International Journal for Quality Research 10(3) 595–606
ISSN 1800-6450

Paweł Lula [1]
Monika Hamerska

ONTOLOGY-BASED MODELS FOR QUALITY OF LIFE MEASUREMENT AND COMPARISON

Article info:
Received 07.03.2016
Accepted 20.07.2016

UDC – 332.05
DOI – 10.18421/IJQR10.03-11

Abstract: *The main goal of the paper is to propose a framework for building ontology-based models for quality of life evaluation and similarity analysis of complex objects derived from social domain. The paper is composed with several sections. The subsequent parts of the paper contain general remarks about measurement of quality of life, description of ontology-based models and details of the proposed approach. Conclusion and bibliography are placed in the final part of the paper*.

Keywords: *ontology-based models, quality of life measurement, analysis of complex data*

1. Introduction

There is no doubt that ontology-based models can describe complex systems very well. It was confirmed be researches from various fields of science. These models are characterized by high flexibility and interpretability and wide scope of application.

The main goal of the paper is to propose a framework for building ontology-based models representing objects from social domain. The proposed model is equipped with analytical tools for evaluation of objects' quality and for performing cluster analyses. Hierarchical structure and possibility of customization cause that the model can perform multifaceted calculations such as quality of life assessment.

The paper is composed with several sections. The first part presents general remarks about measurement of quality of life. Next ontology-based models are presented. Further sections show details of the proposed approach. Conclusion and bibliography are placed in the final part of the paper.

2. Quality of life measurement

Quality of life (*QoL*) is a resultant evaluation of positive and negative aspects of life expressed for individuals or social groups. The concept of *QoL* embraces among others psychological, sociological, economic, health-related and cultural aspects. A survey of different definitions of *QoL* concept was presented in (Barcaccia *et al.*, 2013). Quality of life evaluation is a mixture of objective indicators and subjective assessments. The compound structure of *QoL* idea has a huge impact on its measurement and analysis.

It is rather impossible to express quality of life using basic, one-dimensional indicators which reflect only one aspect of life (for example health and economic situation). It

[1] Corresponding author: Paweł Lula
email: pawel.lula@uek.krakow.pl

was shown by Townsend who proved that poverty could not be described only by low income (Townsend, 1979). Therefore complex character of *QoL* should be expressed by aggregated indicators or by complex structures.

Expressing quality of life by a single value is very convenient. Human Development Index (http://hdr.undp.org/en/content/human-development-index-hdi) introduced by United Nations is a good example of this approach. Interesting proposal of multidimensional *QoL* indicator was proposed in (Watson *et al.*, 2016).

Many researchers state that one measure is not able to capture actual level of quality of life and recommend to use complex structures (e.g. vectors) for its description. For example Ruut Veenhoven proposed four-dimensional measurement of quality of life which describes *chances* (opportunities) and *outcomes* (results, effects) for the environment and for individuals (Veenhoven, 2000). Similar approach is advised by Eurostat in (Eurostat, 2011). Authors of this report strongly advised to define separate indicators for different fields (through aggregation of sub-fields measures) and not to combine them into single value.

This short survey shows that *QoL* may be described by aggregated indicators or by a set of specific indexes grouped in one complex structure. It seems that these two approaches should be covered by a computational model proposed here.

The general overview of the model is presented in next section of the paper. Next four main levels of the model are discussed: properties, components, entities and problem domain.

3. General overview of the ontology-based model of quality of life

We assume that the quality of life model should:

- have hierarchical structure,
- have an ability to use heterogeneous and complex data,
- be universal, flexible and easy for customization,
- be helpful for *QoL* measurement and comparison.

Above requirements justify an application of ontology-based approach to modelling.

Ontology in computer science is a formal tool for modelling complex systems. In 1993 Gruber defined ontology as “explicit specification of a conceptualization” (Gruber, 1993). Borst in 1997 wrote that ontology is a “formal specification of a shared conceptualization” (Borst, 1997). A similar definition was proposed in (Studer *et al.*, 1998) where it is stated that “an ontology is a formal, explicit specification of a shared conceptualization.”

