Highlighting the Gaps in Enterprise Systems Models by Interoperating CGs and FCA

POLOVINA, Simon <http://orcid.org/0000-0003-2961-6207>, SCHERUHN, Hans-Jürgen, WEIDNER, Stefan and VON ROSING, Mark

Highlighting the Gaps in Enterprise Systems Models by Interoperating CGs and FCA

Simon Polovina[1], Hans-Jürgen Scheruhn[2],
Stefan Weidner[3], and Mark von Rosing[4]

[1] Conceptual Structures Research Group, Sheffield Hallam University, UK
[2] Business Informatics, Hochschule Harz, Wernigerode, Germany
[3] SAP UCC & School of Computer Science, University Magdeburg, Germany
[4] LEADing Practice ApS | Global Univerisity Alliance, Denmark
s.polovina@shu.ac.uk, hscheruhn@hs-harz.de,
weidner@ucc.uni-magdeburg.de, mvr@leadingpractice.com

Abstract. Enterprises arise from creative human endeavours, articulated through business concepts encoded in enterprise information systems through a modular Enterprise Information Model (EIM). The EIM thus brings the productivity of computers to bear. Essentially, the EIM represents conceptual structures, which align the computer's structured way of working with the human's conceptual way of thinking. Using an industrial-strength SAP exemplar known as 'Global Bike Inc.', and expressing its EIM's meta-objects as meta-object → relation → meta-object, Conceptual Graphs (CGs) simplified the EIM's modules, which consist of four business layers and two information systems layers. The logical simplification of these modules is extended into four levels of detail that culminate in performance indicators being assigned to each of the six layers. From the CGs, Formal Concept Analysis (FCA)'s *CGtoFCA* algorithm was used to generate the meta-object⌒relation → meta-object binaries that identified the pathways layer-wise and level-wise between the meta-objects. Through the interoperability of CGs and FCA, gaps in the conceptual structure of the EIM as highlighted by its performance indicator or measure, implying that the layer is not as modular as intended.

Keywords: Enterprise Information Model (EIM), Business Layers (Value, Competency, Service, Process), Information Systems Layers (Application, Data), Levels (Enterprise, Department, Workplace, Performance Indicator / Measure), GBI (Global Bike Inc.), ARIS (Architecture of Integrated Information Systems), SAP Solution Manager, Enterprise Resource Planning (ERP) Systems, Meta-Objects, Conceptual Graphs, Formal Concept Analysis

1 Introduction

Enterprise information systems are nowadays mainstream to business activity [16]. Enterprises arise from creative human endeavours, articulated through high-level business concepts that in turn are embodied by the lower-level data structures that computerised enterprise system applications rely on to bring their productivity to bear [10]. An exemplar of enterprise systems to date is SAP SE (www.sap.com). SAP is the world's market-leader for enterprise systems software and services. SAP (which in English stands for Systems, Applications and Products) have built their reputation on understanding business systems. Given its impact, this vendor thus forms the background of our work as we now present.

SAP's systems tend to be purchased and used by larger scale enterprises, with consequently large-scale data sets. Consequently, learning SAP for newcomers is a major task without having a reference implementation of this scale to understand its capabilities and potential [5]. To address this issue, Global Bike Inc. (GBI) has been developed[5] that is used world-wide by SAP's University Alliances program for learning, research and industrial practice (uac.sap.com) [3]. As well as learning material and supporting documentation, GBI includes a Enterprise Information Model (EIM) that demonstrates GBI's alignment of business, information systems and the underlying technology [13]. The model uses the well-known and industry-standard ARIS software that provides a range of modelling constructs in a comprehensive software tool from Software AG (www.softwareag.com). Furthermore the EIM is connected to a running SAP enterprise system implementation through an SAP-specific product known as SAP Solution Manager [12]. The GBI business and information system concepts are depicted as meta-objects and are interconnected with other meta-objects through semantic relations, as we will explain later on in the paper. Pertinent to understand meanwhile is that human creativity through business concepts are articulated into a form that is encoded into a application or data structure that the computer can process.

