From enterprise concepts to formal concepts

POLOVINA, Simon <http://orcid.org/0000-0003-2961-6207>

Published version

POLOVINA, Simon (2017). From enterprise concepts to formal concepts. In: The IJCAI-17 Workshop on Graph Structures for Knowledge Representation and Reasoning (GKR 2017 @ IJCAI), Melbourne, Australia, 19-25th August.

Copyright and re-use policy

From Enterprise Concepts to Formal Concepts

Simon Polovina

Conceptual Structures Research Group, Communication and Computing Research Centre | Department of Computing, Sheffield Hallam University, Sheffield, UK
s.polovina@shu.ac.uk

Abstract. A business enterprise is more than its buildings, equipment or financial statements. Enterprise Architecture frameworks thus include a metamodel that attempts to bring together all the enterprise concepts including the visible entities into a unified conceptual structure. Using a simple case study based on the author's institution, the effectiveness of this conceptual structure is explored. The metamodel is stated in Conceptual Graphs then mapped from these graphs' triples into transitive Formal Concept binaries using the *CGFCA* software. A misalignment in the enterprise concepts is discovered from the derived formal concepts, pointing towards the wider applicability of this approach.

1 Introduction

A business enterprise is more than just the sum of its buildings, equipment or financial statements. Such visible entities are simply the structures that follow from its strategy, which is just as real. Strategy is moreover the driving entity, without which the enterprise falters. Like many other disciplines, business modelling practitioners (such as enterprise architects) rely on useful conceptual models that underpin enterprise activity. The underlying enterprise concepts in these models capture the purpose of the enterprise (why it exists) and articulated through its strategy. To achieve its strategic goals, the enterprise concepts extend into the enterprise's lower level tactical and operational goals that include its locations, finance, assets (e.g. buildings, trading stock, information technology), staff and an organisational structure. History however continues to show these entities becoming the drivers resulting in the emergence of bureaucratic structures, inter-departmental conflicts, inadequate computer systems and other experiences where we have 'The tail wagging the dog' i.e. strategy is lost and ends up following structure [2]. Put another way, the operational enterprise concepts overtake the strategic enterprise concepts when it should be the other way round.

To address this phenomenon the paper is structured as follows. Enterprise concepts are introduced and discussed through the notion of enterprise architecture and the formal depiction of the enterprise concepts through ontology, semantics and metamodels. The relevance and use of Conceptual Structures is then addressed, including the simple case study that is a simplified representation of the author's institution, Sheffield Hallam University (SHU). This section also explicates Conceptual Graphs, Formal Concept Analysis and the *CGFCA* software and how they are used. Next is how the Conceptual Graphs and generated Formal Concept Lattices are iterated to correct the model and metamodel for the simplified SHU case study, and the value that formal concepts bring to enterprise concepts. This is followed by a discussion of the further significance of this work, culminating in the paper's conclusions.

2 Enterprise Architecture

Enterprise Architecture (EA) recognises that enterprises could best be understood by a holistic approach that explicitly refers to every important issue from every important perspective [17]. Hence all the enterprise concepts need to relate to each other.

2.1 Ontology, Semantics and Metamodels

EA arose from Zachman's original Information Systems Architecture Framework [17, 12]. Zachman's EA framework places the enterprise concepts in cells that are interrelated through a simple two-dimensional matrix, consequently referred as an enterprise ontology [14, 16]. The Open Group Architecture Framework (TOGAF) articulates the semantics in such an ontology by formally defining the relations between the enterprise concepts (entities) in a content metamodel rather than simply relying on their position in a matrix (or table) like Zachman [5, 4]. A metamodel is the model about the model. The TOGAF metamodel formally describes the model to which every enterprise conforms, thereby embodying enterprise concepts. In EA, metamodels are a common concept; TOGAF's being one of the most well-known.

3 Conceptual Structures

In his seminal text, Sowa describes Conceptual Structures (CS) as "Information Processing in Mind and Machine" [15]. Enterprises essentially arise as acts of human creativity in identifying business opportunities or other organisational solutions to social needs (e.g. government bodies, charities, schools or universities to name a few). Formal depictions of the metamodels (and the models that they in turn represent) enable them to be computable. Software tools potentially bring the productivity of computers to bear on interpreting the enterprise concepts, offering more expressive knowledge-bases leading to better decision-making. CS brings human creativity and computer productivity into the same mindset; CS thus offers an attractive proposition for capturing, interrelating and reasoning with enterprise concepts.

