Modularising the complex meta-models in enterprise systems using conceptual structures

POLOVINA, Simon <http://orcid.org/0000-0003-2961-6207>, SCHERUHN, Hans-Jurgen and VAN ROSING, Mark

Published version

POLOVINA, Simon, SCHERUHN, Hans-Jurgen and VAN ROSING, Mark (2017). Modularising the complex meta-models in enterprise systems using conceptual structures. In: SUGUMARAN, Vijayan, (ed.) Developments and trends in intelligent technologies and smart systems. Advances in Computational Intelligence and Robotics (ACIR) . Hershey, PA, IGI Global, 261-283.

Developments and Trends in Intelligent Technologies and Smart Systems

Vijayan Sugumaran
Oakland University, USA

A volume in the Advances in Computational Intelligence and Robotics (ACIR) Book Series

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Sugumaran, Vijayan, 1960- editor.
Title: Developments and trends in intelligent technologies and smart systems / Vijayan Sugumaran, editor.
Description: Hershey, PA : Engineering Science Reference, [2018] | Includes bibliographical references.
Identifiers: LCCN 2017020847| ISBN 9781522536864 (hardcover) | ISBN 9781522536871 (ebook)
Subjects: LCSH: Systems engineering. | Self-organizing systems.
Classification: LCC TA168 .D496 2018 | DDC 620/.0042--dc23 LC record available at https://lccn.loc.gov/2017020847

This book is published in the IGI Global book series Advances in Computational Intelligence and Robotics (ACIR) (ISSN: 2327-0411; eISSN: 2327-042X)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 13

Modularising the Complex Meta-Models in Enterprise Systems Using Conceptual Structures

Simon Polovina
Sheffield Hallam University, UK

Hans-Jurgen Scheruhn
Hochschule Harz, Germany

Mark von Rosing
Global University Alliance, Denmark

ABSTRACT

The development of meta-models in Enterprise Modelling, Enterprise Engineering, and Enterprise Architecture enables an enterprise to add value and meet its obligations to its stakeholders. This value is however undermined by the complexity in the meta-models which have become difficult to visualise thus deterring the human-driven process. These experiences have driven the development of layers and levels in the modular meta-model. Conceptual Structures (CS), described as "Information Processing in Mind and Machine", align the way computers work with how humans think. Using the Enterprise Information Meta-model Architecture (EIMA) as an exemplar, two forms of CS known as Conceptual Graphs (CGs) and Formal Concept Analysis (FCA) are brought together through the CGtoFCA algorithm, thereby mathematically evaluating the effectiveness of the layers and levels in these meta-models. The work reveals the useful contribution that this approach brings in actualising the modularising of complex meta-models in enterprise systems using conceptual structures.

DOI: 10.4018/978-1-5225-3686-4.ch013

INTRODUCTION

The development of meta-models in Enterprise Modelling, Enterprise Engineering, and Enterprise Architecture provide insight into the complexity of bus ness organizations (Bork et al., 2015; von Rosing and von Scheel, 2016). These meta-models are extensible across whole industries, individual businesses, their sub-organisations (e.g. departments) and individual workplaces where the actual activity takes place. Thus, a business enterprise's myriad resources (e.g. physical assets, human resources and IT systems) can be aligned with its purpose and strategy ('vision and mission'). The meta-models thereby facilitate enterprises to develop a conceptual model that creates the right context. These meta-models thereby enable enterprises to add value and reduce unnecessary cost and risk in meeting its obligations to its stakeholders (e.g. shareholders, employees, regulatory bodies and the wider environment).

Computer science has over its history contributed to the expressibility in these meta-models through its advances in ontology and semantics; together they capture the objects and relations that describe the interplay and effects of business in a formal, computable model (Floyd, 1967; Gruber, 1995; Oberle, 2013; von Rosing and Laurier, 2015). Computer productivity is thus brought to bear on the creativity of human endeavour, which identifies and sustains enterprise opportunities. Enterprise Architecture and modelling tools are predicated on formally conceptualized meta-models, and this success is already evident (Mayall & Carter, 2015; Bork et al., 2015; von Rosing et al., 2015; von Rosing, 2016).