An opportunity therefore arises to explore the effectiveness of this contemporary approach using Conceptual Structures (CS), given its stated purpose of harmonising the creativity of humans with the productivity of computers [2]. We might expect that CS will discover hidden meaning or gaps as previous work has shown using a technique that maps Conceptual Graphs to Formal Concept Analysis [7]. It is this approach, but this time with an industrial-strength example (i.e. GBI) that we now investigate and present the findings.

2 Conceptual Structures

Table 1 is derived from previous work that describes the EIM [11]. In that work, the particular industry of GBI (bike manufacture) is used to demonstrate a general EIM that can be applied to all industries. This EIM shows the meta-objects of the business layer (Value, Competency, Service and Process) and the information system layer (Application and Data). These horizontal layers are further described by vertical levels ranging from 1 to 4, where 1 are the most high-level meta-object and 4 the most specific meta-objects. Level 1 refers to the whole

[5] The SAP UA GBI environment consists of a set containing three parts: 1) SAP solutions (landscape), 2) GBI model company, and 3) curriculum material (including slides, exercises, case studies)

Table 1. Information Meta Objects Mapping

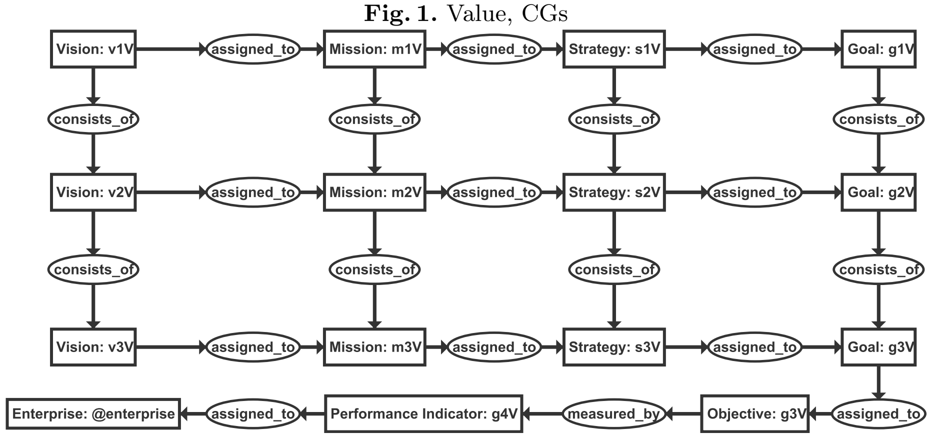
Layer:	Business Layer				Information System Layer (ERP, HANA)	
	Value	Competency	Service	Process	Application	Data
Level						
1	Vision, Mission	Business Function	Business Service	Business Process	Application Module	Enterprise Data Cluster
	Strategy, Goal	Organizational unit			Organizational unit	
2	Vision, Mission	Business Function	Business Service	Process Step	Application Function	Department Data Cluster
	Strategy, Goal	Organizational unit			Organizational unit	
3	Vision, Mission	Business Function	Business Service	Process Activity	Application Task	Workplace Data Entity
	Strategy, Goal				Transaction Code, System organizational Unit	
						Dimension
	Objective	Business Object		Event	Business Object	
					Data Entity Event	Data Entity
		Business Media / Accounts		Data Object	Data Object (Media)	
		Business Roles	Services Roles	Process Role	Application Roles	
		Business Roles	Service Rules	Process Rules	Application Rules	
4	Performance Indicator	Business Compliance	Service Level Agreement (SLA)	Process Performance Indicator (PPI)	IT Governance	Fact Table Customizing Data Table
						Master Data Table / View
						Transaction Data Table
		Revenue/ CostFlow			System Measurements	Key Foreign Key Describing Attributes

enterprise, level 2 a department or organisational unit (e.g. sales) within the enterprise, level 3 the actual workplace (e.g location) in the organisational unit, and level 4 is the performance indicator and other measures used to evaluate the alignment and effectiveness of its above 3 layers. Level 1 is thus the most conceptual and level 4 the most structural. As concepts and structures are thereby aligned as meta-objects in the matrix given by this table, it offers an initial conceptual structure of the EIM.

