Supplement for
“Development of a Web-Based Expert System for Strategic Management using a Knowledge Graph-Based Methodology”

İlter İrdesel
Magneti Marelli
Bursa, Turkiye
ilter.irdesel@magnetimarelli.com

Gürdal Ertek
College of Business and Economics
United Arab Emirates University
Al Ain, UAE
gurdal@uaeu.ac.ae

Ahmet Demirelli
Faculty of Engineering and Natural Sciences
Sabancı University
Istanbul, Turkiye
ahmetdemirelli@sabanciuniv.edu

Lakshmi Kailas
College of Business and Economics
United Arab Emirates University
Al Ain, UAE
lakshmi.kailas@gmail.com

Abstract

This supplement file is for the research paper titled “Development of a Web-Based Expert System for Strategic Management using a Knowledge Graph-Based Methodology”. This supplement document supports the main research paper, by providing further details on literature, design of the visual languages, mathematical modeling, steps of the project, technology selection, assessment of the developed expert system, and a discussion of expert systems and Small and Medium Enterprises (SMEs).

Index Terms

knowledge graphs, knowledge representation, decision support systems, expert systems, strategic planning, data visualization

This research was funded by the United Arab Emirates University, Startup Grant Funding 31B127.

1 LITERATURE

The research presented in this paper is interdisciplinary, drawing from the principles of rule-based systems, knowledge representation, information visualization, graph theory, and strategic management. In this section, related literature in these fields is presented.

1.1 Rule-Based Expert System (RBES)

An RBES and is composed of the following principal components: the *knowledge acquisition facility* allows the specification of knowledge as rules, transforming tacit knowledge into explicit knowledge; the *knowledge base* contains the rules used in the reasoning; the *working storage* stores the database of facts learned for a particular problem situation; the *inference engine* acts on the rules based on the facts provided by the user, and makes suggestions; the *user interface* enables the communication of the user with the expert system [38]. The rules in RBES are expressed as IF *Antecedent* THEN *Consequent*.

The abundance and even overflow of information brings difficulties in absorbing, organizing, processing, and using such vast amounts of information by humans. RBES can be considered as part of narrow artificial intelligence (Narrow AI), as such systems conduct reasoning in a narrow application domain, by processing the rules and other information in knowledge bases.

1.2 Domain-Specific Visual Languages (DSVLs)

This paper proposes an ordered set of domain-specific visual languages (DSVLs) for expert system development. Specifically, a graph-based methodology is proposed to describe expert system rule bases. Then, the proposed expert system methodology and visual languages are applied in a case study in the domain of strategic management. DSVLs support "specifying systems at a high-level of abstraction, using a notation very close to the problem domain and quite intuitive for domain experts" (Rivera *et al.*, 2010 [74]). DSVLs are used in many domains, including educational games (Marchiori *et al.*, 2011 [54]), home automation (Drey and Consel, 2012 [22]), probabilistic reasoning (Erwig and Walkingshaw, 2013 [28]), and emulating software testing environments (Liu *et al.*, 2016 [51]).

Empirical studies have shown that visual languages possess several advantages over text-based languages for software development (Corral et al., 2019 [18]; Weintrop and Wilensky, 2019 [83]). However, the development of visual languages requires precise specification of the vocabulary and grammar. To this end, graph grammars can be a suitable formalism to describe the syntax of visual languages (Grunske *et al.*, 2008 [37]). Triple graph grammars (TGGs) are used to indicate the relations between two visual languages (Leblebici *et al.*, 2017 [47]).

Visualization in business and management is advocated in visual language literature (Eppler and Bresciani, 2013 [26]; Al-Kassab, 2014 [44]), and called upon as "an idea whose time has come" (Eppler and Bresciani, 2013 [26]). Business-related applications of DSVL include enterprise systems security management (Almorsy and Grundy, 2014 [4]), policy definition (Bottoni and Fish, 2013 [10]), and business process management (Laue and Awad, 2011 [46]).

1.3 Knowledge Acquisition Methodologies for Expert Systems

In the expert system development process, knowledge acquisition (gathering, elicitation, and refinement), is the first stage, which is crucially important. Knowledge acquisition "is the process of obtaining knowledge and transforming that knowledge into a representational form that can then be used by the expert system for making decisions" (Gupta and Nagpal, 2020 [38]). Wagner *et al.* (2001) [82] report that knowledge acquisition is the greatest bottleneck in the expert system development process, because of unavailability of experts and knowledge engineers, as well as the difficulties of the rule extraction process.

Wagner *et al.* (2001) [82], while presenting a taxonomy of expert systems according to the task domains in production/operations management, provide a summary of empirical research on knowledge acquisition. The authors list as a guideline, which knowledge acquisition technique has been selected in each research cited in their paper.

An early system that automates knowledge acquisition is the Logic Aids for Problem Solving (LAPS) system, which has been used to produce an expert system in the submarine diving domain (Di Piazza, 1991 [68]). Another early system presented in the literature is CARMEN, an expert system generator built on a methodology that models both declarative domain knowledge and procedural problem-solving structure (Ang and Tong, 1997 [5]). Wu *et al.* (2000) [88] described a goal-driven knowledge acquisition tool that captures both declarative and procedural knowledge. The knowledge acquisition algorithms in Wu *et al.* (2020) [88] construct a graph-theoretic representation, similar to our work. In a more recent study, Ali *et al.* (2018) [3] introduced a knowledge acquisition procedure that extended across all the significant stages in the knowledge engineering process.

The methodology presented in this paper has been developed by employing a task-specific approach (Ang and Tong, 1997 [5]) to analyze and model the problem solving of domain experts. First, a vocabulary that can comprehensively describe and encapsulate domain knowledge has been developed. This vocabulary aims to facilitate knowledge acquisition and encoding, as well as accomplish consistency and transformability between stages. Then, customized subsets of the vocabulary, and customized grammars were specified for each stage. Another example of a task-specific approach is described by Siddiqui *et al.* (2016) [77] who integrated it in a project that supports an industrial robot to chart its surroundings efficiently using visual aids.

1.4 Expert Systems for Strategic Management

Expert systems have been developed in almost every domain, where problem solving is nontrivial. Since this paper presents a case study in the domain of strategic management, studies in this application domain are summarized in this section. Earlier work on expert system development for strategic management is listed in the following survey papers: Davidson and Tenaglia (1993) [19] provide an early directory of strategic management software tools, which also includes Alacity Strategy, a competitive expert system with 2,500 rules, and Business Insight, a strategy expert system. Li *et al.* (2002) [49] listed 11 decision support systems for strategic marketing planning. Metaxiotis and Psarras (2003) [57] presented a review of business applications of expert systems, listing 42 expert system applications in marketing, banking and finance, forecasting, and other areas. Their review also lists expert systems that aid in strategic decisions.

Some of the recent applications of expert system development for strategy development and decision making are as follows: Arias-Aranda *et al.* (2010) [6] present a fuzzy expert system ESROM that aims to enhance the knowledge in organizations and thus help a firm's leaders to achieve their targets more efficiently. Grahovac and Devedzic (2010) [36] developed a cost management expert system (COMEX) that can help in making efficient cost-related decisions. Lendel and Varmus (2010) [48] proposed an expert system that focuses on developing innovation strategies in an organization. Oh *et al.* (2012) [63] proposed a fuzzy expert system to enable portfolio management based on a company's operations and strategy. Min (2013) [58] presented a rule-based expert system that helps make decisions based on examining sales records collected from mobile devices. Tekez (2018) [78] developed a web-based expert system aimed at efficient handling of demand at a furniture factory by considering corporate goals. Relich and Pawlewski (2018) [73] developed a case-based reasoning technique to estimate the cost of new product development (NPD) with mass customization. Zhao *et al.* (2019) [90] used a case-based reasoning technique to support green retrofit decisions in new constructions.

2 Research Topic and Contributions to Literature

2.1 Research Topic

Motivated to solve the challenges in knowledge acquisition, representation, and processing, as well as the gap in the literature, the topic of this research is the development of a multistage visual methodology for expert systems development, which is illustrated in Figure **??**.

The core idea behind the methodology is to use customized knowledge graph-based visual languages for knowledge representation for each project team member (human agent) at each stage of the expert system development process. Such an approach aligns each corresponding visual language to the agent's roles, tasks, and attributes.

In Figure **??**, the round boxes are the knowledge representation schemes proposed in our methodology, the first three of which are visual schemes for knowledge representation. The text below the boxes represents the tasks/processes in the respective stages. Arrows show the interactions between project team members in the expert system project.

The proposed methodology targets cognitive alignment and is especially applicable in domains where there is a gap between the technical competencies of the project team members. One frequently encountered case is when the domain expert's technological competence is low, and the decision problem is at the strategic level. When such a gap exists, a significant challenge arises: How can the representations −customized to the team members at each stage− be integrated for an agile expert system project? The proposed methodology also resolves this challenge through graph transformation algorithms that transform (translate) alternative representations of successive stages. In other words, the proposed methodology not only proposes customized visual languages but also translation algorithms between these languages. Similar to how the knowledge acquisition system of [3] covers all major phases of data mining, the methodology proposed in the present paper covers all major phases of expert system development.