The meta-models themselves however have become large, unwieldy and error-prone. Whilst the size of these models does not initially present a computational hurdle and the software can reveal errors and gaps that surface to human modellers (e.g. enterprise architects) and end-users (e.g. business decision-makers), the readability of the original meta-models have become illegible thus unreviewable by the human modellers. This aspect is pertinent; given the models are instigated by humans they should be re-viewable by them.

To support this review, there needs to be a consistency of concepts and their relations in these meta-models. The objects, their subtypes, descriptions, semantic relations and how they are viewed that collectively make up the meta-models must be consistently interrelated including the level at which they relate and how they could or should interconnect. For example, enterprise strategy permeates across all the areas of an enterprise; it should not just be captured as a disjointed function. Added to these mistakes are the uneven levels of composition and decomposition of the objects and relations. Put simply, the objects are wrongly thrown together at arbitrary levels, in what apparently are obvious connections but emerge to be much more complex. The meta-model ends up undermining rather than elucidating the effectiveness of the enterprise.

LAYERS AND LEVELS IN ENTERPRISE META-MODELS

To rebalance human creativity with computational execution, meta-models have been broken down into components then coupled together by interfaces analogous to the software engineering principles found in object-oriented design. Like programming-in-the-large, this approach has enabled 'metamodelling-in-the-large' (Zivkovic and Karagiannis, 2015). In Enterprise Architecture, the meta-models have been modularised into layers and levels that collectively describe how a business works. A study describes the benefits of this approach (Bork, 2015). The outcome is a matrix structure that is superficially akin to the grid originally pictured by Zachman, the 'father of enterprise architecture' (Zachman, 1987; Sow& Zachman, 1992).

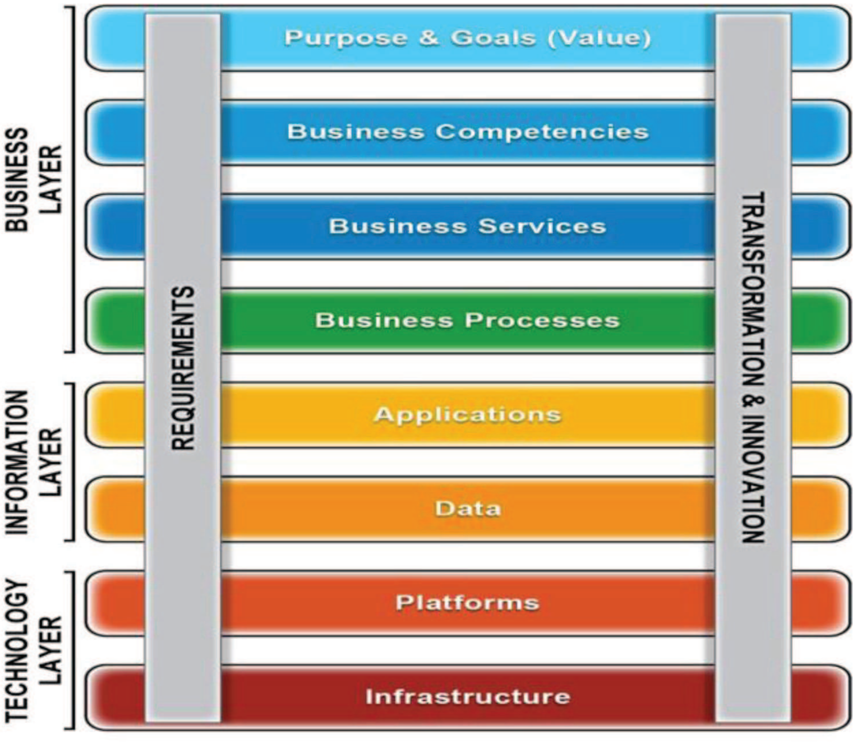
Unlike Zachman however, these layer-and-level components have associated meta-models that include the underlying ontology and semantics, even though Zachman's framework is referred to as the 'Enterprise Ontology' (Zachman, 2015; Malik, 2009). One well-known metamodel example is the Open Group Architecture Framework (TOGAF)'s Content Metamodel (Group, 2011). It articulates the ontology and semantics by formally identifying the relations between the entities (objects) in the meta model. Given that a meta-model is the model about the model, we can refer to these entities or objects as meta-objects.

