded into the ontological layer as ontologies of the procedural type, maintaining the overall CG structure. Naturally, newly discovered business processes should update the ontological knowledge. Several approaches exist to merge declarative and procedural knowledge. For example, the Static and Dynamic Knowledge Representation Framework (SD-KRF) [9] expresses processes, nodes, tasks, events, transitions, actions and decisions as ontology classes. A method for creating ontological knowledge is based on providing compatibility to a set of tools for modeling processes and corporate infrastructure [10].

3.2. Interface layer

Factual knowledge asserts that some entities exist and that they are related by some relationships [2, p. 22]. It is obtained after transforming raw historical (or real-time) data into useful information for analysis and decision-making. In this KB, a way of expressing factual knowledge is proposed. It consists in merging the obtained information from historical data into the upper ontology T, creating a new layer on the CG structure. The newly created layer, which is denoted interface layer, is the boundary between ontologies and historical data and serves as the ground on which business processes take place.

Three general rules are proposed for the generation of this interface. First, relations on the interface layer cannot exist between any arbitrary kind of concepts or relations, but only between instances of concepts or relations. It is important to understand the difference between an instance (instantiated type) and a grounded instance [11, p. 92]. An instance refers to a more precise rendering of a certain concept or relation by means of a specific configuration of its attributes, with the exception of those that express quantification or multiplicity (quantifiers) and grounding (determiners). An instance is an unspecified entity in a CG. However, instances cannot come into existence without first being grounded. An example of an instance would be a concept with a special configuration of its aspect attributes (e.g. size, color, shape). A grounded instance would be the same instance with added attributes of unique identification, quantity and a specific position in space and time. It corresponds to a specific entity in a CG with an individual marker.

The second property constrains the representations on the interface layer to procedures or processes. Specifically, relations between concept instances are expected to represent activities or events of a process execution. This property can be considered as a qualitative projection of the observed world into an ontology, because it determines which concept instances interrelate and how. Thereby, the interface layer is the ground on which most process analyses take place.

The third proposed rule states that the relations between instances on the interface layer should be weighted, which means that some relations will be stronger than others with magnitudes that range over an interval (i.e. from 0 to 1), representing a chosen value (e.g. cost, benefit, implication or causation). Notice that this property enforces a transformation of the layer from a normal CG into a FCG. This property can be considered as a quantitative projection of the observed world into an ontology, because it assigns magnitudes to the qualitative relations defined on the second property. Hence, a structure for the interface layer is proposed as follows.

Consider a basic CG G defined over a vocabulary V, where $graph(G)$ is constrained to be an undirected bipartite graph and the set of individuals $I = \emptyset$; then every node in G corresponds to an instance that is not grounded. Furthermore, the set R can be generalized to contain other kinds of nodes, e.g. actors, *demons*, constraint overlays [12], in order to enable procedural expression on the CG. A restriction is set so that all relations are binary (n-ary relations can be transformed polynomially into binary ones [2, p. 360]). Edges labeled 1 correspond to arcs exiting a node heading to a relation, and edges labeled 2 are directed the opposite way [2, p. 114]. Thus, G corresponds to the interface layer.

Let c be a concept node and r a relation node on the interface layer, $i \in \{1, 2\}$ and $E_c = \{(r*, i*, c*) \in E | c* = c\}$ the set of edges adjacent to the node c. The number of edges incident to c that are labeled i is expressed as follows:

$$q(c) = \sum_{b \in E_c} n_b * u(b),$$

where n_b is the number of occurrences of the grounded edge b. The function u is then defined as follows:

$$u(n) = \begin{cases} 1 & \text{if } l(b) = i, \\ 0 & \text{if } l(b) \neq i, \end{cases}$$

where l is the labeling function. A function $p(c, b)$ denoting the probability of entering or exiting node c through edge b is defined as follows:

$$p(c, b) = \frac{n_b}{q(c)}.$$

Thus, the probability of exiting or entering node c is expressed as follows:

$$P = \sum_{b \in E_c} p(c, b) * u(b),$$

which evaluates to 0 or 1 depending on the existence of exiting or entering edges. When $i = 1$, the function p is denoted as the node routing through b. Updating the function with $i = 2$ provides the probability of entering the node c from b, which is denoted as the node inflow from b. In presence of n-ary relations, the same can be extended to a relation node c by applying p with $i = 1$ to obtain the inflow and $i = 2$ for the routing.