3.1 A Simplified Case Study

To clarify the approach, and explore the value of CS to enterprise concepts, a simple case study is now presented. For ease of understanding a much-simplified metamodel is used as well as a simplified description of the case study, which is Sheffield Hallam University (SHU) where the author of this paper is employed. SHU is a large public university located in Sheffield in the UK. Remembering that the term enterprise does not only apply to profit-making businesses, SHU's strategy is epitomised by the term 'Transforming Lives'. SHU meets this strategy through its location in Sheffield and the staff it employs (noting that these aspects are chosen from all its visible entities for simplicity.) The success of its strategy as realised through its staff and location (in this simplified example) is measured by Key Performance Indicators (KPIs). One such KPI in the UK is the National Student Survey ('the NSS', `www.thestudentsurvey.com`).

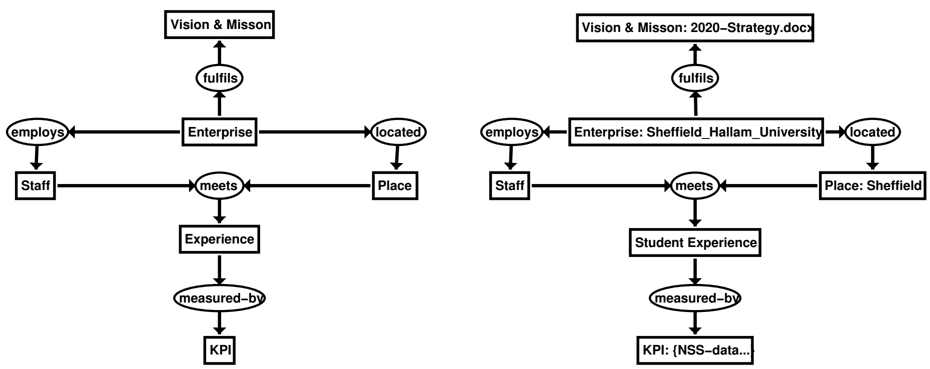


Fig. 1: Metamodel and SHU, in CGs

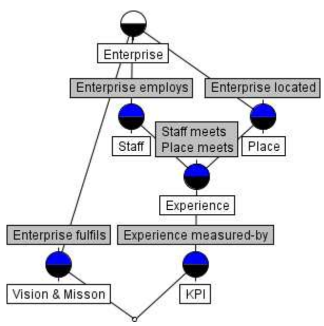
3.2 Conceptual Graphs

To demonstrate CS, Sowa devised Conceptual Graphs (CGs). Essentially, CGs are a system of logic that express meaning in a form that is logically precise, humanly readable, and computationally tractable. CGs serve as an intermediate language for translating between computer-oriented formalisms and natural languages. CGs graphical representation serve as a readable, but formal design and specification language [13, 7]. Fig. 1 reveals that CGs follow an elementary concept→relation→concept structure, which describes the ontology and semantics of the enterprise concepts as explained earlier. The CGs are thus directed graphs that capture the metamodel at the logical level including its direction of flow. Specifically, Fig. 1's left-hand side CG is the metamodel for our simple example, and the right-hand side is the specialised model for SHU that conforms to the metamodel. The type label `Vision & Mission`, `Enterprise`, `Place`, and

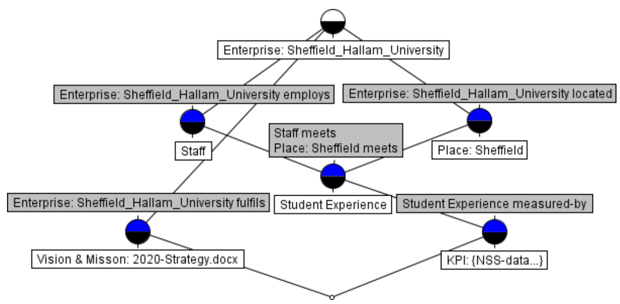
KPI are each specialised by gaining a defined referent, which is an instantiation of the type label. The referent is `2020-Strategy.docx` (a written document), `Sheffield Hallam University` (the enterprise), `Sheffield` (SHU's location) and {`NSS-data...`} (a structured data source) for each type label respectively[1]. The type label `Experience` was specialised to `Student Experience`, which is `Experience`'s subtype.

