Meta Model on Concept Cell

Jeffrey Zheng and Chris Zheng

Abstract Applying network topology schemes, two types of three levels of meta knowledge representations have been established. This chapter proposes a meta model on concept cell that provides a meta organisation of knowledge in natural and artificial intelligent systems structurally.

Keywords Knowledge model · Meta representation · Three levels of concept lattice · Description · Procedure · Core organisation

1 Introduction

A meta model on concept cell is outlined to represent knowledge in knowledge systems (KSs). This model has novel features that are of considerable interest for knowledge representation (KR).

Polanyi proposed a knowledge model in the 1940s. Knowledge is composed of two categories: tacit and explicit [1, 2]. In the 1970s, Anderson from a cognitive psychology identified knowledge with another two categories: declarative and procedural [3–5]. In the early 1990s, a procedural model was proposed by Nonaka who identified four transformations: tacit → tacit (socialisation), tacit → explicit (externalisation), explicit → explicit (combination) and explicit tacit (internalisation) [6, 7]. In 2000, a model was proposed by Nickols to arrange four classes (tacit, explicit, procedural and declarative) into three categories: tacit, explicit and implicit. In my opinion, the Nickols model is unsatisfactory for three reasons:

This work was supported by Australian Commercialising Emerging Technologies, (COMET) program.

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

C. Zheng

Tahto, Sydney, Australia
e-mail: z@caudate.me

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_9

- (i) it is a triangle of categories without a fixed order,
- (ii) there is uncertainty in implicit category and
- (iii) there is no structural correspondence to other KR methodologies.

To improve the first two weaknesses of Nickols approach, an executable knowledge model was proposed. A triplet (tacit, implicit and explicit) is constructed as a procedural structure. Implicit in it is the middle node linked with two other nodes in four transformations: tacit $\rightarrow$ implicit (externalisation), implicit $\rightarrow$ explicit (retrieval), explicit $\rightarrow$ implicit (category) and implicit $\rightarrow$ tacit (internalisation). In addition, the model provides distinguishable foreground/background and human/machine knowledge interfaces [8].

To explore different KS applications from philosophy, logic and digital libraries, to gene, chemistry, software and system engineering [9–11], people arrange common concepts to construct ontology libraries and procedures as core structures [12, 13]. Advanced system modelling tools such as ARIS [14], CIMOSA [15] and IDEF [16] provide function, data and process models and ontology description capture methodologies for constructing modern intelligent knowledge systems [17]. Because many contradictions, confusions, difficulties and unclear properties exist in KR foundation levels [13, 18, 19], consistently categorising practical knowledge into tacit/explicit and procedural/declarative is extremely hard for researchers, scientists, philosophers, psychologists and knowledge workers [14–17, 20, 21].

Practical computer-aided modelling systems use pragmatic approaches to manipulate simple structures (list, tree, stack, class and component) in real applications [14–17, 21]. Usually, declarative concepts seem easier to capture than procedural concepts. Based on this, many people believe that declarative knowledge is explicit and procedural knowledge is tacit [16, 17, 22]. A radical extension of a knowledge model in KR is proposed in a concept cell that arranges knowledge in KS for natural and artificial organisation. This model can fully support the above-mentioned knowledge models to consistently identify four categories of knowledge: tacit, explicit, declarative and procedural. The model also provides a core ontology to distinguish a hierarchy of structures within the core of a concept. According to convention, the word concept is used as an equivalent to knowledge in this chapter.

2 Concept Cell Model

Let K denote a cell of concepts (a concept cell) that is composed of three parts: M membrane, N nuclei and G gel. M is a frame that provides a container to hold both N and G . G is a base description of the content and N establishes a foundation of the cell. M inputs provide external concepts (externals) for N from deeper levels, and then output current content to other upper level cells. N is composed of two components: D declarative nucleus and P procedural nucleus. To illustrate this organisation, a cell $K = M, N, G$ is shown in Fig. 1.