In the interface layer, however, relations are constrained to be binary. This means that both edges incident to a relation will have the same value of p. Thereby, the function $p(c, b)$, where b is incident to a relation r, will be denoted as the *fuzzy membership* of r on the interface layer. Thus, the underlying CG is transformed to a FCG, which can be visualized by a directed graph on which relations are represented by fuzzy edges (See Figure 2).

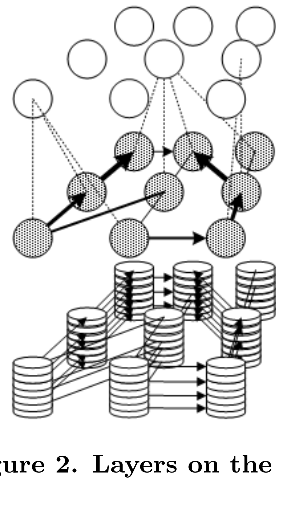

Figure 2. Layers on the KB

A quantification of the relation of a node to the entire layer is also proposed. Consider the function $f(c)$:

$$f(c) = \frac{q(c)}{2\sum_{r\in R} n_r},$$

where nr is the cardinality of the set of grounded instances of the relation r. When $i = 1$, f represents the exit factor or $ext(c)$, which is the weight of the exiting grounded edges from c in relation to the totality of grounded relations. When $i = 2$, the entering edges to c are considered, and this is denoted as the entrance factor or $ent(c)$. From both exit and entrance factors a correspondent pseudometric can be obtained. Consider the following expression:

$$d(x, y) = |f(y) - f(x)|,$$

where x, y are concept nodes. This corresponds to a percentage of the totality of grounded relations. When $i = 1$, function d is denoted as exit distance or d_{ext}. When $i = 2$, the correspondent entrance distance or d_{ent} is obtained. Consider the following relation:

$$h(c) = \frac{ext(c)}{ext(c) + ent(c)}.$$

The function h corresponds to the flow factor. If $h(c) > 1/2$, c is considered a source, with a degree of truth or fuzzy membership of $h(c)$. It is considered a sink if $h(c) < 1/2$, with a degree of truth of $1 - h(c)$. The flow factor can be assigned to nodes as the fuzzy membership value, depending on the task.

3.3. Ground layer

The proposed ground layer is composed of grounded instances of concepts and relations between them. This layer contains the historical (or real-time) data formatted as CG structures. However, this layer is unconstrained in relation to any ontology. In other words, grounded instances, which have been verified not to belong to any ontology, can be preserved on the layer. This is a crucial requirement of any process-oriented system, because it allows process discovery by means of process mining [13, p. 10].

4. Support for process analysis in the KB

4.1. Multiple contexts

Possible interdependence between work units and their ontologies or business processes is not the only source of complexity for analysis: the kind of analysis that is needed can be different every time, depending on a variety of factors. For example, analysis approaches can be generalized for the whole company while considering only a specific kind of resource, or they can be specific for a defined process while considering all the interrelations with adjacent processes. A multiplicity of scenarios can be brought to mind in which a static kind of knowledge is not suitable for the analysis of the task at hand. Therefore, it is necessary to consider the context of the analysis that will take place, so that the appropriate instantiations can be performed (See Figure 3).

A new instantiation of the interface layer can be obtained by defining a subset of all the concepts or relations that satisfy a certain condition (a filtering operation) and generating the interface layer in the ordinary way over the resulting node subset. New instantiations of the layer can also be obtained by extracting all the grounded instances that satisfy a certain condition, and thus obtaining fuzzy values for the interface nodes by means of performing a certain operation over the resulting grounded subsets. Combinations of these approaches can also be suitable for complex analyses.