2.2 Novelties of the Contributions

The novelties of the study's contributions are as follows:

Contribution 1 is mostly novel. Commercial or open-source development tools such as VisiRule [1], Actico [2] and Drools [3] provide high-quality modeling facilities for business analysts, system designers, and software engineers, but not for domain experts. The methodology presented in our study is much more comprehensive and complete, because it considers and covers all the agents involved in expert system development and integrates the stages. Although the idea of role-based knowledge flows was proposed by Liu *et al.* (2013) [50], they did not describe a rule-based expert system.

Contribution 2 is also mostly novel. The proposed methodology can be complete only after specifying the respective visual languages for the stages, each responding to the respective requirements of that stage. This is accomplished in the present study. [17] reviewed the studies that use knowledge graphs in recommender systems. Husáková and Bureš (2020) [40] reviewed the literature focusing on the utility of ontologies during construction of information systems. Five key functions which ontologies support have been identified through this exercise, which also used mind maps to consolidate the large amount of information in the literature. Nalepa (2018) [60] presented and illustrated models for devising and implementing rule-based intelligent systems using knowledge engineering. Following this knowledge engineering perspective, each proposed visual language is formally described. Chen (2010) [15] proposed an ontology-based knowledge representation and reasoning framework, and Chergui *et al.* (2020) [16] proposed a methodology to acquire an individual's tacit knowledge by using an ontology. Other knowledge representation frameworks built using ontologies have been presented in several domains in [89] [84] [8] [31], as well as others. However, these frameworks do not cater to various users simultaneously, as presented for the first time in this paper.

Contribution 3 is, to the best of our knowledge, the first in expert system literature; the various visual representations in the successive stages are integrated by associating them with one another. This integration is formally described through mathematical formalism, including graph transformation algorithms that link pairs of successive stages. Chein and Mugnier (2009) [14] and Ramaswamy *et al.* (1997) [72] have been the principal sources for guiding the formal specification of languages. Graph transformations have also been studied in detail by Ehrig *et al.* (2015) [24]. However, graph transformations between different visual rule base representations were not encountered.

1. https://www.visirule.co.uk/
2. https://www.actico.com/platform/business-rules/
3. https://drools.org/

2.3 Relation to Knowledge Graphs

The present study builds on knowledge graphs (semantic networks) [42]. The proposed methodology encapsulates and represents knowledge as a knowledge graph (semantic network, semantic graph); concepts are shown with nodes (vertices) and the semantic relationships between the concepts are shown with directed arcs (edges) connecting the concept nodes [91] [32]. In the present study, we propose the adoption and implementation of knowledge graphs for knowledge representation and sharing for expert systems development. As example application of knowledge graphs is Long *et al.* (2019) [52], where semantic networks are used to represent supply chain network knowledge obtained through computational experiments. In a related study, López-Cuadrado et al. (2012) [53] presented a "collaborative semantic framework for knowledge sharing" based on a "shared and controlled vocabulary." Diamantini et al. (2016) [21] presented and illustrated, through a web application, a semantic methodology that creates and facilitates the sharing of performance indicators. The framework in our paper goes one step further than the mentioned studies, by describing how knowledge graphs with the same vocabulary yet different grammars can be transformed into each other.

3 Design of the Visual Languages

3.1 Design Principles

In the present research, when designing the visualization schemes for the first two stages, the Gestalt Principles of School of M. Wertheimer (Wertheimer, 2020 [85]), the knowledge visualization guidelines of Eppler and Burkhard (2008) [27], and the ontology design guidelines of Gavrilova (2007) [33] were applied. These principles have also been applied in the case study to represent the knowledge of the selected domain.

The design principles applied in the research are as follows:

1) The five-step methodology of Gavrilova (2007) [33] was applied. These steps are glossary development, laddering (defining principal levels of abstraction), disintegration (breaking high-level concepts into detailed ones), categorization (structuring hierarchy), and refinement (updating the visual structure to facilitate cognition).
2) The visual representation should reflect the semantic of the knowledge.
3) Concepts (nodes in the mind map) are linked to the parent by only one type of relationship at each level.
4) The order of node types at a particular stage is always the same. For Stage 1, this order is *modules, suggestions (patterns), objects, attributes, sub-attributes*, and *(attribute) values*. For Stage 2, this order is *objects, attributes, sub-attributes, (attribute) values*, and *suggestions (patterns)*.
5) The depth of the branches should be approximately the same.
6) Cross-links should be avoided, to preserve the simple mind map structure.
7) Symmetry should be preserved as much as possible.
8) Color and icons are used to distinguish various node types.
9) Attribute values are marked with icons, to distinguish the attribute values that trigger the rules.

3.2 Mind Maps

The proposed methodology enables the expert to structure and represent knowledge through the mind map of the first stage. The mind map encodes knowledge in a hierarchical form, and the visual encoding is translated into IF-THEN rules in the successive stages of the methodology. Because the cognitive load of the mind map is low (Eppler, 2006 [25]), the extracted knowledge is concurrently tested by the expert.

In the developed methodology, the mind map encapsulates the explicitly stated IF-THEN rules directly, and the know-how from business cases indirectly by representing the know-how as IF-THEN strategy rules. Thus, our approach integrates case-based reasoning with traditional rule-based reasoning.

Mind maps, attributed to Buzan and Buzan (1995) [13], are extensively recognized and used worldwide. Mind maps also appear in the academic literature, and the studies most relevant to the one here are listed in this section.

The mind map was selected as a visual archetype in Stages 1 and 2 of the developed methodology, for knowledge acquisition from the knowledge source. The reasons for this choice are as follows.

1) Mind map is the simplest structure among candidate visual methods (Eppler, 2006 [25]) and is thus accessible to domain experts of any technical background.
2) Mind mapping mimics the way the brain works, with ideas being associated and linked (Serrat, 2017 [75]), and thus appeals to human cognition.
3) The mind map is sufficient to model IF-THEN rules with logical operators (with certain limitations), despite its simplicity. In the case study, the complete domain of strategic management was reflected through mind mapping, without the need to extend mind maps with additional vocabulary.
4) The mind map can encapsulate not only the rules, but also the *semantics* of the rules, thanks to its hierarchical structure, and special grammars (the sequence of node types in the hierarchy) can be defined for it.
5) With mind maps, knowledge can be structured as it is acquired.
6) Mind mapping has already found extensive recognition and application for problem solving, especially in the business world.

Wheeldon (2011) [86] examined the efficacy of mind maps in qualitative research by helping participants skillfully structure their expressions of previous experiences. Adodo (2013) [2] studied the benefits of mind mapping for students in enhancing their thinking proficiency. Edwards and Cooper (2010) [23] elaborated on the advantages of mind mapping as a teaching tool in clinical education.

A mind map can be considered an ontology because ontology refers to a hierarchical structuring of terms for representing any general domain (Gavrilova *et al.*, 2015 [34]). A mind map can also be considered a semantic network (graph), where nodes (vertices) represent concepts and arcs (edges) describe semantic relations between concepts (Zhu and Iglesias, 2017 [91]). Even though graph-based schemes exist for knowledge representation in expert system design, the schemes proposed in this paper are novel in the way they structure knowledge. Furthermore, these schemes are integrated within the methodology, and the transitions between them are formally described.

The mind map was originally proposed by Buzan and Buzan (1995) [13]), while the concept map, a generalization of the mind map allowing cycles and labels over the branches, was developed by Joseph Novak (Novak and Gowin, 1984 [61]). A comparison between different tools, such as mind maps and concept maps, is given in Davies (2011) [20] and Abbas *et al.* (2018) [1].

3.3 Popularity of Mind Mapping Tools

The usage and popularity of mind mapping and its tools are increasing. The mind mapping software used in this study was FreeMind[4]. FreeMind software has been downloaded more than 25 million times as of June 2021 and more than half million times in the year 2020[5], making it one of the most popular mind mapping software in the world. An alternative to FreeMind, MindMeister.com mind mapping service[6] has been used by more than 20 million users worldwide.

3.4 Drawbacks of Visualization

While the benefits of visualization in expert system development have been extensively discussed so far, a discussion of possible drawbacks also deserves mention. This section presents the discussion on drawbacks.

In a survey of Business Rule Management Systems (BRMS), (Ohlsson, 2006) [64] observed that the drawbacks of visual modeling of rules apply only to advanced users, not managers. (Ohlsson, 2006 [64]) also gives high scores to VisualRules (the only visual modeling system in the survey) for almost all interface-related characteristics, such as ease of use, ease of learning, validation-verification ability, management tools, and providing guidance. Bresciani and Eppler (2015) [11] conducted an extensive review of the visualization literature and presented a classification of related common errors. The authors compiled and expanded on a matrix of pitfalls based on two categories of causes (designers and users) and three types of outcomes (cognitive, emotional, and social). Mitigating for the listed root causes can enable effective management of the risks involved in using visualization for expert system development. Thus, according to these two studies, the use of visualization in expert system development may only limit some of the agents, specifically, only those with the most technical skills. To solve this problem, the process can always revert to the default rule modeling methodologies, primarily text-based ones, when advanced use is required.

4 METHODOLOGY

4.1 Novelty

The novelty of the methodology is that it supports four sequential stages, with visual representations at each of the first three stages customized for the tasks and agents of that stage, and with links between the stages through graph transformation algorithms.