It allows to describe a domain of interests, with complex entities existing in it and relationships between entities and between their properties. From formal point of view ontological models are based on labelled and weighted graphs.

Ontology-based models have many advantages, for example:

- they can be used for description multi-layered, compound systems,
- entities can have heterogeneous features represented by properties,
- ontology-based models are relatively easy for interpretation and implementation.

We would like to propose four-layered model for quality of life modelling and analysis. At every level different data structures and different functions are defined. The set of data structures and functions used in the model can be extended to new features. General characteristic of the model is presented in Table 1.

Table 1. General characteristic of multi-layered model for quality of life modelling and analysis

Level	Scope of analysis	Main data structures	Main functions
1.	Domain of interested	Set of trees	• Multi-criteria analysis of entities (ranking), • Cluster analysis of entities
2.	Entity	Tree	• Description of entity structure, • Quality evaluation of a given entity, • Calculation similarity to another entity
3.	Component of a given entity	Node of a tree	• Modelling of entity components, • Quality evaluation of components, • Calculation similarity to another component
4.	Property	Field in a node	• Storing values describing a given component

Source: Own elaboration

Table 2. Short description of an illustrative model

Level	Scope of analysis	Real object	Data structure	Main tasks
1.	Domain of interested	A settlement with its inhabitants	Set of trees	• Analysis of the whole set of inhabitants – for instance ranking them according to their *QoL*
2.	Entity	An inhabitant	Tree	• Representation of one inhabitant, • Quality function defined at this level allows to evaluate *QoL* for one inhabitant • Similarity function calculates a similarity to another inhabitant
3.	Component of a given entity	One aspect of an inhabitant	Node of a tree	• For example “economic factors” for an inhabitant which can be further divided into “incomes”, “expenditures” or “real properties” – representing by sub-nodes of a node “economic factors” • Functions for quality of similarity evaluation related to a given aspect (e.g. quality function for assessment economic indicator of *QoL*)
4.	Property	One feature of a given aspect	Field in a node	• For example a field “base salary” in a node “incomes” for a given inhabitant

Source: Own elaboration

The model as a whole represents a domain of interest which is equivalent to the scope of analysis. In the domain a set of objects represented by entities exists. At the domain level of analysis functions for processing the whole set of objects are defined. These functions are responsible for entities' ranking or clustering.

Every entity is represented by one tree. Tree structure should reflect the composition of studied objects and correspond to these object's components which have impact on quality evaluation. At this level of the proposed model also functions for quality evaluation of the entity and for similarity calculation to another entity are defined.

Tree nodes correspond to entities' components. The root node of the tree can be treated as a pointer to the entity as a whole. Descendants of the root node represent main components on the entity. Every descendant can have its sub-nodes. Also for every node some functions can be defined. They serves to calculate a quality of a component represented by a given node and to measure a similarity to another component. It is possible to define several different quality or similarity functions for every node. Values calculated by different node-level functions can be aggregated by entity-level functions.

The significance of components is represented by weights assigned to connection leading to them. All weights should be non-negative and all weights assigned in components starting from the same node should sum to one (if not, they are normalized during calculation).

Every entity component (represented by a node) can have properties represented by fields. A number and type of fields for every component can be different.

To illustrate a proposed approach an illustrative example can be considered. Let's assume that a model for quality of life evaluation of a group of inhabitants of a settlement is studied. The role of model's elements is presented in Table 2.

4. Properties

Properties store actual values describing a component which is represented by a given node. Values stored in properties are used for quality evaluation and for comparison analysis of nodes and entities. Due to huge diversity of types of date used for object description and quality analysis, broad range of properties' types can be used:

- values expressed on typical measurement scales:
  - nominal (binary and multistate variables),
  - ordinal (e.g. Likert scale),
  - interval,
  - ratio,
- textual descriptions (useful for open questions in surveys),
- taxonomy-based values (hierarchical, ontology-based),
- ranges (of ordinal or numeric values),
- sets,
- sequences (ordered sets),
- distributions.