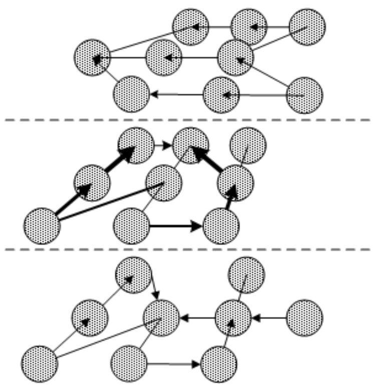

Figure 3. Interface instantiations

4.2. Process analysis

Graph search over the interface layer provides an interesting array of possibilities for process analysis. Business metrics can be reflected on concept instances on the layer, making it possible to visualize the correlations between them. Abnormal behavior of a metric can be determined by critical factor analysis (CFA) over the layer [1]. A straightforward approach to CFA is to translate a subgraph of the interface layer into a decision tree, in which degrees of membership and fuzzy relations of the graph can represent uncertainty. This creates a visual representation of every outcome from a starting node. If a visual approach is complicated to obtain, a path to a certain goal state (metric) from an initial state on the interface layer can be created by means of graph search techniques. A search can further be delimited by applying a set of constraints over the interface layer, e.g. preventing it to enter certain nodes where a condition is valid. The probability of ending in the goal state is obtained by the product of the fuzzy values of the edges of the path, assuming that they are normalized.

If only a specific business metric needs to be analyzed, an approach is to compare the relevant nodes with the proposed exit and entrance distances, as a measure for impact or priority. Furthermore, if the extracted process data comprises information about allocation of resources, it is possible to perform capacity planning. Groups of instances can be considered as separate entities subsumed under a common concept. By adding to the layer groups of instances, together with the correspondent constraints, a normal search can provide appropriate resource allocations to achieve the desired performance. The proposed flow factor can provide insight about existing bottlenecks and expose possible focus points for improvement or optimization.

5. Conclusion

The present work proposes the use of Conceptual Graphs with Fuzzy Logic for representing knowledge in process-oriented environments. Conceptual Graphs were discussed as a suitable and straightforward approach for representing and unifying knowledge. For this reason, a Knowledge Base structure based on Fuzzy Conceptual Graphs was proposed, with a high-level of abstraction. The KB features three layers: an ontological layer, merging declarative and procedural knowledge; a middle or interface layer, where real business processes are projected from historical or real-time data; a ground layer, where concrete instances and relations among them are stored for retrieval. The following coefficients are proposed to support process analysis on the interface layer: entrance factor, exit factor, entrance distance, exit distance, flow factor. Possible methods to analyze processes over the KB are exposed to provide evidence for the suitability of the proposed KB for process-oriented organizations.

References

1. J. Cardoso, W. Aalst, Handbook of Research on Business Process Modeling, Information Science Reference, Hershey, PA, 2009.
2. M. Chein, M. Mugnier, Graph-Based Knowledge Representation, Springer, London, 2009.
3. T. Cao, Conceptual Graphs and Fuzzy Logic, Springer-Verlag, Berlin, 2010.
4. G. Jakus, V. Milutinović, S. Omerović, S. Tomažič, Concepts, Ontologies, and Knowledge Representation, Springer, Heidelberg, 2013.
5. G. M. Novikova, An integral knowledge representation model for an intelligent management system, in: E. V. Popov (Ed.), Dynamic intelligent systems, Russian Central House of Knowledge, pp. 110–112.
6. J. F. Sowa, Logic: Graphical and algebraic, Manuscript (1997).
7. L. Zadeh, Toward extended fuzzy logic - a first step, Fuzzy Sets and Systems 160 (21) (2009) 3175–3181.
8. C. Laudy, J. Ganascia, C. Sedogbo, High-level fusion based on conceptual graphs, in: 10th International Conference on Information Fusion, 2007.
9. F. Smith, M. Proietti, Behavioral reasoning on semantic business processes in a rule-based framework, in: Communications in Computer and Information Science, 2014, pp. 293–313.
10. N. Silega, T. Loureiro, M. Noguera, Model-driven and ontology-based framework for semantic description and validation of business processes, IEEE Latin America Transactions 12 (2) (2014) 292–299.
11. R. Langacker, Foundations of Cognitive Grammar: descriptive application, Vol. 2, Stanford University Press, California, USA, 1991.
12. J. Lee, L. F. Lai, W. T. Huang, Task-based specifications through conceptual graphs, IEEE Expert 11 (4) (1996) 60–70.
13. W. Aalst, Process mining, Springer, Berlin, 2011.