Semantics are an aspect of semiotics, like syntax, which distinguishes valid from invalid symbol structures, and like pragmatics, which relates symbols to their meaning within a context e.g. the community in which they are shared (Cordeiro & Filipe, 2004). Organizations should thus be considered holistically according to views and models that are laid out in a principled way that capture the:

- Business Perspective: Such as the purpose and goal, competencies, processes, and services aspects;
- Information Perspective: Such as the application systems, as well as the data components;
- Technology Perspective: Such as the platform and infrastructure components.

From the research and analysis conducted by the Global University Alliance (GUA)[1]. The GUA has been developing these contexts and structures, the most common identified structures and context in organizations are represented in Figure 1.

Figure 1. The layered enterprise view



Relating Layers to Levels

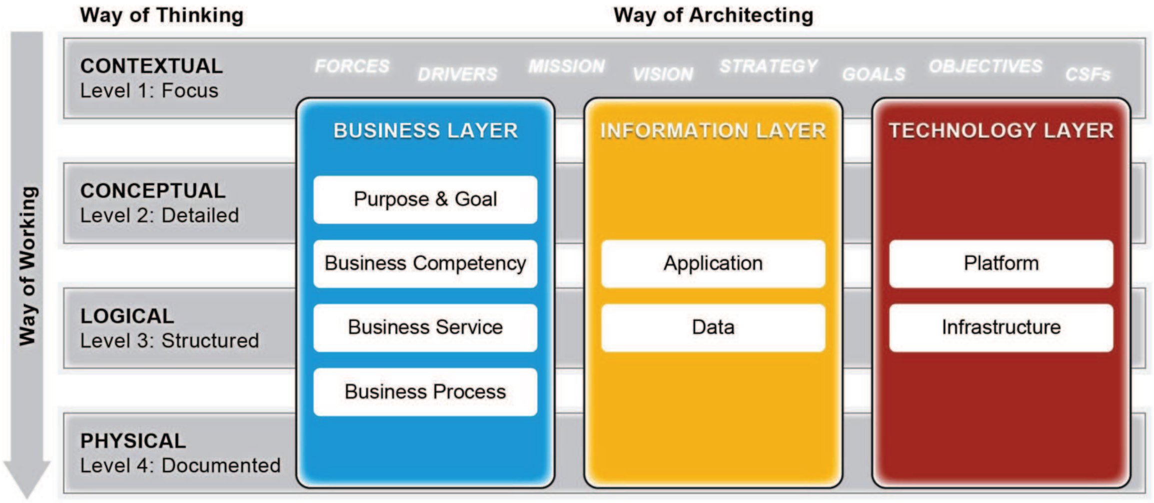
Using layers enables the enterprise metamodel to be modularised so that it is human understandable. For example, a policy, act, regulation or even a strategy is a part of the business layer, while the application systems and data aspects is a part of the information systems layer. In Enterprise Architecture (as well as Enterprise Modelling and Enterprise Engineering) such layers need to relate to the levels represented within the organisation. Relating levels to layers creates a matrix structure as illustrated in Figure 2, where the overall enterprise layers i.e. business, information and technology are on top and the levels with the relevant views are represented on the left side.