4.2 Justification

The cascaded visualizations for different stages can be justified by the observation of Whyte *et al.* (2016) [87], who reported on the complexity of visual representations in project-based tasks and how "cascading" of these representations empowers different project members to comprehend and collaborate better.

Motivated by the effectiveness of graph-based knowledge representations and of multiple visual formats in a project environment, the proposed methodology integrates customized graph-based representations for the different stages of expert system development.

A graph-based knowledge can be justified by the advantages for knowledge it brings with respect to facilitating knowledge modeling and computation: graphs are easily understandable by users, they facilitate reasoning and hence form a compelling means to understand and represent knowledge [14]. Finally, graphs offer the knowledge constructs, such as paths and cycles, that do not exist in logical rule representations.

4.3 Design Criteria

The proposed methodology and its visual grammars were designed by applying select design criteria in the context of the roles and attributes of agents in expert system development. When designing the visualization schemes for the first two stages, the Gestalt Principles of the Psychological School of M. Wertheimer (2020) [85], the knowledge visualization guidelines of Eppler and Burkhard (2008) [27] and the ontology design guidelines of Gavrilova (2007) [33] were selected. These principles have also been applied in the case study, when representing the knowledge for the selected domain, and are described further in Appendix B.1 of the Supplement [41]. Other notable works considered and incorporated in the design include [7], which proposes a knowledge visualization system that can easily adjust to changes in the underlying contexts, [35], which presents a methodology for constructing ontologies for organizing knowledge related to managing projects, and [12], which summarizes the literature focusing on user assessment of graph visualizations.

4. https://sourceforge.net/projects/freemind/
5. https://sourceforge.net/projects/freemind/files/stats/timeline
6. http://www.mindmeister.com/

5 MATHEMATICAL FORMALISM

In this section, a mathematical formalism is presented to describe the constructs and visual grammars of the methodology. The constructs used here are used in the algorithms that transform the graphs of the successive stages into one another.

Language consists primarily of vocabulary and grammar. In the presented methodology, the vocabulary is the same for Stages 1 through 3, although the (visual) grammar is different for each stage.

The terminology and symbols used are adopted from a noteworthy application of graphs in knowledge acquisition by Ramaswamy *et al.* (1997) [72]. The presented formalism can be improved and extended by adopting from other studies; Portmann *et al.* (2015) [69] also describe different types of graphs that facilitate knowledge representation. Other basic graph-theoretic definitions were introduced based on Rahman (2017) [70].

The three main constructs for the proposed visual languages are sets, functions, and graphs.

Sets

- M: set of modules, $m \in M$.
- S : set of suggestions $s \in S$. S_m is the set of suggestions in module m.
- O: set of objects, $o \in O$.
- A: set of attributes and sub-attributes, $a \in A$. For each object, a_1 denotes one of its attributes, and $a_1, a_2, a_{attr(p)}$ denote the sub-attributes, in order of hierarchy. $attr(p)$ refers to the number of levels in path p.
- V: set of all possible attribute/sub-attribute values, $v \in V^* \cup V'$, where V^* is the set of values that fire suggestions (possibly through an $\wedge$ (and) logical operator), and $V^{'}$ is the set of values that do not fire any suggestions.
- L: set of logical operators $\wedge$ (AND), $l \in L$. Each time a logical operator AND is used, it is represented with a different element in set L.
- P: set of paths, $p \in P$.
  $P = \{(m, s, l, o, a_1, a_2, \ldots, a_{attr(p)}, v)\}$, where $attr(p)$ is the number of hierarchical (sub-) attributes.
  $P = P^* \cup P'$, where
  $P^* = \{(m, s, l, o, a_1, a_2, \ldots, a_{attr(p)}, v^*) : v^* \in V^*\}$ is the set of paths that fire suggestions (possibly through an $\wedge$ (and) logical operator), and
  $P^* = \left\{(m, s, l, o, a1, a2, \ldots, attr(p), v^*) : v' \in V^{'}\right\}$ is the set of paths that do not fire any suggestions.
- Q: set of questions.
- X: set of start/finish nodes for the modules: $X = \cup_{m \in M} \{x_m, v_m\}$.

Functions

- $valueset(q)$: function that returns the set of values (firing or non-firing) for question q.

Graphs

- $G(N, A)$: the graph that represents the complete knowledge of the domain, where N is the set of nodes and A is the set of arcs.

The complete vocabulary is

$$N = M \cup S \cup O \cup A \cup V \cup L \cup Q \cup X$$

The vocabulary (nodes) and the grammar (arcs) change for each of the three knowledge graphs G, as follows:

$$G_{Mindmap} = G_1(N_1, A_1)$$

$$G_{DOM} = G_2(N_2, A_2)$$

$$G_{Rulemap} = G_3(N_3, A_3)$$

$$N_1 = N_2 = M \cup S \cup O \cup A \cup V \cup L$$

$$N_3 = M \cup S \cup O \cup Q \cup V \cup L \cup X$$

N_3 differs from $N_1$and N_2 in the following ways:

- Attributes A are not represented as nodes in N_3.
- Questions Q are formed by combining objects O and attributes/sub-attributes A, and are included in N_3 instead of O and A explicitly.
- The start/finish nodes exist in the rule map, so X is included in N_3.

5.1 Graph Transformation Algorithms

The graph transformation algorithms that transform the different knowledge representations are shown in Figures 1, 2, 3, and 4. Unless the graphs where the operations are performed are specified in another way, the operations are assumed to be performed on the mind map. The mind map is the source of the domain knowledge for all three graphs.

In the pseudo codes,

- the symbol $\leftarrow$ denotes an assignment or replacement.
- $|S|$ denotes the cardinality of the set.
- $S, S = a, b, c$ denotes an unordered set.
- $S = (a, b, c)$ denotes an ordered set (tuple).
- $p = \langle a, b, c \rangle$ denotes an ordered set of nodes connected as a path through arcs (a, b) and (b, c).
- (a, S) refers to the element that comes after element a in ordered set S.
- The set operator $\sqcup$ combines two disjoint ordered sets (ex: $S_1 \sqcup S_2$) or adds a new (non-existent) element into a set (e.g., $S_1 \sqcup a$).
- When $\sqcup$ combines two disjoint ordered sets, the sequence for the latter set is appended after the sequence for the first set.
- It is assumed that a subroutine for lexicographical ordering (Harzheim, 2005 [39], pages 88-89) is readily available and can be called from within the transformation algorithms through the call *lexicographical_sort*.
- Only the algorithms in the forward direction, which transform mind map to DOM, and DOM to rule map, are given in the paper because of space limitations.

---

output G_2 **Transform_MindMap_to_DOM**(input G_1)

initial DOM is a replica of the mind map

$G_2(N_2, A_2) \leftarrow G_1(N_1, A_1)$

create a list of all paths that fire a suggestion

$P^* = \langle m, s, l, o, a_1, a_2, \ldots, v^* \rangle$

remove all arcs that emanate from logic nodes and terminate at object nodes

remove $\forall (l, o) \in A_2$

reverse the direction of all arcs from module nodes to logic nodes

$\forall (m, s) \in A_2$

$\quad (m, s) \leftarrow (s, m)$

reverse the direction of all arcs from suggestion nodes to logic nodes

$\forall (s, l) \in A_2$

$\quad (s, l) \leftarrow (l, s)$

an arc is drawn from every firing value to the suggestion that it fires

$\forall p^* = \langle m, s, l, o, a_1, a_2, \ldots, v^* \rangle \in P^*$

$\quad A_2 \sqcup (v^*, l)$

the transformation is complete, and the DOM can be returned as the output

return G_2

Fig. 1. Transformation algorithm that transforms the mindmap into DOM.

output G_3 **Transform_DOM_to_RuleMap**(input G_2)

initial rule map is a replica of the DOM

$G_3(N_3, A_3) \leftarrow G_2(N_1, A_1)$

remove all the arcs in the rule map (arcs will be constructed later)

$A_3 \leftarrow \oslash$

the attributes A are not represented as nodes in N_3

remove $\forall a \in A$

questions Q, which are formed using objects and attributes/subattributes, are included in N_3

the set of questions will be constructed step by step for each module

for each module

$\forall m \in M$

define the set of all objects in that module in the DOM

create set $O_m = \{o : \exists p^* = \langle m, s, l, o, a_1, a_2, \dots, v^* \rangle \in P^*, o \in N_3\}$

sort the objects in O_m lexicographically, turning O_m into an ordered set.

$O_m = \langle o_m 1, o_m 2, \dots, o_m, |O_m| \rangle \leftarrow$lexicographical_sort$(O_m)$

for each object in module m

$\forall o \in O_m$

a start node x_m and finish node y_m is created for each module

create nodes x_m, y_m

$N_3 \sqcup x_m, y_m$

define the set of all firing paths related to the object o in module m

create set $P^*_{mo} = \{p^* : p^* = \langle m, s, l, o, a_1, a_2, \dots, v^* \rangle \in P^*\}$

create the set of questions related to the object o in module m

create set $Q_m o \leftarrow \oslash$

for each firing path p^* for m and o, define its question $q^*$and append to Q_{mo}

$\forall p^* = \langle m, s, l, o, a_1, a_2, \dots, v^* \rangle \in P^*_{mo}$

$q^* = \langle o, a_1, a_2, \dots, v^* \rangle$

$Q^*_{mo} = Q^*_{mo} \cup q^*$

draw an arc from the module m to the object o

$A_3 \cup (m, o)$

sort the questions in Q^*_{mo} lexicographically, turning Q^*_{mo}into an ordered set.