3 Conceptual Graphs and Formal Concept Lattices

From this starting point, Conceptual Graphs (CGs) enabled us to focus on the meta-objects and their relations, through CGs' simple concept → relation → concept nature [6, 14]. Furthermore we could make use of CGs `[Type-Label: Referent]` components in each CGs concept. To identify gaps and missing meaning (semantics) and in line with previous work, FCA (Formal Concept Analysis) was then applied through the *CGtoFCA* algorithm [1, 7, 8]. The outcome is then presented as a Formal Concept Lattice (FCL). A CG was produced for each layer [6, 14]. Each layer has a meaning its own right given their distinctive headings (i.e. Value, Competency, Service, Process, Application, Data). Our intention was thereby to capture each layer as a modular 'semantic unit' in its own right. The result for the Value Layer in the EIM is accordingly shown in Fig. 1.

Fig. 1. Value, CGs



The Value CG depicts the each meta-object name (i.e. Vision, Mission, Strategy, Goal) as a CG type label. To instantiate it an a particular meta-object, a unique identifier appears in the referent field. For example, `v1V` denotes that a meta-object that is Vision (v), Level 1 (1), and V (Value layer). Likewise, `g3V` for example describes Goal, Level 3, Value and so on. The `[Enterprise: @enterprise]` concept follows an alternative pattern where (`@enterprise`) is a CGs measure referent. The pointer to Enterprise follows that of the previous work [1, 7, 8]. The overall significance of this concept that all the activities that make up an enterprise ultimately point to the enterprise[6], even though Enterprise is absent in the table. The relations (e.g. ((assigned_to)) also do not appear in the table; they are however in the EIM. Space does not permit the detailed description of the relations but are explicated elsewhere [11]. Essentially ((assigned_to) refers to a horizontal relation usually in the same layer while (consists_of) is a vertical relation between the levels in the layers. (There is no associated layer or level for Enterprise as it reflects the ultimate culmination of all the layers and levels). The relation measured-by has its usual meaning. The FCL for the Value layer will be discussed shortly.

The CG and the FCL for the next layer i.e. Competency is not shown due to space restrictions. Fig. 2 shows the following Service layer CG, which nonetheless highlights similar findings to that of Competency. It exhibits the same [Type-Label: Referent] and (relation) pattern as illustrated by Fig. 1 and its explanation above. So, for example bs1S in [Business Service: bs1S] describes bs for Business Service, 1 is Level 1, and S is Service. In this layer there is an (occurrence_copy) relation too. Essentially, this relation describes two meta-objects that are synonymous, except they appear in different layers. For example, [Business Service: bs1S] → (occurrence_copy) → [Business Process: bp1P]. They are therefore not co-referent, and might be described as 'pseudo-synonym' meta-objects. The FCL generated by *CGtoFCA* for the Service layer is shown by figure 3.

Fig. 3 is the result of the *CGtoFCA* algorithm transforming the meta-object→ relation → meta-object triples in the CG of Fig. 2 to meta-object⌒relation→ meta-object binaries[7]. Like in the previous work, it thus identified the pathways layer-wise and level-wise between the concepts (meta-objects) and (semantic) relations [1,7,8]. Using index numbers to illustrate that the relations are distinct from each other (e.g. assigned_to$N$ where $N\geq 1 \leq 3$), the *CGtoFCA* FCL Fig. 3 evidences that [Enterprise: @enterprise] is not bottommost. That is because of the meta-object⌒relation attributes that are outside the intent of the level 4 key performance indicator (KPI) meta-object [Service Level Agreement (SLA): sla4s], which evaluates the Service layer.

An inspection of Fig. 2 shows that there are the three (occurrence_copy) relations that do not point directly or indirectly through the other concepts to [Enterprise: @enterprise] in that figure. As they are not co-referent they cannot simply be removed. To do so would break the link with the associated meta-objects in the Process layer. Nor should vertical relations be (arbitrarily) added; such relations would anyway be outside of the semantic scope of this

[6] We could have stated [Enterprise: GBI] but given our earlier remark about the generality of the EIM we chose @enterprise.