3.3 Formal Concept Analysis

Formal Concept Analysis (FCA) adds a mathematical level to the logical level captured in CGs [6]. The FCA formal context is generated from the CGs by the *CGFCA* software[2] [1]. Essentially, this software transforms CGs' underlying concept→relation→concept triples structure into source-concept⌒relation→ target-concept binaries thereby making them suitable for FCA. Fig. 2 shows the corresponding formal concept lattices that result by this transformation from the corresponding CGs in Fig. 1.



(a) Metamodel



(b) SHU

Fig. 2: Metamodel and SHU, Formal Concept Lattices

4 Iterating Enterprise Concepts from Formal Concepts

We can see that the *infimum* (bottommost) formal concept in Fig 2 doesn't have its own labels. We will now explore why this is significant.

[1] { } denote 'plural' referents, meaning they hold more than one referent. Here NSS-data may be one of many datasets that collectively provide KPIs of SHU's strategy and shown simply to illustrate multiple cardinality of concepts. The Staff type label would also have a plural referent to depict the many staff that SHU employs. Plural referents are however not elaborated further for this simple case study's purposes.

[2] `https://sourceforge.net/projects/cgfca/`

4.1 An Architectural Principle

As stated earlier, EA takes a holistic perspective. To draw from a building architect's analogy, architecture ranges "From the blank piece of paper to the last nail in the wall." Likewise EA (Enterprise Architecture) follows the same principle; indeed an enterprise is set by its vision and mission (articulated in its strategy) and–taking the analogy to the same extent–applies it to every asset it owns.

4.2 Transitivity of Enterprise Concepts

In reality we would not evaluate every asset to such a extreme, but it demonstrates that enterprise concepts follow a transitive path from the highest level purpose of the enterprise, percolating through its strategic, tactical and operational enterprise concepts as interconnected by their semantic relations to its most specific assets that determine its success. There should be an overall flow from the very top to the very bottom with every concept and relation thus interlinked along the way. In the SHU case study, the 'culprit' is the `fulfils` relation in Fig 1, evident by the upward direction that the arrows point up to `Vision & Mission` from `Enterprise`. All the other arrows point downwards. A formal concept lattice has a *supremum* (topmmost) concept and an *infimum* (bottommost) concept. Notably though, the *infimum* has no labels, so what is it's "...to..." enterprise concept? The CGs suggest it's `KPI`, But it's one of the formal concepts above in the lattice. The answer is that the enterprise concepts in Fig 1 are not all transitive thereby do not concur with the architectural principle. Note also that SHU's model is based on the metamodel, which could be applied to all enterprises thus replicate the issue. Hence we need to validate the metamodel in order to verify the models that are populated from it (in this case SHU).

4.3 Correcting the Transitivity

Given the elementary scope of the case study (and remembering that SHU is in fact a much more sophisticated enterprise), the direction of the arrows around the `fulfils` relation simply need to change direction as stated. This correction is given by Fig 3, which also shows the `fulfils` relation has become `fulfilled-by`. Although it's a simple renaming in this case, the metamodel (and the SHU model) is fully transitive i.e. architectural. FCA, through *CGFCA* identified this architectural gap. The CGs are conventionally generated by hand, akin to how metamodels and models are developed in many EA software tool environments[3]. As indicated earlier, CGs graphical representation serve as a readable, formal design and specification language at a logical level but FCA adds rigour at a mathematical level that allows the formal concepts to be computer generated. The productivity of computers has been applied to the creativity of human thinking—the rationale for conceptual structures (CS).

[3] The tools tend to depict the models and metamodels in other notations such as UML (`www.uml.org`), but this underlying remark still holds true.