$Q^*_{mo} = (q_{mo1}, q_{mo2}, \dots, q_{m,o,|Q^*_{mo}|})$ *lexicographical_sort*(Q^*_{mo})

draw arcs from each object to its questions

$\forall q \in Q^*_{mo}$

$A_3 \sqcup (o, q)$

define the value set for each question q

$valueset(q) = (v^*, v'_1, v'_2, \dots, v'_n)$

draw arcs from each question to each of its possible values

$\forall v \in valueset(q)$

$A_3 \sqcup (q, v)$

combine all Q^*_{mo} to obtain a sorted Q^*_m

$Q^*_m = (q_1, q_2, \dots, q_n) \leftarrow Q^*_{m1} \sqcup Q^*_{m2} \sqcup \dots \sqcup Q^*_{m|O^*_m|}$

draw arcs from the start node to the first question

$A_3 \sqcup (x_m, q_1)$

draw arcs from each value to the next question in the rulemap, except the last question

Fig. 2. Transformation algorithm that transforms the DOM into Rule Map.

$$\text{for } i = 1 \text{ to } |Q_m^*| - 1$$
$$\quad q \leftarrow q_i$$
$$\quad \forall v \in valueset(q)$$
$$\quad A_3 \sqcup (v, next(q, Q_m^*))$$

draw arcs from each value of the last next question to y_m, the finish node for m

$$q \leftarrow q_{|Q_m^*|}$$
$$\forall v \in valueset(q)$$
$$A_3 \sqcup (v, y_m)$$

draw arcs from firing values to logic nodes

$$\forall p^* = \langle m, s, l, o, a_1, a_2, \ldots, v^* \rangle \in P^*$$
$$\quad A_3 \sqcup (v^*, l)$$

and connect logic nodes to suggestion nodes, if not already connected

$$\quad \text{if } (l, s) \notin A_3 \text{ then}$$
$$\qquad A_3 \sqcup (l, s)$$
$$\text{return } G_3$$

Fig. 3. Transformation algorithm that transforms the DOM into Rule Map (continued).

The stages of the proposed methodology were implemented in the case study as follows:

Stage 1: Mind Maps

In Stage 1, a separate mind map was created for each business function and abstracted as a separate module. As shown in Figure 5, the module name (for example, **Knowledge**) was placed as the root node $m \in M$ of a tree, and each strategy (profit pattern) $s \in S$ within that module is placed as the immediate child node of the root node. Related elements with decreasing importance were placed at successive levels of the tree in the mind map. Following the profit pattern nodes, the tree branches into business objects $o \in O$, object attributes $a \in A$, and attribute values $v \in V$. Icons and node colors (black for objects, blue for object attributes and dark red for attribute values) were used consistently in the mind map to facilitate cognition.

The logical operator **AND** states that a profit pattern applies if all the leading indicators (conditions) listed under the branches of that pattern's node are observed (satisfied). The **OR** operator allows the firing of a suggestion for a particular profit strategy if any of its leading indicators are satisfied. These logical operators $l \in L$ are written over the branches of the mind maps. The mind map representation can handle cases in which the rule contains conditions that depend on several business objects.

Stage 2: Domain Objects Map (DOM)

In Stage 2 of the methodology, the mind map is transformed into another mind map representation, called the Domain Objects Map (DOM), as shown in Figure 6. This representation visualizes the rules in the knowledge base, such that the focus is on business objects. Figure 6 illustrates the representation of the **Knowledge to Product** pattern (the knight icon with a dark-red background) and its related rules in the DOM. In the figure, the knight icons represent the strategies, the crystal balls represent the modules, the squares represent the objects, the magnifiers represent attributes and sub-attributes, and the blue balls represent attribute values. The DOM builds a tree (graph with no cycles) for each domain object (each business object of the case study) and places the object's name at the root node of each tree. DOM is thus a collection of trees, and the number of trees is equal to the number of objects in the domain. DOM is a special case of the decision tree diagram (Mentzer and Moon, 2004 [56], p. 166), with the distinction that it strictly follows an object-oriented hierarchy with a decreasing level of importance and detail, just as the mind map in the first stage. In the case study, the DOM was constructed manually, making updates difficult. For widespread applicability in industry, an integrated development environment (IDE) should be designed and developed to construct the DOM automatically, based on mind maps.

For the rule base in the case study, suggestions (profit patterns) are triggered through IF-THEN strategy rules conditioned on a single domain object at a time. Thus, the map consists of a collection of paths for each of the domain objects, and the suggestions reside directly at the leaf nodes. If the antecedent of the strategy rule was conditioned on several domain objects, this could be reflected by creating dummy nodes that have incoming arcs from the leaf nodes of each object's paths.

---

output **RuleBase Generate_RuleBase**(input G_3)

for each module

$\forall m \in M$

insert a new record to NODE table for each suggestion s

$\forall s \in S_m$

create r *where*

$r.Node\_ID=$ generateUniqueIDfor*(s)*

$r.Module\_No=$ m

$r.Node\_Type=$ "Suggestion"

$r.Text=$ returnTextForStrategy(s)

insert r into NODE

for each object in that module

$\forall o \in O_m$

for each question in that object O_m of module m

$\forall q \in Q_{om}$

insert a new record r to NODE table for that question q

create r where

$r.Node\_ID=$ generateUniqueIDfor(q)

$r.Module\_No= m$

$r.Node\_Type=$ "Question"

$r.Text=$ "What is the value for"+o+"."

+a_1+"."+ ... + "." +a_n+ "?"

insert r into NODE

for each value v, insert a new record to ANSWER_NODE table

$\forall v \in valueset(q)$

create r where

r.$Answer\_Node\_ID=$ generateUniqueIDfor(q, v)

$r.Link\_from\_Node\_ID=$ returnTheUniqueIDfor(q)

$r.Answer\_Order=$ generateNextValueFor(q, v)

$$r.Is\_Link\_to\_Suggestion= \begin{cases} 1, v \in V* \\ 0, o/w \end{cases}$$

$r.Answer\_Text= v^*$

insert r into ANSWER_NODE

link the question and answer nodes, through defining arcs

create r where

$r.Answer\_Node\_ID=$ returnTheUniqueIDfor(q, v)

$r.Link\_Order=$ generateNextValueFor(q, v)

$r.Link\_to\_Node\_ID=$ returnTheUniqueIDfor(q)

insert r into ARC

Fig. 4. Transformation algorithm that generates the Rule Base from the Rule Map.