[7] The *CGtoFCA* algorithm is implemented in software (http://cgfca.sourceforge.net/) and uses the CGIF file format output from CharGer (http://charger.sourceforge.net/), in which the CGs were drawn. Concept Explorer (http://conexp.sourceforge.net/) was used to generate the FCL.

Fig. 2. Service, CGs

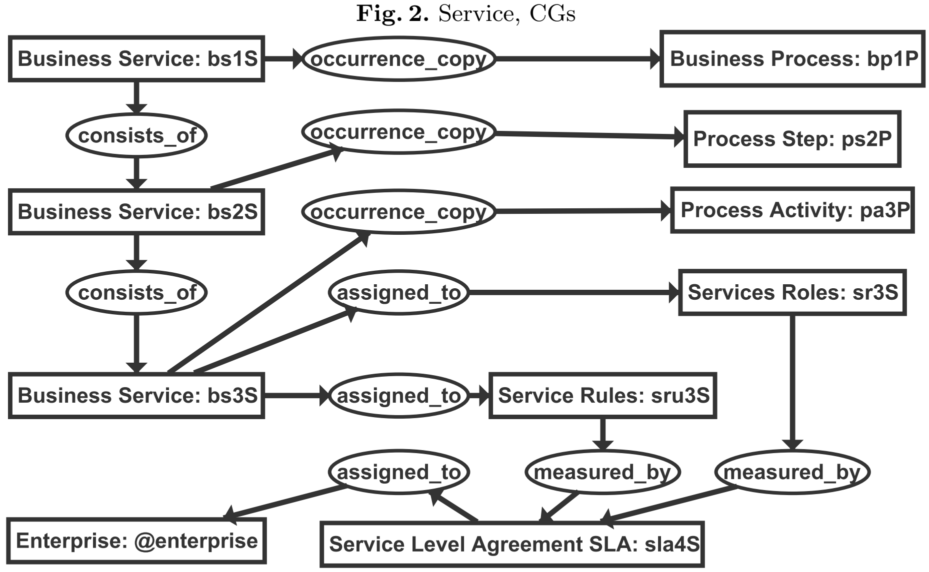
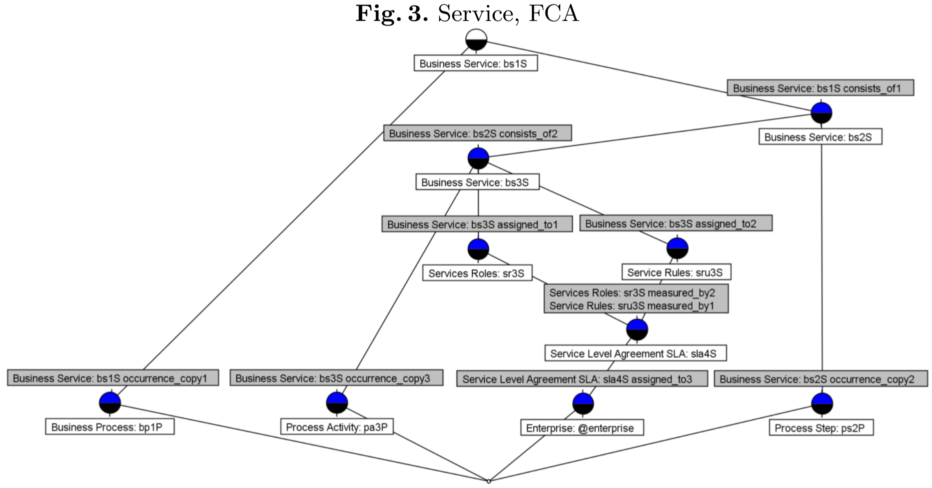
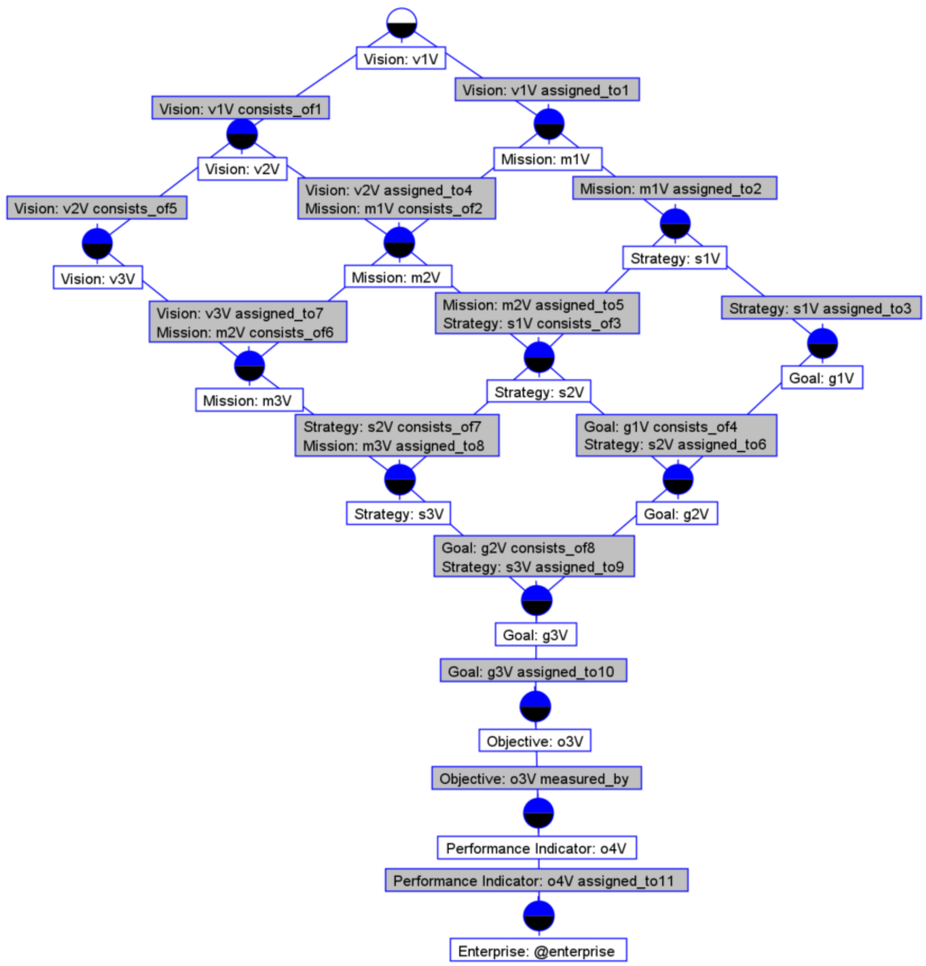