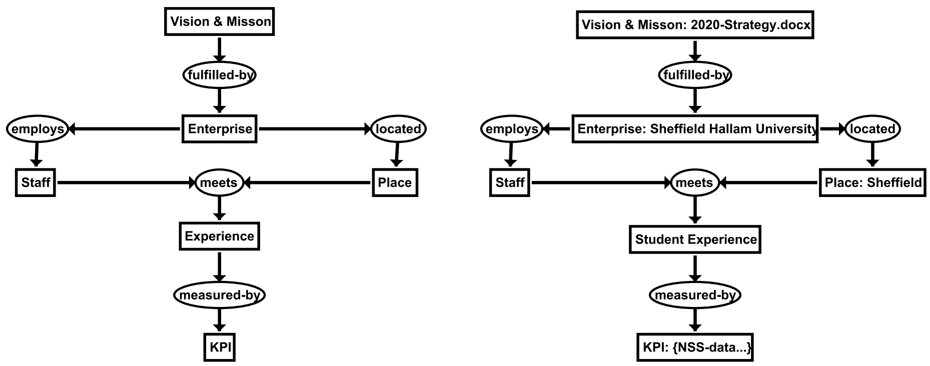
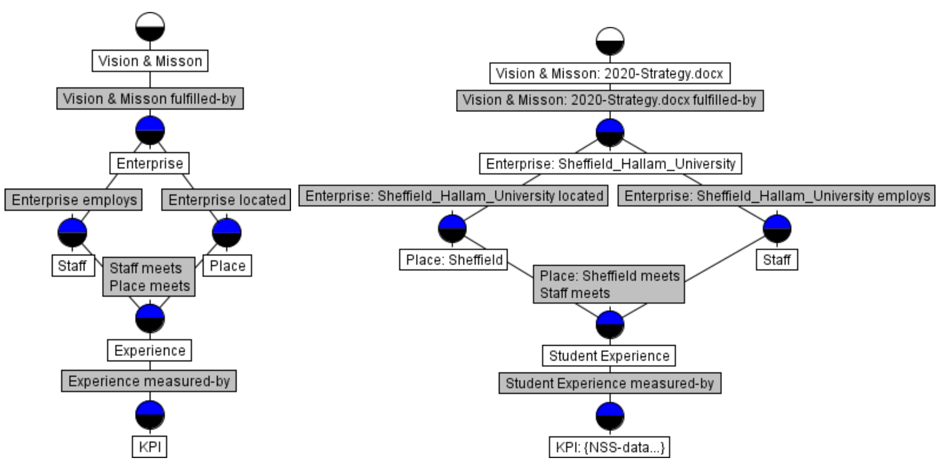


Fig. 3: Corrected Metamodel and SHU, in CGs



(a) Metamodel (b) SHU

Fig. 4: Lattices, after correction

5 Discussion

The significance of the change of arrow direction and relation renaming described above is that the enterprise (SHU) *derives* from it's vision and mission. The 2020 Strategy document is produced by the enterprise (SHU) but that in fact articulates SHU's purpose for existing in the first place. Hence SHU is the *organisational outcome* of that purpose. The "...last nail in the wall" *infimum* formal concept `KPI` brings together all the enterprise concepts to show how they are all (ultimately) measured, hence evaluated and managed for achieving SHU's purpose.

Of course, a straightforward visual inspection of the CGs would reveal that the arrows would all need to be in a fully transitive direction as described. But the simplified SHU case study demonstrates the principle. In reality, and as even the simplified SHU case indicates, the metamodels and models can run to many hundreds and even thousands of interlinked enterprise concepts and semantic relations. An examination of metamodel libraries for example reveals their possible extent [4, 3, 11]. There are also other comprehensive examples that support the *CGFCA* approach [1, 8–10]. Therefore trying simply to inspect the hand-drawn models for misalignments in the enterprise concepts with the human eye would become an arduous if not impossible task, whereas the mathematically, computer generated formal concepts from FCA and *CGFCA* would find them in an instant. Evidencing this further is the subject of planned work, including an extended version of this paper.

6 Conclusions

Enterprise concepts benefit from FCA through *CGFCA*. Following the architectural principle of "The blank piece of paper to the last nail in the wall", *CGFCA* discovers the transitivity in the enterprise concepts, highlighting where that transitivity is deficient. For enterprise concepts articulated through enterprise architecture, the transitivity extends throughout including the *infimum* formal concept. By aligning enterprise concepts with formal concepts, an enterprise's visible entities such as its buildings, equipment or financial statements can thus be directed to support rather than hinder the enterprise. It also serves to remind business enterprises that structure follows strategy; the enterprise's organisational form is the *outcome* of its purpose ('vision and mission').