Fig. 1 A concept cell K. **a** A slice, **b** hierarchy

For the convenience of construction, a special lattice is employed [23]. Only directed graphs are used similar to the most popular signal flow graphs [24] to analyse and syntheses process control [9], computer architecture [10], electric circuits [25], network topology [26–28] and dynamic systems [25, 29]. However, no lattices allow containing a loop and all lattices are composed of directed acyclic graphs [26, 28]. In a lattice, a node represents a cell and lattice links are determined by dependencies among nodes. Because the most complex part of a cell is its nuclei structures, detailed interior organisation is necessary to explore meanings of knowledge. To simplify, a simple cell (or a cell, if there is no confusion) is studied here, where nuclei of the cell are composed of only one declarative lattice and one procedural lattice.

Using lattice language, a cell K is described in Fig. 2. Different graphic symbols represent distinct forms of concepts as nodes. A rounded rectangle represents a general node; an octagon is a specific node; a rectangle shows a declarative node and an oval corresponds to a procedural node. A simple lattice cell is composed of

Fig. 2 A concept cell in lattices

four levels: A node M that interfaces between externals and internals are the first level. Two nodes of G and N link with an M node is the second level. The node G contains the base description and the node N plays a foundation role in the cell. Two nodes of D and P link with node N on the third level. Node D contains one lattice in declarative dependency and node P contains one lattice that assumes procedural dependency. Finally, two sets of nodes linked with nodes D and P at the fourth level. Each node of D or P contains four nodes, respectively. Among each four nodes, two links are associated with three nodes.

3 Core Components

The following four conditions can create the content of a concept cell:

- (i) M acts as an interface to import a finite number of externals into nuclei and to export the content to other cells.
 - (ii) G provides the base description of the cell and N collects all externals from M for development.
 - (iii) Two lattices D, P are constructed from Ns externals to carry out two dependen-
- cies.

An N external corresponds to a D node. A declarative dependency is employed to order all nodes of D as a declarative lattice. If two distinct nodes have declarative dependency, then the node with more general meanings is located at the first node and a declarative link connects from the first to the second. After building up declarative dependency among all nodes, D becomes a directed acyclic lattice.

Instances of an N external correspond to nodes of P satisfying procedural dependency. P is composed of sequences of nodes by instances of externals. If two instances represent two nodes, then the node that has to be handled earlier is specified as the first node and a procedural link connects two nodes from the first to the second. After all procedural dependencies are established among nodes, P is converted into a directed acyclic lattice.

(iv) Two lattices are composed of eight distinguishable node sets:

Four sets of declarative nodes C, T, I, E are identified: C core, T tacit, I implicit and E explicit, respectively.

Four sets of procedural nodes L, S, O, F are identified: L life cycle, S start, O operation and F finish.

The meanings of the construction process can be explained as: In the first level of kernel, M collects all externals to provide extra knowledge for its nuclei. The second level has two parts: G, N. The G node provides the base description. To map each external as a node, the number of N externals has the same number of nodes in D. A declarative dependency is valid for all D nodes that create a directed acyclic declarative lattice. Using instances of N externals as nodes, P has been assembled using procedural dependency linked with selected nodes and finally to form P itself as procedural lattice. Since both declarative and procedural lattices are organised by ordered dependencies, declarative and procedural lattices are directed acyclic to support wider requirements from theoretical foundations to practical applications. A simple construction example is shown in Fig. 3(i–v).

For an acyclic lattice, four distinct node sets are notable in Fig. 3(vi). They are (singleton, source, branch and sink) node sets, respectively, borrowed from network topology [23, 26, 30]. A singleton node provides an isolated concept. A source node exports a concept. A sink node imports concept(s) and a branch node transfers concept(s) from input link(s) to output link(s). If there is only one external in N, then the singleton set contains one single node and the other three sets are empty. If there is more than one node in N, then the singleton set is empty. In this case, the source set is composed of nodes that have at least one link to another node; however, a source node does not have a link from other nodes. Each node must have at least one in branch, or sink set consequently. In contrast to the source set, a sink set collects all nodes with links from other nodes, without a link to a node. A sink node has to be the last node in a node path of a lattice to which at least one node is linked, from branch or source set. Unlike source and sink sets, a node in a branch set may link with at