Fig. 3. Service, FCA



layer, which is Service not Process. Rather, these (occurrence_copy) destination meta-objects in Process represent a messaging structure between these layers. As it stands, the layer is not a self-contained semantic unit. Although not shown the same happens for the Competency layer meta-objects. In passing it is worth remarking in ARIS that these occurrences are the same object, so in that sense the problem does goes away. But they are 'pseudo-synonyms', evidenced by them not being co-referent semantically. Put simply, Process needs Service to be a semantic unit, and vice versa. The same issue applies to Competency and its same dependencies with Service.

Fig. 4. Value, FCA



The CGs and FCL for the remaining layers (i.e. Process, Application, Data) are not shown once more due to brevity of space. Application and Data are however illustrated elsewhere [9], in which Data in the EIM emerges to be a semantic unit i.e. `[Enterprise: @enterprise]` is the object in the bottommost concept. This also happens with the Value FCL that is now presented in Fig. 4. This is evidenced by Fig. 1 in which the arrows in the Value CG show a flow from one starting point (`v1V`) to one ending point (`@enterprise`). Value and Data can thus be recognised as meta-objects in their own right. Such a meta-object would be a meta-object of all the meta-objects inside their respective layers. An analogy emerges with the object-oriented principle of high cohesion (where expressibility is encapsulated within the layer) and low coupling (where the dependencies between the layers are decreased as much as possible), thereby reducing the maintenance and modification costs of the enterprise system, the models and the human energy to maintain them [4, 15, 17].