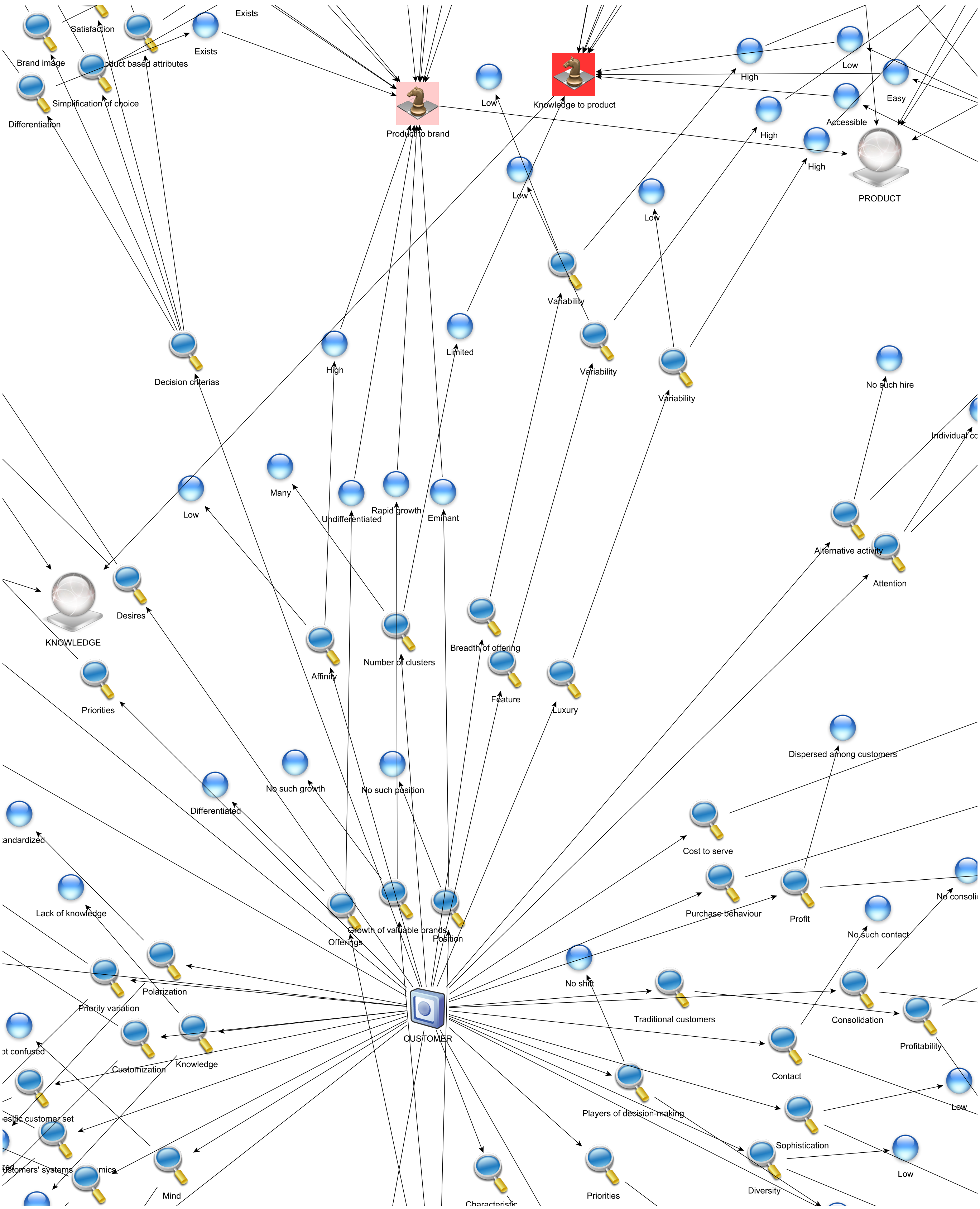

Fig. 6. The implementation of the domain objects map (Stage 2, DOM) in the case study.

Stage 3: Rule Maps

Stage 3 represents the rule base as rule map(s), which is a representation scheme that is readily used in industry. Figure 7 is constructed based on the logic representation in Visirule, a commercial software. In Step 3 of the case study, the reasoning for each

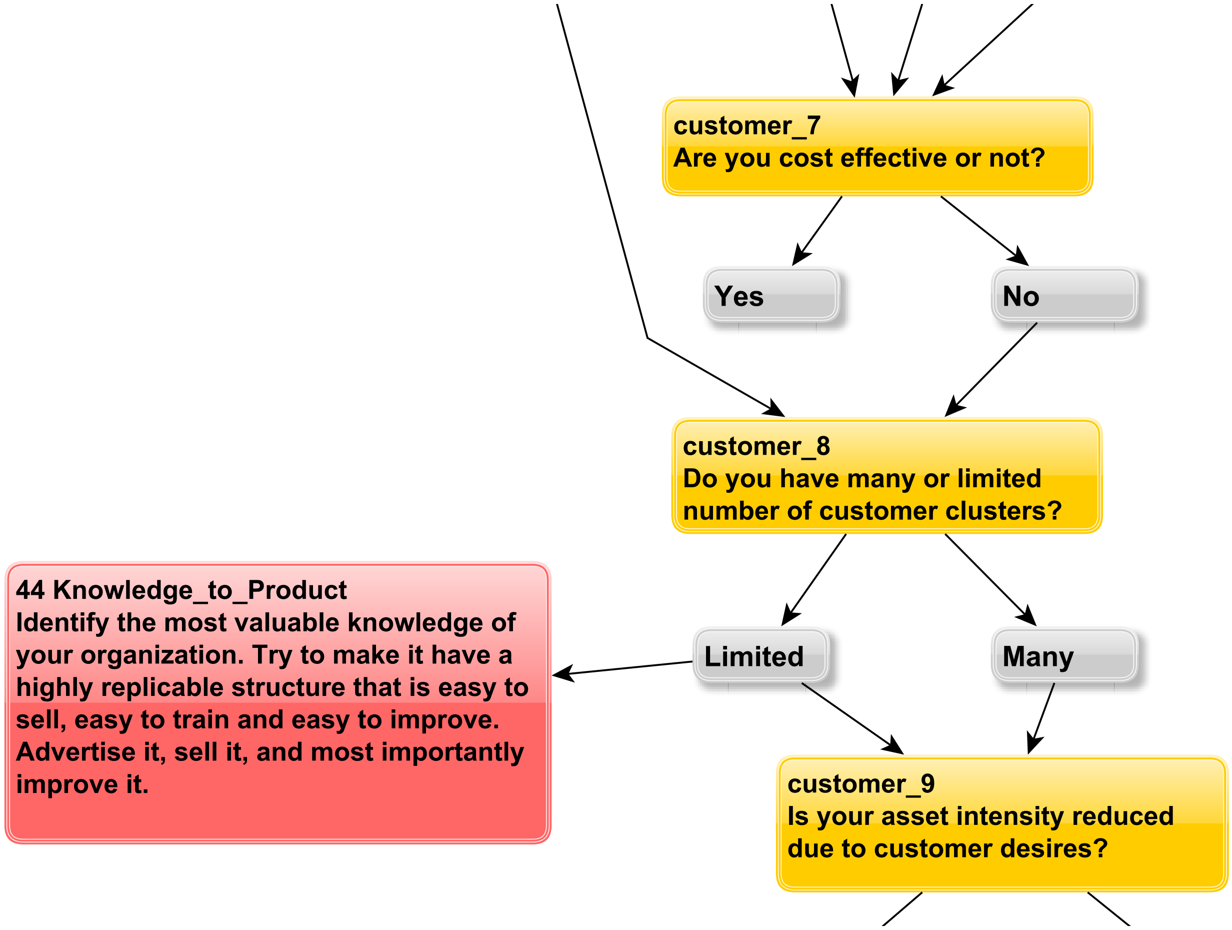

Fig. 7. The implementation of the rule map (Stage3) in the case study.

business function is reflected independently in a separate rule map as a module. A main menu is used to access each module.

In Figure 7, the yellow boxes labeled as **customer_7**, **customer_8** and **customer_9** are question boxes and the small white boxes labeled as **Yes**, **No**, **Limited**, and **Many** are the input boxes that show the possible answers to the questions. The profit pattern **44_Knowledge_to_Product** is shown as a red suggestion box. Even though many other arcs go into this suggestion box, they have been removed from the figure for clarity.

Stage 4: Software Implementation

6 Future Work

An important trend in data analytics is the automated extraction of information and its representation in knowledge graphs (Janev *et al.* (2020) [42]). The methodology, especially the first stage, can be enhanced with such an automated process, where the graph at that stage is constructed directly from the knowledge source.

With respect to application, one future work would be to integrate all the stages of the methodology in a single expert system modeling software, which enables the specification of the rules into the mind map, DOM, rule map, or an expert system knowledge base, automatically translating models between the different languages and stages.

Currently, each question has an equal weight in calculating the score for each profit pattern. Furthermore, the suggestions were shown with their scores. As an alternative, the expert system can be programmed to “fire” in more complicated situations or only above a certain threshold score. Another extension could be to integrate fuzzy reasoning (Pappis and Siettos, 2014 [65]) and multi-criteria decision-making (MCDM), for scoring and ranking suggestions.

Another dimension of future research is the analysis of the obtained data: if the electronic logs obtained from the users are pooled in a database, detailed analysis of the logs can yield significant insights into how managers and business professionals reason, what characteristics companies possess, and what needs companies, industries, and most importantly, people have. The survey data and software logs can also be used to map market environments in several industries and, therefore, can be used for benchmarking. Managers can assess the position of their firms and the overall situation of their industries through benchmarking. This knowledge can also be used for the education of managers in strategic decision-making.

With respect to implementation, a visible future work would be the automation of the proposed methodology as an integrated development environment (IDE) software. Such software should support seamless transition among the stages, from one format to another, and thus empower several project members with various preferences to use their own preferred interface. In the ideal case, the software should automatically generate the source code of the desktop or web applications and the executable of the desktop application. The development of such software can be facilitated by building on existing open-source software code, such as FreeMind for the mind map, Eclipse GMF Runtime[7] for the DOM and rule map, Drools[8] for the rule engine and rule map, and plug-ins for Eclipse, such as Telosys[9], for automated code generation.

Future work on the case study includes the extension of the knowledge base for the developed system, in the following ways:

- Recently emerging profit patterns, reported by Slywotzky in his more recent books can be included.
- The strategic know-how of other influential business thinkers, such as Michael Porter, Bill Gates, and C. K. Prahalad can be appended[10].
- New modules in other functions of strategic management, such as finance, supply chain management (SCM), and human resource management (HRM) can be established.

The use of StrategyAdvisor Cloud by SMEs as a web-based application −as opposed to a desktop application− would facilitate access to the body of knowledge and expertise built into the system, increasing the influence of the study (Verhodubs and Grundspenkis, 2011 [81]; Patel *et al.*, 2012 [66]). The essential benefits of Software as a Service (SaaS) include reduced costs in delivering the software and faster improvements in software because of instant access to the usage logs. The SaaS scheme is especially suitable for Small Medium Enterprises (SMEs) because of its minimal initial cost and transactional overhead.

7 Technology Selection for Developing StrategyAdvisor Cloud

Throughout the case study, mind maps (Stage 1) for the strategic management domain were modeled using open-source Freemind software. Computer support for mind mapping has several advantages (Kommers and Lanzing, 1997 [45]), including ease of adaptation and manipulation, ease of conversion to other electronic formats, convenience and speed in communication, dynamic linking of elements on changes, and immediate analysis of the structured knowledge (zooming, panning, automated layout). Bhattacharya and Mohalik (2020) [9] presented several advantages of digital mind maps, such as enabling easy understanding of information and furthering the engagement pattern of students by discussing concepts through visual representations.