Fig. 3 External concepts, declarative and procedural lattices and node sets

least two nodes to and from source, sink, and branch sets. A branch node receives from other node(s) and outputs to other node(s). These sequence nodes provide connectivity among nodes. Although four node sets can be identified by their different connectivity, it is not convenient to use the same vocabulary to describe two distinct lattices under different dependencies. For convenience, each node set includes a proper name to indicate its specific relationship in familiar KR terms. D lattice represents an invariant structure (the simplest cases: tree, list) similar to a traditional data structure hierarchy. Because a sink node is equivalent to a factor data at the leaf level (at the lowest location) of data structure, the sink node has to be represented as an explicit knowledge. Therefore, the sink set of D is explicit. In contrast, a source node provides invaluable knowledge from the highest level of externals. There is

no link to this node and anyone wanting to explain the meaning of the node must capture knowledge from other sources far beyond the node itself. Consequently, a source node always contains deeper meanings than those can be articulated. Hence, the source set of D is tacit. Different from sink and source sets, a node in a branch set has connectivity from higher tacit node(s) and to higher explicit node(s). The branch set of D represents a typical intermediate property. Consequently, the branch set of D is implicit. A singleton node provides a complete concept. The node itself is the central of the D lattice. Therefore, the singleton node set of the D lattice is a core. Four node sets of P lattice satisfy different properties. The P lattice has a close relationship to process modelling that provides a time arrow as controllable sequences. A node in the P lattice is an instance of a node in the D lattice. The singleton node set of the P lattice is not empty if only one node is in the P lattice. The singleton node set of procedural lattice represents a complete procedure of P itself. Logically, the procedural singleton node set is a life cycle. When two or more nodes are included, three node sets of the P lattice have to link together in sequential relationships. Time relevant sequences in finite numbers of connected nodes, must have distinguishable commence and end nodes that correspond to start and finish conditions respectively. In addition, all intermediate nodes provide operational capacities to deliver knowledge to consequent nodes. Consequently, three node sets of the P lattice are called: start, finish and operation, respectively. The relative properties of the cell model with other schemes are compared in Table 1. In the table, TM represents Theoretical Model that is used in KS applications. ST denotes Structural Theory that uses structured organisations to represent complex dependency among members. ES indicates Engineering Systems that provide mixed theories, experiences and skills with commercial system modelling tools for pragmatic applications especially in enterprise management, manufacturing and building industries, software and hardware systems, global communication networks, web and Internet environment. ES applies advanced TM methodologies plus business experiences and engineering kills to solve practical problems efficiently using system engineering methodologies in global business explorations.

From this comparison, it is clear that existing systems that are the most similar to the cell concept model come from enterprise modelling that provides all functionality for ten meta nodes from engineering practices. However, other theoretical models cannot support full functionality. This property indicates the potential capacity for applying the cell concept model from theoretic foundations to practical applications. Details of the concept cell have published [31] to represent further classifications, recursive constructions, non-simple cells and sample applications for knowledge construction systems.

Table 1 Comparisons on different models

Model	D	T	I	E	C	P	S	O	F	L	Notes
Concept Cell	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	A hierarchy of four levels
Polanyi[1,2]		TM		TM							Tacit and Explicit
Anderson[3-5]	TM					TM					Declarative and Procedural
Nonaka[6,7]							ST		ST		Four Transformations
Nickols[8]		TM	TM	TM							Tacit, Explicit and Implicit
Zheng et al.[9]							ST	ST	ST		Four Transformations
Lattice Theory[17,28]		TM	TM	TM			TM	TM	TM		Theoretical Model
Ontology Metalogic[13-15]	TM	TM	TM	TM	TM						Logic, Metaphysics, ...
Conceptual graphs[15]						TM	TM	TM	TM	TM	Logic reasoning in graphs
/First order logic[15]						ES	ES	ES	ES	ES	/Symbol notations
Enterprise Modelling [11-14,27]	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES	ARIS, IDEF ...
Object Oriented[13,27]	ES				ES	ES	ES	ES	ES	ES	OTM, UML, C++, Java
Function/.../Logic[13,27]						ES	ES	ES	ES	ES	Algol, Fortran, Lisp, Prolog

Ten basic symbols: {D, T, I, E, C}, { P, S, O, F, L}

D: Declarative; T: Tacit, I: Implicit, E: Explicit, C: Core;