In case of necessity new types of values can be introduced into model.

For every type of data it is necessary to define formulas for calculation similarity between them. We understand similarity as a measure showing proximity of two objects. The greater value of similarity measure, the smallest distance between compared objects. For practical reasons we assume that similarity measure is normalized to a range $[0; 1]$. Similarity can be defined for all types of data.

Exemplary measures for similarity calculation for different types of data are presented in Table 3.

Table 3. Measures for similarity calculation

Type of data	Similarity measure
Binary	• A survey presented in (Choi & Cha, 2010)
Nominal	• Hamming similarity measure (Hamming, 1950), • Hamming weighted measure • A survey presented in (Boriah, Chandola, & Kumar, 2008)
Ordinal	• Canberra metric (Riccia, Dubois, Lenz, & Kruse, 2009) • Similarity based on normalized rank transformation (Teknomo, 2015) • Kendall distance • Walesiak distance (Walesiak, 1999)
Interval	• different types of Minkowski distances (e.g. Euclidean or Manhattan distance)

Ratio	• different types of Minkowski distances (e.g. Euclidean or Manhattan distance)
Textual	• cosine similarity • distances based on topic models • ontology-based distances (Budanitsky & Hirst, 2001)
Hierarchical	• measures based on path length • Information-theoretic similarity (e.g. Resnik similarity, Lin similarity) (Resnik, 1995), (Lin, 1998)
Ranges	• Jaccard index
Sets	• Jaccard index • Tanimoto distance • Sørensen–Dice coefficient
Sequences	• Edit distance
Distributions	• A survey presented in (Cha, 2007)

Source: Own elaboration.

5. Components

5.1. Node structure

Entity components are represented by nodes of a given tree. Usually a node has compound and heterogeneous character. Data describing a component are stored in node's properties (fields). Also for every property a reference value (optimal value for a property) can be kept. The comparison of actual and reference values stored in a node has crucial significance for *QoL* assessment.

A list of sub-nodes is also a vital element of a node. As has been mentioned above sub-nodes should represent various aspects of a component represented by a current node.

Table 4. The structure of a node

Node	
• Properties	• Reference values for properties
• List of sub-nodes (with weights)	
• Functions for: ○ quality evaluation, ○ similarity calculation.	

Source: Own elaboration

The formal description of the node also incorporates definitions of functions for quality evaluation and similarity calculation. A given component (represented by a node) can have one or more functions of these two types.A node's structure is shown in the Table 4.

Nodes can be divided into two groups: inner nodes (which have descendants) and leaves (without descendants). Every inner node has a list of sub-nodes which importance (and impact on a given node) is defined by weights.

5.2. Quality evaluation

Every node should be equipped with one or more more functions for evaluation node's quality. For a given node this task is performed by a function f_Q^N:

$$f_Q^N\colon \boldsymbol{P} \times \boldsymbol{R} \times \boldsymbol{D} \rightarrow [0;1] \quad (1)$$

where:

$\boldsymbol{P}$ – domain of node's properties,

$\boldsymbol{R}$ – domain of reference values for properties,

$\boldsymbol{D}$ – domain of node's descendants.

During quality assessment realized for a given node, the function f_Q^N should not only compare $\boldsymbol{P}$ (actual, real values) and $\boldsymbol{R}$ (reference, expected, optimal values) but also should refer to qualities of descendants. Therefore it is possible to define f_Q^N for a N_x node as:

$$f_Q^N(N_x) = \alpha_x f_{QP}^N(P_x, R_x) + (1 - \alpha_x) f_{QD}^N\big(descend(N_x)\big) \quad (2)$$

where:

$f_{QP}^N(P_x, R_x)$ – sub-function for evaluation of quality related to properties and their reference values,

$f_{QD}^N\big(descend(N_x)\big)$ – sub-function for evaluation of quality related to descendants of a N_x node,

α_x – defined for a node N_x impact coefficient for sub-functions.