CGFCA is actually *triples* to *binaries* through FCA. This opens its potential to be generalised to other, more widely-used notations that enterprise modellers take advantage of such as UML Class Diagrams that use directed graphs (which are commonly found in EA metamodels). Going even wider, RDFS and OWL from the Semantic Web or any other notation that uses directed triples could benefit too. The experiences from applying *CGFCA* to enterprise metamodels has also raised these additional avenues. Aligning computer productivity with human creativity is a tenet of conceptual structures, and we have shown that FCA in this sense can be brought to bear to make it so.

References

1. Simon Andrews and Simon Polovina. *A Mapping from Conceptual Graphs to Formal Concept Analysis*, pages 63–76. Conceptual Structures for Discovering Knowledge, The 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK. Springer, Heidelberg, lecture notes in computer science, vol. 6828 (subseries: lecture notes in artificial intelligence) edition, 2011.
2. A. D. Jr Chandler. *Strategy and Structure: Chapters in the History of the American Industrial Enterprise*. MIT Press, Cambridge, MA, 1962.
3. EAS. Essential meta-model reference, 2016. `http://www.enterprise-architecture.org/documentation/doc-meta-model`.
4. The Open Group. 34. content metamodel, 2011. `http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap34.html`.
5. Thomas R. Gruber. A translation approach to portable ontology specifications. *KNOWLEDGE ACQUISITION*, 5:199–220, 1993.
6. Pascal Hitzler and Henrik Scharfe. *Conceptual Structures in Practice*. CRC Press, 2009.
7. Simon Polovina. *An Introduction to Conceptual Graphs*, pages 1–15. Proceedings of the 15th International Conference on Conceptual Structures (ICCS 2007): Conceptual Structures: Knowledge Architectures for Smart Applications, July 2007, Sheffield, UK. Springer, Heidelberg, lecture notes in artificial intelligence (lnai 4604) edition, 2007.
8. Simon Polovina and Simon Andrews. Cgs to fca including peirce's cuts. *International Journal of Conceptual Structures and Smart Applications (IJCSSA)*, 1(1):90–103, 2013.
9. Simon Polovina, Hans-Jürgen Scheruhn, Stefan Weidner, and Mark von Rosing. Discovering the gaps in enterprise systems via conceptual graphs & formal concept analysis. In *Poster Proceedings, The 22nd International Conference on Conceptual Structures (ICCS)*, pages 5–8, 5th to 7th July 2016.
10. Simon Polovina, Hans-Jürgen Scheruhn, Stefan Weidner, and Mark von Rosing. Highlighting the gaps in enterprise systems models by interoperating cgs and fca. In Simon Andrews and Simon Polovina, editors, *Proceedings of the Fifth Conceptual Structures Tools and Interoperability Workshop (CSTIW 2016)*, pages 46–54. CEUR-WS, 5th-7th July 2016.
11. LEADing Practice. Knowledge sharing by practitioners, 2017. `http://www.leadingpractice.com/knowledge-center/knowledge-sharing/`.
12. Roger Sessions. A comparison of the top four enterprise-architecture methodologies, 2007. `http://msdn.microsoft.com/en-us/library/bb466232.aspx`.
13. J. F. Sowa. *Conceptual Graphs*, pages 213–237. Handbook of Knowledge Representation, Foundations of Artificial Intelligence. Elsevier, Amsterdam, volume 3 edition, 2008.
14. J. F. Sowa and J. A. Zachman. Extending and formalizing the framework for information systems architecture. *IBM Syst.J.*, 31(3):590–616, jun 1992.
15. John F. Sowa. *Conceptual Structures - Information Processing in Mind and Machine*. Addison-Wesley, the systems programming series edition, 1984.
16. John A. Zachman. John zachman's concise definition of the zachman framework. `https://www.zachman.com/about-the-zachman-framework`.
17. John A. Zachman. A framework for information systems architecture. *IBM Syst.J.*, 26(3):276–292, sep 1987.