P: Procedural; S: Start, O: Operation, F: Finish, L: Life cycle

Three types of models: {ST, TM, ES}

ST: Structural theory

TM: Theoretical model

ES: Engineering system

References

1. M. Polanyi, *Knowing and Being* (The University of Chicago Press, Chicago, 1969)

2. M. Polanyi, *Tacit knowledge (the tacit dimension)*, in *Knowledge in Organizations* (Butterworth-Heinemann, Boston, 1997), pp. 135–146

3. J.R. Anderson, *Language, Memory and Thought* (Erlbaum, Hillsdale, 1976)

4. J.R. Anderson, *Rules of the Mind* (Erlbaum, Hillsdale, 1993)

5. J.R. Anderson, *Cognitive Psychology and Its Implications* (W.H. Freeman and Company, New York, 1995)

6. I. Nonaka, *The Knowledge Creating Company*, Harvard Business Review (November–December 1991), pp. 96–104

7. I. Nonaka, *The Knowledge Creating Company* (Oxford University Press, 1995)

8. J.Z.J. Zheng, M. Zhou, J. Mo, A. Tharumarajah, Background and foreground knowledge in knowledge management, in *Global Engineering, Manufacturing and Enterprise Networks* (Kluwer Academic Publisher, 2001), pp. 332–339

9. K. Hartmann, K. Kaplick, *Analysis and Synthesis of Chemical Process Systems* (Elsevier, Amsterdam, 1990)

10. D. Agrawal, *Advanced Computer Architecture* (IEEE Computer Society Press, Los Alamitos, 1986)

11. J. Cuena, *Knowledge Oriented Software Design* (North-Holland, Amsterdam, 1993)

12. A.P. Sage, W.B. Rouse, *Handbook of Systems Engineering and Management* (Wiley, New York, 1999)

13. W. Ziarko (ed.), *Rough Sets* (Fuzzy Sets and Knowledge Discovery (Springer, Berlin, 1994)

14. A.W. Scheer, *Architecture of Integrated Information Systems: Foundation of Enterprise Modelling* (Springer, 1992)

15. P. Bernus, L. Nemes, T.J. Williams, *Architecture for Enterprise Integration* (Chapman & Hall, New York, 1996)
16. IDEF Family of Methods A Structured Approach to Enterprise Modeling and Analysis (IDEF0 5). <http://www.idef.com>
17. P. Bernus, K. Mertins, G. Schmidt (eds.), *Handbook on Architectures of Information Systems* (Springer, New York, 1998)
18. P. Hjek, T. Havrnek, R. Jirouek, *Uncertain Information Processing in Expert Systems* (CRC Press, Boca Raton, 1992)
19. I. Graham, P. Jones, *Expert Systems-Knowledge, Uncertainty and Decision* (Chapman and Hall, London, 1988)
20. T. Davenport, L. Prusak, *Working Knowledge* (Harvard Business School Press, Boston, 1998)
21. J. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations* (Brooks Cole Publishing, Pacific Grove, 2000)
22. F. Nickols, *Knowledge in Knowledge Management, Knowledge Management Yearbook* (Butterworth-Heinemann, 2000), pp. 12–21. http://home.att.net/~nickols/Knowledge_in_KM.htm
23. G. Birkhoff, *Lattice Theory* (Providence, 1967)
24. J.R. Abrahams, G.P. Coverley, *Signal Flow Analysis* (Pergamon Press, 1965)
25. G. Lago, L.M. Benningfield, *Circuit and System Theory* (Wiley, New York, 1979)
26. S. Chan, *Introductory Topological Analysis of Electrical Networks* (Holt, Rinehart and Winston Inc, 1969)
27. R. Clay, *Nonlinear Networks and Systems* (Wiley, New York, 1971)
28. W. Chen, *Linear Networks and Systems* (Brooks/Cole Engineering Division, New York, 1983)
29. W.E. Lewis, D.G. Pryce, *The Application of Matrix Theory to Electrical Engineering* (E&FN Spon, London, 1965)
30. H. Hopf, *Differential Geometry in the Large* (Springer, Berlin, 1983)
31. J. Zheng, C. Zheng, T.L. Kunii, *Int. J. Inf. Acquisition* **01**, 149 (2004). <https://doi.org/10.1142/S021987890400015X>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.