Assuming that a node N_x has descendants defined as:

$$descend(N_x) = \{N_{x,1}, N_{x,2}, \dots, N_{x,S}\}$$

we can define $f_{QD}^N\big(descend(N_x)\big)$ as:

$$f_{QD}^N\big(descend(N_x)\big) = \sum_{k=1}^{S} w_{x,k} f_Q^N(N_{x,k}) \quad (3)$$

where $w_{x,k}$ is a weight assigning to connections starting at N_x and leading to k-th its sub-node.

Finally, the function $f_Q^N(N_x)$ can be presented as:

$$f_Q^N(N_x) = \alpha_x f_{QP}^N(P_x, R_x) + (1 - \alpha_x) \sum_{k=1}^{S} w_{x,k} f_Q^N(N_{x,k}) \quad (4)$$

We can see that $f_Q^N(N_x)$ has recursive character. It means that for quality evaluation of N_x node all direct and indirect descendants ought to be analysed.

It's very important to keep measures of quality normalized to the range $[0; 1]$.

5.3. Similarity and difference calculation

For similarity calculation between two nodes the function f_S^N is used:

$$f_S^N: (\boldsymbol{P} \cup \boldsymbol{D}) \times (\boldsymbol{P} \cup \boldsymbol{D}) \to [0; 1] \quad (5)$$

For two nodes N_x and N_y a similarity function should take into account similarity of nodes' properties and similarity of nodes' descendants:

$$f_S^N(N_x, N_y) = \frac{\beta_x + \beta_y}{2} f_{SP}^N(P_x, P_y) + \left(1 - \frac{\beta_x + \beta_y}{2}\right) f_{SD}^N\left(descend(N_x), descend(N_y)\right) \quad (6)$$

where:

f_{SP}^N – sub-function for similarity evaluations between P_x and P_y (properties of both nodes),

f_{SD}^N – sub-function for similarity evaluations between $descend(N_x)$ and $descend(N_y)$ (descendants of both nodes),

β_x, β_x – impact coefficients for sub-functions (respectively for node N_x and N_y.

Assuming that both nodes have identical sets of properties we can define f_{SP}^N as:

$$f_{SP}^N(P_x, P_y) = f_{SP}^N\begin{pmatrix} sim(p_{x1}, p_{y1}), sim(p_{x2}, p_{y2}), \dots, \\ sim(p_{xN}, p_{yN}) \end{pmatrix} \quad (7)$$

where $sim(p_{xi}, p_{yi})$ is a similarity measure calculated for a i-th property in two nodes.

Also the result of f_{SP}^N function should be normalized to the range $[0; 1]$.

Unfortunately, in contrast to properties, the assumption concerning the same number and the same type of descendants for both nodes is not realistic. Also we should take into account that corresponding descendants from node N_x and from node N_y usually have different weights assigned. Therefore we define $f_{SD}^N\left(descend(N_x), descend(N_y)\right)$ as:

$$f_{SD}^{N}\left(descend(N_x), descend(N_y)\right) = \frac{f_{SD}^{N}\left(descend(N_x)\rightarrow descend(N_y)\right)+f_{SD}^{N}\left(descend(N_y)\rightarrow descend(N_x)\right)}{2} \quad (8)$$

where:

$f_{SD}^{N}\left(descend(N_x) \rightarrow descend(N_y)\right)$ – is a measure of similarity of $descend(N_x)$ to $descend(N_y)$,

$f_{SD}^{N}\left(descend(N_y) \rightarrow descend(N_x)\right)$ – is a measure of similarity of $descend(N_y)$ to $descend(N_x)$.

To calculate above values it is necessary to create a similarity matrix (**SM**) between descendant nodes:

$$\mathbf{SM} = \begin{bmatrix} f_S^N(N_{x,1}, N_{y,1}) & \dots & f_S^N\left(N_{x,1}, N_{y,S_y}\right) \\ \dots & \dots & \dots \\ f_S^N(N_{x,S_x}, N_{y,1}) & \dots & f_S^N\left(N_{x,S_x}, N_{y,S_y}\right) \end{bmatrix} \quad (9)$$

where $f_S^N(N_{x,\mathrm{i}}, N_{y,\mathrm{j}})$ is a similarity measure between i-th descendant of N_x and j-th descendant of N_y.