In Stage 2, the domain objects map (DOM) was initially constructed as a mind map, and then transferred to yEd[11] software. An organic layout algorithm was used to visualize the mind map.

The rule map of Stage 3 was constructed using commercial LPA VisiRule[12] software. The software allows programming the rules visually as a flow chart and can generate Flex and Prolog codes that can be further edited and debugged, and can eventually be executed to provide decision support. Flex is a language developed and supported commercially, and Prolog is a well-known special-purpose language for expert system development. The Win-Prolog system is an integrated development environment (IDE) that allows the development and execution of expert systems using Prolog language, and is developed by the company that offers VisiRule. Alternative software, including the commercial VisualRules software[13] and the open-source Drools Flow software[14] support graphical modeling of rule logic, and any of these could have been used at this stage. However, none of the aforementioned systems implement the full multistage methodology proposed here because they do not have agent-oriented customization in all the stages.

In Stage 4, a cloud application, namely StrategyAdvisor Cloud[15], was developed using Javascript[16] scripting language and the Node.js[17] framework, with data being read from an online Google sheet[18]. The application was coded using the Visual Studio.NET IDE[19] (Integrated Development Environment), and deployed on the cloud on a Ubuntu[20] server. The application runs in Google Chrome and Microsoft Edge browsers, through the *http* protocol, and may require the cleaning of the cache in the browser to run. The continuous running of the cloud application is monitored and enabled through PM2 Process Manager[21]. The source code for the application is publicly available in the GitHub[22] repository.

8 Assessment of StrategyAdvisor Cloud

8.1 Success Factors

Multiple studies in the literature list success factors in the development of expert systems (ES), executive information systems (EIS), and information systems (IS). Notable studies include the work of Shibl *et al.* (2013) [76] on decision support systems,

7. https://projects.eclipse.org/projects/modeling.gmf-runtime
8. http://www.drools.org/
9. https://telosys.org/eclipsePlugin.html
10. http://www.thinkers50.com/
11. https://yworks.com
12. https://www.visirule.co.uk/
13. https://www.actico.com/visual-rules
14. https://drools.org
15. https://strategyadvisor.herokuapp.com/
16. https://javascript.com
17. https://nodejs.org
18. https://www.google.com/sheets/about/
19. https://visualstudio.microsoft.com/vs/features/net-development/
20. https://ubuntu.org/
21. https://pm2.io/
22. https://github.com/gurdalertek/strategyadvisor

Kamaruddin *et al.* (2011) [43], and Rahnavard and Gholami (2012) [71] on executive information systems, Petter *et al.* (2013) [67] on information systems, and most notably the work of Nurminen *et al.* (2003) [62] on expert system assessment. Other noteworthy studies regarding the evaluation of ES, EIS, and IS include the following: Munaiseche and Liando (2016) [59] illustrate how to assess the usability values of expert system applications, such as "learnability, efficiency, memorability, errors, and satisfaction". Urbach and Müller (2012) [80] described in detail the popular IS success model devised by Petter *et al.* (2013) [67]. Marx *et al.* (2012) [55] conducted a survey among executives for EIS redesign. From among the listed frameworks, as presented in Appendix 8, the comprehensive yet succinct framework of Nurminen *et al.* (2003) [62] is applied in the current research for the assessment of StrategyAdvisor Cloud, an expert system for strategic management developed based on the developed methodology.

8.2 Framework Selection

Nurminen *et al.* (2003) [62] list 20 criteria for assessing whether a developed expert system will be successful, with eventual usage. The developed StrategyAdvisor Cloud expert system is evaluated to determine how it conforms to these 20 criteria. Among the several studies that list success factors, this particular framework has been selected, because it

1) is one of the most comprehensive frameworks,
2) focuses explicitly on ES, rather than IS at large, and
3) focuses on long-term success, rather than short-term success, which is more important for strategic-level expert systems, such as StrategyAdvisor Cloud.

The 20 criteria listed in Nurminen *et al.* (2003) [62] are based on hypotheses proposed in expert systems literature and the paper itself. The authors' evaluation of the importance of criteria is based on experience with successful expert system applications designed and developed at Nokia Research Center, which have remained in industrial use for a long time.

8.3 Assessment

Table 1 lists the criteria of Nurminen *et al.* (2003), and whether StrategyAdvisor Cloud meets these criteria. The categories for the criteria are *Domain, Development, Operation* and *General*. The first column (column A) in Table 1 lists the specific criteria presented by Nurminen *et al.* (2003) [62]. The second column (column B) shows whether each criterion is significant. The first two columns are taken directly from the reference paper and can be used to judge the potential success of any expert system. The remaining two columns in the table (columns C and D) assess StrategyAdvisor Cloud with respect to the criteria.

Table 1 provides strong evidence for the potential success of StrategyAdvisor Cloud, because all the applicable criteria, except "High degree of automation," are satisfied by the system.

9 Expert Systems and SMEs

Strategic management is of utmost importance for any company, because the implications of strategic decisions propagate down to tactical and operational management. However, human expertise in this field is not abundant, and one-to-one interaction with a human expert on strategic management is too expensive to access for small-medium enterprises (SMEs). According to the European Union (EU), an SME is defined has having fewer than 250 employees and 50 million Euro revenue (EurActiv) [30]. In 2018, there were 25 million SMEs in EU-28 which accounted for "99.8% of all enterprises in the EU-28 non-financial business sector (NFBS), generating 56.4% of value-added and 66.6% of employment in the NFBS" (Europa [29]). In the United States, SMEs accounted for approximately 32% of the country's exporters in 2018, with their exports amounting to $473.3 billion (U.S. Census Bureau [79]). The developed StrategyAdvisor Cloud expert system supports strategic decision making and can have a positive impact both on both the resource-bounded SMEs at the micro scale and regional and national economies at the macro scale.

Data Availability

The domain objects map (DOM) knowledge representation of the rules is publicly available under https://ertekprojects.com/new-knowledge-in-strategic-management/data-yed-graphs/ as yEd graph files.

Acknowledgment

This research was based on Startup Grant Funding 31B127 of the United Arab Emirates University. The authors thank Ahmet Lekesiz for developing the StrategyAdvisor Cloud software and Riaz Uddin Bhuiyane for deploying the software on the cloud. The authors thank Ashraf Khalil, Matloub Hussain, Aysha Al-Kaabi, Damla Uygur, Ceylin Özcan, Özge Onur, Ceren Atay and Gizem Kökten, Dr. Gül Tokdemir, and Nihat Kasap for their suggestions to improve the paper and help with the literature search, Cem Kanpara for redrawing the DOM in yEd software, Kamil Çöllü and Soner Ulun for their support in editing, Richard Wilkinson for his help in proofreading, and Clive Spenser from Logic Programming Associates (creator and distributor of VisiRule and Win-Prolog software products) for his extensive help and support during the project.

References

[1] S. S. Abbas, A. Eldin, and A. Elsayed. "The effect of concept mapping and mind mapping utilization on students ′ understanding level: An empirical study". In: *Proc. of the Eighth Int. Conference on Concept Mapping*. 2018.

[2] S. Adodo. "Effect of Mind-Mapping as a Self-Regulated Learning Strategy on Students' Achievement in Basic Science and Technology". In: *Mediterranean Journal of Social Sciences* 4.6 (2013), p. 163.

[3] M. Ali et al. "A Data-Driven Knowledge Acquisition System: An End-to-End Knowledge Engineering Process for Generating Production Rules". In: *IEEE Access* 6 (2018), pp. 15587–15607.

[4] M. Almorsy and J. Grundy. "SecDSVL: A Domain-Specific Visual Language to Support Enterprise Security Modelling". In: *2014 23rd Australian Software Engineering Conference*. 2014, pp. 152–161.

[5] J. Ang and X. Tong. "Using the CARMEN's framework to model and acquire knowledge". In: *Expert Systems* 14.2 (1997), pp. 57–68.

[6] D. Arias-Aranda et al. "A fuzzy expert system for business management". In: *Expert Systems with Applications* 37.12 (2010), pp. 7570–7580.

[7] X. Bai, D. White, and D. Sundaram. "Contextual adaptive knowledge visualization environments". In: *The Electronic Journal of Knowledge Management* 10.1 (2012), pp. 1–14.

[8] Sadeer Beden, Qiushi Cao, and Arnold Beckmann. "SCRO: A Domain Ontology for Describing Steel Cold Rolling Processes towards Industry 4.0". In: *Information* 12.8 (2021), p. 304. DOI: 10.3390/info12080304.

[9] D. Bhattacharya and R. Mohalik. "Digital Mind Mapping Software: A New Horizon in the Modern Teaching-Learning Strategy". In: *Journal of Advances in Education and Philosophy (JAEP)* 4.10 (2020), pp. 400–406.

[10] P. Bottoni and A. Fish. "Extending Spider Diagrams for policy definition". In: *Journal of Visual Languages & Computing* 24.3 (2013), pp. 169–191.

[11] S. Bresciani and M. J. Eppler. "The Pitfalls of Visual Representations: A Review and Classification of Common Errors Made While Designing and Interpreting Visualizations". In: *SAGE Open* 5.4 (2015), p. 215824401561145.