4 Conclusions

GBI, ARIS and SAP Solution Manager provide a comprehensive and detailed EIM (Enterprise Information Model), in the pursuit of conceptual structures. CGs (Conceptual Graphs) and FCA (Formal Concept Analysis) highlighted the gaps in the existing EIM. These discoveries were achieved by a focus on the layers (from Value to Data) and their levels, ranging from 1 to 4 denoting their level of detail going from the most conceptual to the most structural. Through the interoperability of CGs and FCA, gaps in the conceptual structure of the EIM were highlighted in certain layers (i.e. Competency, Service, Process and Application) and its performance indicator or other measure (i.e. level 4), implying that these layers (unlike Value and Data) may not be as modular as intended. Enterprise systems could thereby benefit from layers made up of semantic units with better-defined interfaces (borrowing ideas from object-orientation) rather than as present. Taking this approach, enterprise systems can increase their effectiveness in contributing to the enterprise's human-driven purpose.

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. Simon Andrews, Simon Polovina, Richard Hill, and Babak Akhgar. Conceptual structures for discovering knowledge, 19th international conference on conceptual structures, iccs 2011,. volume 6828, pages XIV–424. Springer, July 25-29 2011.
3. Simha R Magal and Jeffrey Word. *Integrated business processes with ERP systems*. Wiley Publishing, 2011.

4. Andrian Marcus and Denys Poshyvanyk. The conceptual cohesion of classes. In *Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on*, pages 133–142. IEEE, 2005.
5. Dirk Peters. *Adaptive Lehr-und Lernsysteme zur Unterstützung der praktischen Ausbildung an ERP-Systemen.* Shaker, 2014.
6. Simon Polovina. An introduction to conceptual graphs. In *Proceedings of the 15th International Conference on Conceptual Structures: Knowledge Architectures for Smart Applications*, ICCS '07, pages 1–14, Berlin, Heidelberg MY Reference 4, 2007. Springer-Verlag.
7. Simon Polovina. A transaction-oriented architecture for enterprise systems. *International Journal of Intelligent Information Technologies (IJIIT)*, 9(4):69–79, 2013.
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 *The International Conferences on Conceptual Structures (ICCS).* (in press), 5th to 7th July 2016.
10. August-Wilhelm Scheer and Alexander Hars. Extending data modeling to cover the whole enterprise. *Communications of the ACM*, 35(9):166–ff, 1992.
11. H. Scheruhn, R. Fallon, and M. Rosing. *Information Modelling and Process Modelling*, volume 1 of *The Complete Business Process Handbook. Body of Knowledge from Process Modelling to BPM*, pages pp 511–550. Elsevier, 2015.
12. Hans-Jürgen Scheruhn, Daniel Ackermann, Roman Braun, and Ulrich Förster. *Human-Computer Interaction. Users and Contexts of Use: 15th International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part III*, chapter Repository-Based Implementation of Information Pyramid: A Study Based on an ERP Case Study, pages 446–455. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
13. Hans-Jürgen Scheruhn, Sandro Sicorello, and Stefan Weidner. *Business Information Systems Workshops: BIS 2012 International Workshops and Future Internet Symposium, Vilnius, Lithuania, May 21-23, 2012 Revised Papers*, chapter Repository-Based ERP Case Studies: A Study about Chances and Benefits of Agile Case Study Development, pages 186–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
14. J. F. Sowa. *Conceptual Graphs*, pages 213–237. Handbook of Knowledge Representation, Foundations of Artificial Intelligence. Elsevier, Amsterdam, volume 3 edition, 2008.
15. W.P. Stevens, G.J. Myers, and L.L. Constantine. Structured design. *IBM Systems Journal*, 13(2):115–139, 1974.
16. Aslam Usman, Crispin Coombs, and Doherty Neil. Use of erp systems: a social shaping perspective. *MCIS 2014 Proceedings*, 2014.
17. Edward Yourdon and Larry L. Constantine. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design.* Yourdon Press, 1979.