Next an optimal assignment problem ought to be solved to find pairs of the most similar descendants of N_x and N_y nodes. It can be done with Hungarian algorithm (Kuhn, 1955).

During $f_{SD}^{N}\left(descend(N_x) \rightarrow descend(N_y)\right)$ calculation for every $N_{x,k}$ node the closest $N_{y,\tilde{k}}$ is taken and after it a following formula is used:

$$f_{SD}^{N}\left(descend(N_x) \rightarrow descend(N_y)\right) = \frac{\sum_{k=1}^{S_x} w_{x,k} f_S^N\left(N_{x,k}, N_{y,\tilde{k}}\right)}{\sum_{k=1}^{S_x} w_{x,k}} \quad (10)$$

If the number of descendants of both nodes is not the same than some non-existing, dummy nodes are added to compensate it. Similarities for pairs containing dummy nodes are set to 0 and weights to dummy nodes or nodes paired with dummy nodes to 1.

In similar manner the $f_{SD}^{N}\left(descend(N_y) \rightarrow descend(N_x)\right)$ is calculated:

$$f_{SD}^{N}\left(descend(N_y) \rightarrow descend(N_x)\right) = \frac{\sum_{k=1}^{S_y} w_{y,k} f_S^N\left(N_{y,k}, N_{x,\tilde{k}}\right)}{\sum_{k=1}^{S_y} w_{y,k}} \quad (11)$$

Also the case on unequal number of descendants should be taken into account.

Finally, the similarity $f_S^N(N_x, N_y)$ can be expressed using the following formula:

$$f_S^N(N_x, N_y) = \frac{\beta_x + \beta_y}{2} f_{SP}^N(P_x, P_y) + + \frac{1}{2}\left(1 - \frac{\beta_x+\beta_y}{2}\right)\left(\frac{\sum_{k=1}^{S_x} w_{x,k} f_S^N\left(N_{x,k}, N_{y,\tilde{k}}\right)}{\sum_{k=1}^{S_x} w_{x,k}} + \frac{\sum_{k=1}^{S_y} w_{y,k} f_S^N\left(N_{y,k}, N_{x,\tilde{k}}\right)}{\sum_{k=1}^{S_y} w_{y,k}}\right) \quad (12)$$

The calculation of $f_S^N(N_x, N_y)$ has recursive character and consecutive calls of $f_S^N(\dots,\dots)$ function are performed until leaves of a tree are reached.

All similarity function f_S^N, f_{SP}^N and f_{SD}^N should be symmetric. That means:

$$f_S^N(N_x, N_y) = f_S^N(N_y, N_x) \tag{13}$$

$$f_{SP}^N(P_x, P_y) = f_{SP}^N(P_y, P_x) \tag{14}$$

$$f_{SD}^N(D_x, D_y) = f_{SD}^N(D_y, D_x) \tag{15}$$

6. Entities

An entity is a representation of a real object studied during analysis. An entity is described by a tree which reflects entity's structure (its components). For every entity functions for quality and similarity calculation are defined. The structure of an entity is shown in the Table **5**.

Table 5. The entity definition

Entity
• Tree
• Functions for: ○ quality evaluation, ○ similarity calculation

Source: Own elaboration

6.1. Quality evaluation for entities

Quality function for an entity can be defined as:

$$f_Q^E : \boldsymbol{E} \rightarrow \boldsymbol{Q} \tag{16}$$

where $\boldsymbol{E}$ is a domain from which entities come from and:

$$\boldsymbol{Q} = \{Q_1, Q_2, \dots, Q_P\} \tag{17}$$

is a structure which stores the results of calculation. Operations performed during quality function execution calculate subsequent components of $\boldsymbol{Q}$. According to remarks presented in the previous section of the paper a f_Q^N function for quality evaluation for a root node of a given tree returns a quality measure calculated on the basis of all nodes in the tree. Despite it, there is a key difference between f_Q^N and f_Q^E functions. The f_Q^N function always returns a measure from a period $[0; +1]$. Whereas f_Q^E can return any, often complex structure (for example a vector which elements describe various aspects of quality). Very often f_Q^E function uses f_Q^N (executed for a root node of a tree) for calculations values of $\boldsymbol{Q}$'s elements.