[12] Michael Burch et al. "The State of the Art in Empirical User Evaluation of Graph Visualizations". In: *IEEE Access* 9 (2021), pp. 4173–4198. DOI: 10.1109/ACCESS.2020.3047616.

[13] T. Buzan and B. Buzan. *The Mind Map Book*. BBC Books, 1995.

[14] M. Chein and M.-L. Mugnier. *Graph-based knowledge representation: computational foundations of conceptual graphs*. London: Springer-Verlag London, 2009.

[15] Y.-J. Chen. "Development of a method for ontology-based empirical knowledge representation and reasoning". In: *Decision Support Systems* 50.1 (2010), pp. 1–20.

Table 1: Assessment of the StrategyAdvisor Cloud expert system based on the assessment criteria of Nurminen *et al. (2003)*

A	B	C	D
Category of Criterion and Criterion	**Evidence by Nurminen *et al.* (2003)**	**Satisfaction of the criterion by StrategyAdvisor Cloud**	**Comments**
Domain			
Narrow scope	Mixed	+	
Focused objective	No	+	
Stability of environment	No	+	
High degree of automation	Mixed	–	
High degree of repetition	Yes	+	StrategyAdvisor Cloud can be used by not only top managers but also by middle managers and professionals for learning strategic management and broadening their vision.
Small project	Yes	+	The expert system was developed by one system analyst (first author), one software specialist (third author for the desktop application and a contributor in Acknowledgements) and managed by a single supervisor (second author).
Development			
Users prefer usability over automation	Yes	+	Top and middle managers give importance to simplicity and ease of use, rather than automation. StrategyAdvisor Cloud is easy to learn and use.
Expert systems complement rather than replace human experts	Yes	+	Human consultants or specialists will eventually be required to adopt the actions required by the profit patterns and enable their execution at the client company.
Early benefits to experts themselves are important	Yes	N/A	
AI should be an embedded part of a bigger system	Yes	N/A	StrategyAdvisor Cloud is currently a simple application, yet can be enhanced with data collection and analytics functionality.
Simple, straightforward solutions work best	Yes	+	The focus of StrategyAdvisor Cloud is on bringing practical insights through a friendly interface.
If-then rules considered harmful	Yes	+	Even though IF-THEN rules are found to underrepresent the knowledge domain in technical fields, such as fault diagnosis, they are observed to work well with domains such as legal reasoning (Nurminen et al., 2003). We believe that reasoning for strategic management is closer to reasoning for legal cases compared with reasoning for technical engineering domains.
Lots of custom work	Yes	+	A new visual methodology was created for the development of an expert system for strategic management consulting.
Fast and agile development is important	Yes	+	The development of an earlier desktop application was completed in a timespan of 14 weeks, with an activity duration of approximately two weeks. The development of the cloud software took the same activity duration of two weeks.
Knowledge-based applications are based on domain specific application engines and generators	Yes	+	Custom code was developed for the StrategyAdvisor Cloud application that reads the rule base from a database.
Monolithic applications are sometimes better than knowledge-based applications	Yes	N/A	StrategyAdvisor Cloud is closer to a knowledge-based application, rather than a monolithic application, because there is a certain degree of separation between knowledge and application logic. However, for the user, the usage of the application is through a monolithic interface.
Rules of normal software development apply	Yes	+	In developing StrategyAdvisor Cloud, basic software development practices, such as modularization, separation of system design and software implementation, etc., have been followed.

Operation			
Move towards mainstream software and hardware platforms	Yes	+	StrategyAdvisor Cloud was implemented by using a mainstream programming languages and tools.
Successful systems tend to move out from the company	Some	+	StrategyAdvisor Cloud is freely accessible on the Internet for any business professional to benefit from.
General			
The difference between expert systems and information systems is small	Yes	+	From a manager's perspective, StrategyAdvisor Cloud is not much different from another web applications in terms of its use; it is straightforward to use with mainstream visual elements of web applications.

[16] W. Chergui, S. Zidat, and F. Marir. "An approach to the acquisition of tacit knowledge based on an ontological model". In: *Journal of King Saud University - Computer and Information Sciences* 32.7 (2020), pp. 818–828.

[17] Janneth Chicaiza and Priscila Valdiviezo-Diaz. "A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions". In: *Information* 12.6 (2021), p. 232. DOI: 10.3390/info12060232.

[18] José Marıa Rodrıguez Corral et al. "A Study on the Suitability of Visual Languages for Non-Expert Robot Programmers". In: *IEEE Access* 7 (2019), pp. 17535–17550. DOI: 10.1109/ACCESS.2019.2895913.

[19] A. Davidson and M. Tenaglia. "A directory of strategic management software tools". In: *Planning Review* 21.4 (1993), pp. 38–40.

[20] M. Davies. "Concept mapping, mind mapping and argument mapping: what are the differences and do they matter?" In: *Higher education* 62.3 (2011), pp. 279–301.

[21] C. Diamantini, D. Potena, and E. Storti. "SemPI: A semantic framework for the collaborative construction and maintenance of a shared dictionary of performance indicators". In: *Future Generation Computer Systems* 54 (2016), pp. 352–365.

[22] Z. Drey and C. Consel. "Taxonomy-driven prototyping of home automation applications: A novice-programmer visual language and its evaluation". In: *Journal of Visual Languages & Computing* 23.6 (2012), pp. 311–326.

[23] S. Edwards and N. Cooper. "Mind mapping as a teaching resource". In: *The Clinical Teacher* 7.4 (2010), pp. 236–239.

[24] H. Ehrig et al. *Graph and model transformation: General framework and applications*. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg, 2015.

[25] M. J. Eppler. "A Comparison between Concept Maps, Mind Maps, Conceptual Diagrams, and Visual Metaphors as Complementary Tools for Knowledge Construction and Sharing". In: *Information Visualization* 5.3 (2006), pp. 202–210.

[26] M. J. Eppler and S. Bresciani. "Visualization in management: From communication to collaboration. A response to Zhang". In: *Journal of Visual Languages & Computing* 24.2 (2013), pp. 146–149.

[27] M. J. Eppler and R. A. Burkhard. "Chapter 65: Knowledge visualization". In: *Knowledge management: concepts, methodologies, tools, and applications*. Ed. by M. E. Jennex. IGI Global, 2008, pp. 781–793.

[28] M. Erwig and E. Walkingshaw. "A visual language for explaining probabilistic reasoning". In: *Journal of Visual Languages & Computing* 24.2 (2013), pp. 88–109.

[29] European Commission. *Executive summary, 2018/2019 Annual report on European SMEs*. https://ec.europa.eu/growth/smes/sme-strategy/performance-review_en/. Accessed: 2021-02-28. 2019.

[30] European Commission. *User guide to the SME Definition*. https://ec.europa.eu/docsroom/documents/42921. Accessed: 2021-03-01. 2020.

[31] K. Gai et al. "Ontology-Based Knowledge Representation for Secure Self-Diagnosis in Patient-Centered Teleheath with Cloud Systems". In: *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*. 2015, pp. 98–103.

[32] Giovanni Garifo et al. "The Geranium Platform: A KG-Based System for Academic Publications". In: *Information* 12.9 (2021), p. 366. DOI: 10.3390/info12090366.

[33] T. Gavrilova. "Ontological Engineering for Practical Knowledge Work". In: *Knowledge-Based Intelligent Information and Engineering Systems*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, pp. 1154–1161.

[34] T. Gavrilova, I. Leshcheva, and E. Strakhovich. "Gestalt principles of creating learning business ontologies for knowledge codification". In: *Knowledge Management Research & Practice* 13.4 (2015), pp. 418–428.

[35] T. Gavrilova, I. Leshcheva, and E. Strakhovich. "Gestalt principles of creating learning business ontologies for knowledge codification". In: *Knowledge Management Research & Practice* 13.4 (2015), pp. 418–428.

[36] D. Grahovac and V. Devedzic. "COMEX: A cost management expert system". In: *Expert Systems with Applications* 37.12 (2010), pp. 7684–7695.

[37] L. Grunske, K. Winter, and N. Yatapanage. "Defining the abstract syntax of visual languages with advanced graph grammars −− A case study based on behavior trees". In: *Journal of Visual Languages & Computing* 19.3 (2008), pp. 343–379.

[38] I. Gupta and G. Nagpal. *Artificial Intelligence and Expert Systems*. Sterling, VA: Stylus Publishing, LLC, 2020.

[39] E. Harzheim. *Ordered Sets*. New York: Springer-Verlag, 2005.

[40] Martina Husáková and Vladimır Bureš. "Formal Ontologies in Information Systems Development: A Systematic Review". In: *Information* 11.2 (2020), p. 66. DOI: 10.3390/info11020066.

[41] İ. İrdesel et al. *Supplement for "A Graph-Based Methodology for Constructing Rule-Based Expert Systems"*. https://.... 2021.

[42] Valentina Janev et al. *Knowledge graphs and big data processing*. Springer Nature, 2020.

[43] M. Kamaruddin, R. Razali, and A. Deraman. "Critical success factors of executive information systems development for education management - A preliminary investigation". In: *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. 2011, pp. 1–6.