Figure 2's Layered Enterprise way of working was developed by Zachman and the Global University Alliance (GUA), as mentioned earlier. and fathered the Business Ontology (von Rosing & Laurier, 2015). The practitioners' enterprise standard body LEADing Practice[2] has embodied this work as the Layered Enterprise Architecture Development (LEAD) (von Rosing & von Scheel, 2016).

THE EIMA

A thread within this work is the Enterprise Information Model (EIM), which demonstrates the above described modularized layout (Scheruhn et al., 2015; Polovina et al., 2016b). The EIM features an information-centric view that takes as its starting point the information concepts within the enterprise. It thus has a layer for value, one for competency, and others for service, process, IT applications and data respectively. Furthermore, these models occur at different levels in the enterprise, hence the layers-and-levels structure. Thus level 1 may be the enterprise itself, level 2 its departments, and level 3 the workplaces. Then last but not least, level 4 documents the other 3 layers. It governs the input and output data structures. These reflect the enterprise's external and internal environment that provides its op-

Figure 2. The layered enterprise way of working



portunities and sets its constraints from level 1 downwards. As such, level 4 epitomizes the performance indicators of the other 3 layers that make up each layer.

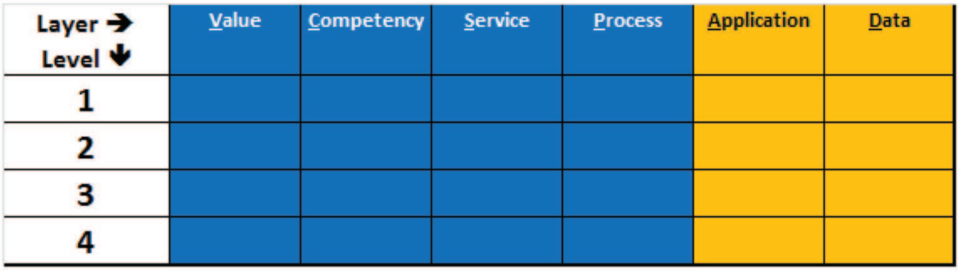
Figure 3 outlines the layer-and-level structure as the resulting 6 x 4 matrix. The shading highlights that Value, Competency, Service and Process are a Business Layer. Application and Data are an Information Systems Layer shown in the earlier Figures 1 and 2. Table 1 explicates the layers and shows how that matrix is populated with the corresponding information concept specific meta-objects. They illustrate the meta-model matrix for the EIM i.e. the Enterprise Information *Meta*-model Architecture (EIMA).

The modularisation depicted in the matrix can equally be applied to other enterprise meta-models, including those that are orthogonal to it e.g. the a) business, informational, application and technology layers with the b) conceptual, logical and physical levels for each layer in the Essential meta-model (Mayall and Carter, 2015). The EIMA is applied to an SAP ERP (Enterprise Resource Planning) exemplar known as ‘Global Bike Inc.’ (GBI) that is used in SAP’s University Alliances program[3]. The GBI exemplar incorporates the GUA and LEADing Practice’s Layered Enterprise views, initiated by the earlier Figures 1 and 2. It continues previous academic work based on SAP as a case study, as a market-leadership and industrial-strength exemplar (Scheruhn et al., 2006; Scheruhn et al., 2013; Scheruhn et al., 2015; Zhao et al., 2014; Polovina et al., 2016a; Polovina et al., 2016b). The case study therefore serves as a further reminder of the enterprise layers-and-levels underlying rigor and practical application.

CONCEPTUAL STRUCTURES

There are undoubtedly many ways that the metamodels effectiveness could be evaluated. Whilst the comparative benefits of each approach are not evaluated here, amongst them is the disciple of Conceptual Structures (CS). In his seminal text, Sowa describes CS as “Information Processing in Mind and Machine” (Sowa, 1984). Enterprises essentially arise as acts of human creativity in identifying business opportunities or other organizational 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 (among them Essential as mentioned earlier) bring the productivity of computers to bear, 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 meta-models within and across the layers and levels of the EIMA.