From practical point of view function f_Q^E can calculate various indicators which constitute the holistic evaluation of an entity (for example different indicators showing economic, psychological and sociological aspects of the quality of life).

6.2. Similarity calculation for entities

Also similarity calculation for a given entity is based on adequate functions defined for a root node of a tree. Having f_S^N functions for a root node of the tree, a function $f_S^E(E_x, E_y)$ for similarity evaluation between E_x and E_y entities can be defined:

$$f_S^E : \boldsymbol{E} \times \boldsymbol{E} \rightarrow [0; 1] \tag{18}$$

where $\boldsymbol{E}$ is a set of all entities (domain for entities).

A f_S^E function can be defined as an composition of f_S^N functions defined for root nodes of entities for which similarity is calculated.

7. Domain of interest

Domain of interest constitutes an environment in which entities exist. Definition of a domain of interest contains description of functions which can operate on the whole set of entities. Among them there are functions for multi-criteria analysis and clustering of entities.

The structure of the domain of interest definition is presented as a Table 6.

Table 6. The domain of interest definition

Domain of interest
• Set of entities (represented by trees)
• Functions for: o multi-criteria analysis of entities, o entities clustering

Source: Own elaboration.

7.1. Multi-criteria analysis of entities

Using the approach presented in this paper quality indicators $Q_1, Q_2, \ldots, Q_P$ for every entity can be calculated. Different aspects of quality expressed by indicators can vary in significance which can be specified by weights associated with indicators.

All these data form input matrix which can be further processed by multi-criteria methods. The form of the matrix is presented in Table 7.

Table 7. The input data for multi-criteria analysis

Quality indicators		Q_1	Q_2	Q_3	...	Q_P
Weights		w_1	w_1	w_1	...	w_1
Entities	E_1	$q_{1,1}$	$q_{1,1}$	$q_{1,1}$	...	$q_{1,1}$
	E_2	$q_{1,1}$	$q_{1,1}$	$q_{1,1}$	...	$q_{1,1}$
	...	...	...	...	...	...
	E_T	$q_{1,1}$	$q_{1,1}$	$q_{1,1}$	...	$q_{1,1}$

Source: Own elaboration

Thanks to multi-criteria analysis, by means of proper algorithms entities can be ordered in accordance with established criteria. The issue is widely discussed in literature (e.g. Greco, 2006; Trzaskalik, 2013).

7.2. Cluster analysis of entities

Similarity evaluation between objects represented by entities can be treated as a starting point for cluster analysis which allows to investigate the structure of objects and to find patters in data set. Cluster analysis is performed on the basis of similarity matrix between entities calculated by $f_S^E(E_x, E_y)$ function (the structure of input data is shown in Table 8).

Table 8. The input data for cluster analysis

		Entities			
		E_1	E_1	...	E_1
Entities	E_1	$f_S^E(E_x, E_y$	$f_S^E(E_x, E_y$	...	$f_S^E(E_x, E_y$
	E_1	$f_S^E(E_x, E_y$	$f_S^E(E_x, E_y$	...	$f_S^E(E_x, E_y$
	...	...	...	...	...
	E_1	$f_S^E(E_x, E_y$	$f_S^E(E_x, E_y$	...	$f_S^E(E_x, E_y$

Source: Own elaboration

The topic of cluster analysis is presented in many books (e.g. Everitt *et al.*, 2001).

8. Conclusion

In the paper an ontology-based framework for complex objects representation and analysis was presented. The model is oriented on quality evaluation of objects, their components and their sets. Also the problem of similarity analysis between complex objects was widely discussed.