[44] J. Al-Kassab et al. "Information visualization to support management decisions". In: *International Journal of Information Technology & Decision Making* 13.2 (2014), pp. 407–428.

[45] P. Kommers and J. Lanzing. "Students' conceptmapping for hypermedia design: Navigation through worldwide web (www) space and self-assessment". In: *Journal of Interactive Learning Research* 8.3–4 (1997), pp. 421–455.

[46] R. Laue and A. Awad. "Visual suggestions for improvements in business process diagrams". In: *Journal of Visual Languages& Computing* 22.5 (2011), pp. 385–399.

[47] E. Leblebici et al. "Multi-amalgamated triple graph grammars: Formal foundation and application to visual language translation". In: *Journal of Visual Languages & Computing* 42 (2017), pp. 99–121.

[48] V. Lendel and M. Varmus. "The expert system as a proposal for creating innovative strategy". In: *Journal of Competitiveness* 2.2 (2010).

[49] S. Li et al. "Integrating group Delphi, fuzzy logic and expert systems for marketing strategy development: the hybridisation and its effectiveness". In: *Marketing Intelligence & Planning* 20.5 (2002), pp. 273–284.

[50] D.-R. Liu, C.-W. Lin, and H.-F. Chen. "Discovering role-based virtual knowledge flows for organizational knowledge support". In: *Decision Support Systems* 55.1 (2013), pp. 12–30.

[51] J. Liu et al. "A domain-specific visual modeling language for testing environment emulation". In: *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 2016, pp. 143–151.

[52] Qingqi Long, Ke Song, and Shuiqing Yang. "Semantic Modeling for the Knowledge Framework of Computational Experiments and Decision Making for Supply Chain Networks". In: *IEEE Access* 7 (2019), pp. 46363–46375. DOI: 10.1109/ACCESS.2019.2908979.

[53] J. L. López-Cuadrado et al. "SABUMO: Towards a collaborative and semantic framework for knowledge sharing". In: *Expert Systems with Applications* 39.10 (2012), pp. 8671–8680.

[54] E. J. Marchiori et al. "A visual language for the creation of narrative educational games". In: *Journal of Visual Languages & Computing* 22.6 (2011), pp. 443–452.

[55] F. Marx, J. H. Mayer, and R. Winter. "Six Principles for Redesigning Executive Information Systems −− Findings of a Survey and Evaluation of a Prototype". In: *ACM Transactions on Management Information Systems* 2.4 (2012).

[56] J. T. Mentzer Jr and M. A. Moon. *Sales Forecasting Management: A Demand Management Approach*. USA: SAGE Publications, Inc., 2004.

[57] K. Metaxiotis and J. Psarras. "Expert systems in business: applications and future directions for the operations researcher". In: *Industrial Management & Data Systems* 103.5 (2003), pp. 361–368.

[58] M. Min. "A Rule Based Expert System for Analysis of Mobile Sales Data on Fashion Market". In: *2013 International Conference on Information Science and Applications (ICISA)*. Los Alamitos, CA, USA: IEEE Computer Society, 2013, pp. 1–2.

[59] C. P. C. Munaiseche and O. E. S. Liando. "Evaluation of expert system application based on usability aspects". In: *IOP Conference Series: Materials Science and Engineering*. 2016, p. 012001.

[60] G. J. Nalepa. *Modeling with Rules Using Semantic Knowledge Engineering*. Cham: Springer International Publishing, 2018.

[61] J. D. Novak and D. B. Gowin. *Learning how to learn*. USA: Cambridge University Press, 1984.

[62] J. K. Nurminen, O. Karonen, and K. Hätönen. "What makes expert systems survive over 10 years - empirical evaluation of several engineering applications". In: *Expert Systems with Applications* 24.2 (2003), pp. 199–211.

[63] J. Oh, J. Yang, and S. Lee. "Managing uncertainty to improve decision-making in NPD portfolio management with a fuzzy expert system". In: *Expert Systems with Applications* 39.10 (2012), pp. 9868–9885.

[64] J. Ohlsson. "Enforcing Business Rules in E-Business Systems: A Survey of Business Rule Engines". MA thesis. Skövde, Sweden: Institutionen för kommunikation och information, University of Skövde, 2006.

[65] C. P. Pappis and C. I. Siettos. "Fuzzy Reasoning". In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. MA, US: Springer, 2014, pp. 519–556.

[66] M. Patel, P. Virparia, and D. Patel. "Web based fuzzy expert system and its applications − a survey". In: *International Journal of Applied Information Systems* 1.7 (2012), pp. 11–15.

[67] S. Petter, W. DeLone, and E. R. McLean. "Information Systems Success: The Quest for the Independent Variables". In: *Journal of Management Information Systems* 29.4 (2013), pp. 7–62.

[68] J. S. di Piazza. "Logic aids for interviewing experts". In: *Expert Systems* 8.3 (1991), pp. 183–191.

[69] E. Portmann, P. Kaltenrieder, and W. Pedrycz. "Knowledge Representation through Graphs". In: *Procedia Computer Science* 62 (2015), pp. 245–248.

[70] Md. S. Rahman. *Basic Graph Theory*. Cham: Springer International Publishing, 2017.

[71] F. Rahnavard and M. Gholami. "Critical Success Factors in Implementation of Executive Information Systems: A Case Study". In: *Iranian Journal of Information Processing and Management* 27.3 (2012), pp. 667–680.

[72] M. Ramaswamy, S. Sarkar, and Ye-Sho Chen. "Using directed hypergraphs to verify rule-based expert systems". In: *IEEE Trans. Knowl. Data Eng.* 9.2 (1997), pp. 221–237.

[73] M. Relich and P. Pawlewski. "A case-based reasoning approach to cost estimation of new product development". In: *Neurocomputing* 272 (2018), pp. 40–45.

[74] J. E. Rivera, F. Durán, and A. Vallecillo. "On the Behavioral Semantics of Real-Time Domain Specific Visual Languages". In: *Rewriting Logic and Its Applications*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 174–190.

[75] O. Serrat. "Drawing mind maps". In: *Knowledge Solutions*. Singapore: Springer, 2017, pp. 605–607.

[76] R. Shibl, M. Lawley, and J. Debuse. "Factors influencing decision support system acceptance". In: *Decision Support Systems* 54.2 (2013), pp. 953–961.

[77] J. R. Siddiqui et al. "Towards visual mapping in industrial environments - a heterogeneous task-specific and saliency driven approach". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 5766–5773.

[78] E. K. Tekez. "Development of an expert system for demand management process". In: *International Journal of Computer Integrated Manufacturing* 31.10 (2018), pp. 970–977.

[79] U.S. Census Bureau. *Preliminary Profile of U.S. Exporting Companies, 2019*. https://www.census.gov/foreign-trade/Press-Release/edb/2019/2019prelimprofile.pdf. Accessed: 2021-02-27. 2020.

[80] N. Urbach and B. Müller. "The Updated DeLone and McLean Model of Information Systems Success". In: *Information Systems Theory. Integrated Series in Information Systems*. New York, NY: Springer, 2012.

[81] O. Verhodubs and J. Grundspenkis. "Towards the Semantic Web Expert System". In: *Applied Computer Systems* 44.1 (2011), pp. 116–123.

[82] W. P. Wagner, M. K. Najdawi, and Q. B. Chung. "Selection of knowledge acquisition techniques based upon the problem domain characteristics of production and operations management expert systems". In: *Expert Systems* 18.2 (2001), pp. 76–87.

[83] David Weintrop and Uri Wilensky. "Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms". In: *Computers & Education* 142 (2019), p. 103646. ISSN: 0360-1315. DOI: 10.1016/j.compedu.2019.103646.

[84] Shao-Fang Wen and Basel Katt. "Managing software security knowledge in context: an ontology based approach". In: *Information* 10.6 (2019), p. 216. DOI: 10.3390/info10060216.

[85] M. Wertheimer. *Max Wertheimer Productive Thinking*. Birkhäuser Basel: Springer International Publishing, 2020.

[86] J. Wheeldon. "Is a Picture Worth a Thousand Words? Using Mind Maps to Facilitate Participant Recall in Qualitative Research". In: *Qualitative Report* 16.2 (2011), pp. 509–522.

[87] J. Whyte, K. Tryggestad, and A. Comi. "Visualizing practices in project-based design: tracing connections through cascades of visual representations". In: *Engineering Project Organization Journal* 6.2-4 (2016), pp. 115–128.

[88] C.-H. Wu, S.-C. Kao, and K. Srihari. "GDKAT: A goal-driven knowledge acquisition tool for knowledge base development". In: *Expert Systems* 17.2 (2000), pp. 90–105.

[89] Feixiang Xu, Xinhui Liu, and Chen Zhou. "Developing an ontology-based rollover monitoring and decision support system for engineering vehicles". In: *Information* 9.5 (2018), p. 112. DOI: 10.3390/info9050112.

[90] X. Zhao et al. “Case-based reasoning approach for supporting building green retrofit decisions”. In: *Building and Environment* 160 (2019), p. 106210.

[91] G. Zhu and C. A. Iglesias. “Computing Semantic Similarity of Concepts in Knowledge Graphs”. In: *IEEE Trans. Knowl. Data Eng.* 29.1 (2017), pp. 72–85.