УДК 004.82:519.17

Нечеткие концептуальные графы для представления знаний в процессно-ориентированной организации

Э. Х. Азофейфа, Г. М. Новикова

Кафедра информационных технологий
Российский университет дружбы народов
ул. Миклухо-Маклая, д. 6, Москва, Россия, 117198

Рассматривается использование нечетких концептуальных графов для представления знаний в процессно-ориентированных организациях. Рассмотрены два типа знаний — процедурные и декларативные, показано их различие и особенности представления и использования в базах знаний. Дано формальное определение нечетких концептуальных графов. Показаны их возможности для представления в простой и понятной форме как декларативных, так и процедурных знаний предметной области. Предложена структура Базы Знаний, которая включает три уровня: онтологический слой, содержащий концепты понятий и интегрирующий декларативные и процедурные знания; средний слой, описывающий схемы бизнес-процессов на основе исторических данных, формирующихся на базовом уровне; нижний слой, слой реальных данных, где собирается первичная информация о текущих состояниях объектов и отношений между ними. Показано отличие типов информации на каждом из слоев базы знаний, а также способ использования нечеткого концептуального графа на средне-интерфейсном слое. Рассмотрены механизмы взаимодействия и правила преобразования информации между средне-интерфейсным слоем и базовым слоем. Описан математический аппарат и методы анализа первичной информации для поддержки принятия решений по оптимизации и уточнению процедурных знаний системы, а также показатели, используемые в процессе анализа нечетких знаний.

Ключевые слова: нечёткие концептуальные графы, представления знаний, бизнес-процесс, базы знаний, онтологии, интеллектуальный агент.

Литература

1. *Cardoso J., Aalst W.* Handbook of Research on Business Process Modeling. — Hershey, PA: Information Science Reference, 2009. — Pp. 456–480.
2. *Chein M., Mugnier M.* Graph-Based Knowledge Representation. — London: Springer, 2009.
3. *Cao T.* Conceptual Graphs and Fuzzy Logic. — Berlin: Springer-Verlag, 2010.
4. Concepts, Ontologies, and Knowledge Representation / G. Jakus, V. Milutinović, S. Omerović, S. Tomažič. — Heidelberg: Springer, 2013.
5. *Новикова Г. М.* Интегрированная модель представления знаний в интеллектуальной системе управления // Динамические интеллектуальные системы / под ред. Э. В. Попов. — ЦРДЗ. — С. 110–112.
6. *Sowa J. F.* Logic: Graphical and Algebraic. — Manuscript. — 1997.
7. *Zadeh L.* Toward Extended Fuzzy Logic - a First Step // Fuzzy Sets and Systems. — 2009. — Vol. 160, No 21. — Pp. 3175–3181.
8. *Laudy C., Ganascia J., Sedogbo C.* High-Level Fusion Based on Conceptual Graphs // 10th International Conference on Information Fusion. — 2007.
9. *Smith F., Proietti M.* Behavioral Reasoning on Semantic Business Processes in a Rule-Based Framework // Communications in Computer and Information Science. — 2014. — Pp. 293–313.
10. *Silega N., Loureiro T., Noguera M.* Model-Driven and Ontology-Based Framework for Semantic Description and Validation of Business Processes // IEEE Latin America Transactions. — 2014. — Vol. 12, No 2. — Pp. 292–299.
11. *Langacker R.* Foundations of Cognitive Grammar: descriptive application. — California, USA: Stanford University Press, 1991. — Vol. 2.
12. *Lee J., Lai L. F., Huang W. T.* Task-Based Specifications Through Conceptual Graphs // IEEE Expert. — 1996. — Vol. 11, No 4. — Pp. 60–70.
13. *Aalst W.* Process mining. — Berlin: Springer, 2011.