In our opinion the proposed approach is very flexible and can be used for different types of data and for solving different types of tasks.

The issue of program implementation was not discussed in the paper but similar framework was partly implemented by authors in Java (Tuchowski *et al.*, 2011). Experience gained during this project showed that Java language meets all requirements connected with implementation of ontology-based solutions.

References:

Barcaccia, B., Esposito, G., Matarese, M., Bertolaso, M., Elvira, M., & Marinis, M. De. (2013). Defining Quality of Life: A Wild-Goose Chase? *Europe's Journal of Psychology*, *9*(1). article. Retrieved from: http://ejop.psychopen.eu/article/view/484

Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. *In Proceedings of the eighth SIAM International Conference on Data Mining*, 243–254. inproceedings.

Borst, W.N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Enschede.

Budanitsky, A., & Hirst, G. (2001). Semantic distance in WordNet : An experimental , application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics.* Pittsburgh.

Cha, S.-H. (2007). Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, *1*(4). misc.

Choi, S., & Cha, S. (2010). A survey of Binary Similarity and Distance Measures. *Journal of Systemics, Cybernetics and Informatics*, 43–48. article.

Eurostat. (2011). *Multidimensional measurement of the quality of life*.

Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster Analysis*. book, Wiley. Retrieved from: https://books.google.ie/books?id=htZzDGlCnQYC

Greco, S. (2006). *Multiple Criteria Decision Analysis: State of the Art Surveys*. book, Springer New York. Retrieved from: https://books.google.pl/books?id=lzyNasqOxAQC

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *KNOWLEDGE ACQUISITION*, *5*, 199–220. article.

Hamming, R. W. (1950). Error Detecting and Error Corecting Codes. *Bell System Technical Journal*, *29*(2), 147–160. article.

Kuhn, H.W. (1955). The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, *2*, 83–97. article.

Lin, D. (1998). An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning {(ICML} 1998), Madison, Wisconsin, USA, July 24-27, 1998*, 296–304. inproceedings.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1* (pp. 448–453). inproceedings, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from: http://dl.acm.org/citation.cfm?id=1625855.1625914

Riccia, G., Dubois, D., Lenz, H. J., & Kruse, R. (2009). *Preferences and Similarities*. book, Springer Vienna. Retrieved from: https://books.google.pl/books?id=qqf6BJZKAUMC

Studer, R., Benjamins, R., & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, *25*(1–2), 161–198. article.

Teknomo, K. (2015). Similarity Measurement. Retrieved from: http://people.revoledu.com/kardi/tutorial/Similarity

Townsend, P. (1979). *Poverty in the United Kingdom*. London: Allen Lane and Penguin Books.

Trzaskalik, T. (2013). *Analiza wielokryterialna: wybrane zagadnienia*. book, Wydawnictwo Uniwersytetu Ekonomicznego. Retrieved from: https://books.google.pl/books?id=gBL0oAEACAAJ

Tuchowski, J., Wójcik, K., Lula, P., & Paliwoda-Pekosz, G. (2011). OBCAS - An Ontology-Based Cluster Analysis System. In S. Wrycza (Ed.), *Research in Systems Analysis and Design: Models and Methods* (pp. 106–112). inproceedings, Berlin: Springer.

Veenhoven, R. (2000). The Four Qualities of Life. *Journal of Happiness Studies*, *1*(1), 1–39. article. http://doi.org/10.1023/A:1010072010360

Walesiak, M. (1999). Distance Measure for Ordinal Data. *Argumenta Oeconomica*, *8*(2).

Watson, D., Maître, B., Whelan, C., & Russell, H. (2016). *Technical Paper on the Measurement of Multidimensional Quality of Life in Ireland*. Dublin. Retrieved from https://www.esri.ie/pubs/BKMNEXT310.pdf

Paweł Lula
Cracow University of Economics,
Cracow
Poland
pawel.lula@uek.krakow.pl

Monika Hamerska
Cracow University of Economics,
Cracow
Poland