Figure 3. EIMA overview



Layer → Level ↓	Value	Competency	Service	Process	Application	Data
1						
2						
3						
4						

Table 1. Enterprise Information Meta-model Architecture (EIMA)

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

To demonstrate CS, Sowa devised Conceptual Graphs (CGs) (Sowa, 1984; Polovina, 2007; Sowa, 2008). 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.

Although CGs provide a logical level of rigor, their constituent concepts and relations are essentially put together by hand according to the human's subjective interpretation of the real-world phenomena for it to be captured in a logical structure. A second form of CS known as Formal Concept Analysis (FCA) provides an objective mathematical interpretation of CGs' logical but subjective human interpretations (Ganter et al., 2005). FCA is brought to bear through the *CGtoFCA* algorithm (Andrews and Polovina, 2011). The outcome is then presented as a Formal Concept Lattice (FCL). A CG (Conceptual Graph) was produced for each layer. Each layer has a meaning its own right given their distinctive headings (i.e. Value, Competency, Service, Process, Application, and Data). Our intention was thereby to capture each layer as a modular 'semantic unit' in its own right.

The Business Layer

Value

The result for the Value Layer module is accordingly shown in Figure 4. It reveals how CGs follow an elementary concept→ relation concept structure that describes the ontology and semantics of the metamodel as explained earlier. Furthermore the figure shows how we can make use of CGs [Type-Label: Referent] components in each CGs concept. Its significance will be explained during the following discussion.

The Value CG depicts the each meta-object name (i.e. Vision, Mission, Strategy, Goal) as a CG type label. To instantiate it 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 previous work (Polovina et al., 2016a; Polovina et al., 2016b). The key significance of this concept is that all the activities that make up an enterprise ultimately point to the enterprise, 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 (Scheruhn et al., 2015). 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.

Figure 5 shows the FCL (Formal Concept Lattice) ((Formal Concept Lattice) for the Value layer. It is the result of the *CGtoFCA* algorithm transforming the meta-object → relation → meta-object triples

Figure 4. Value, CGs

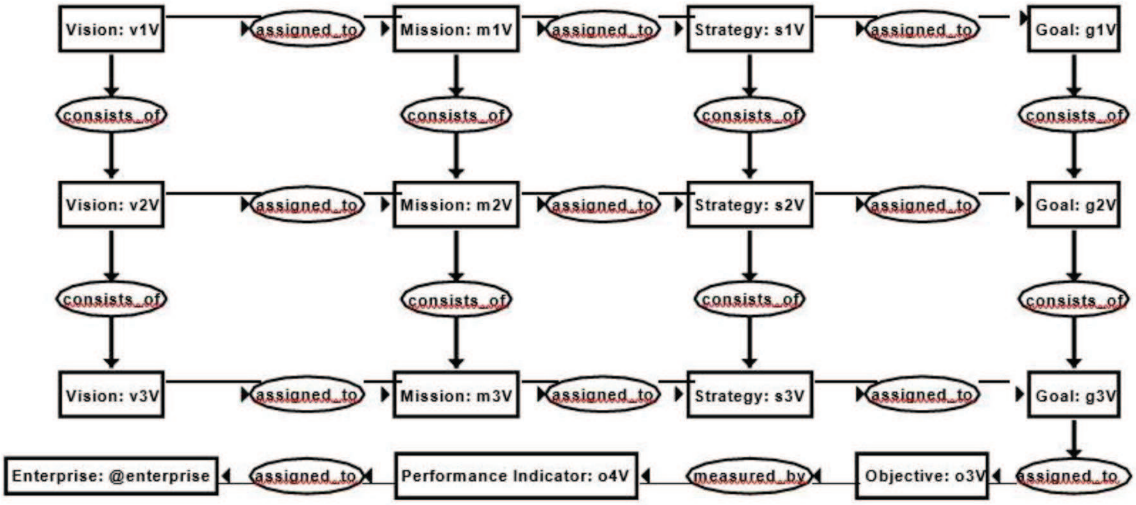
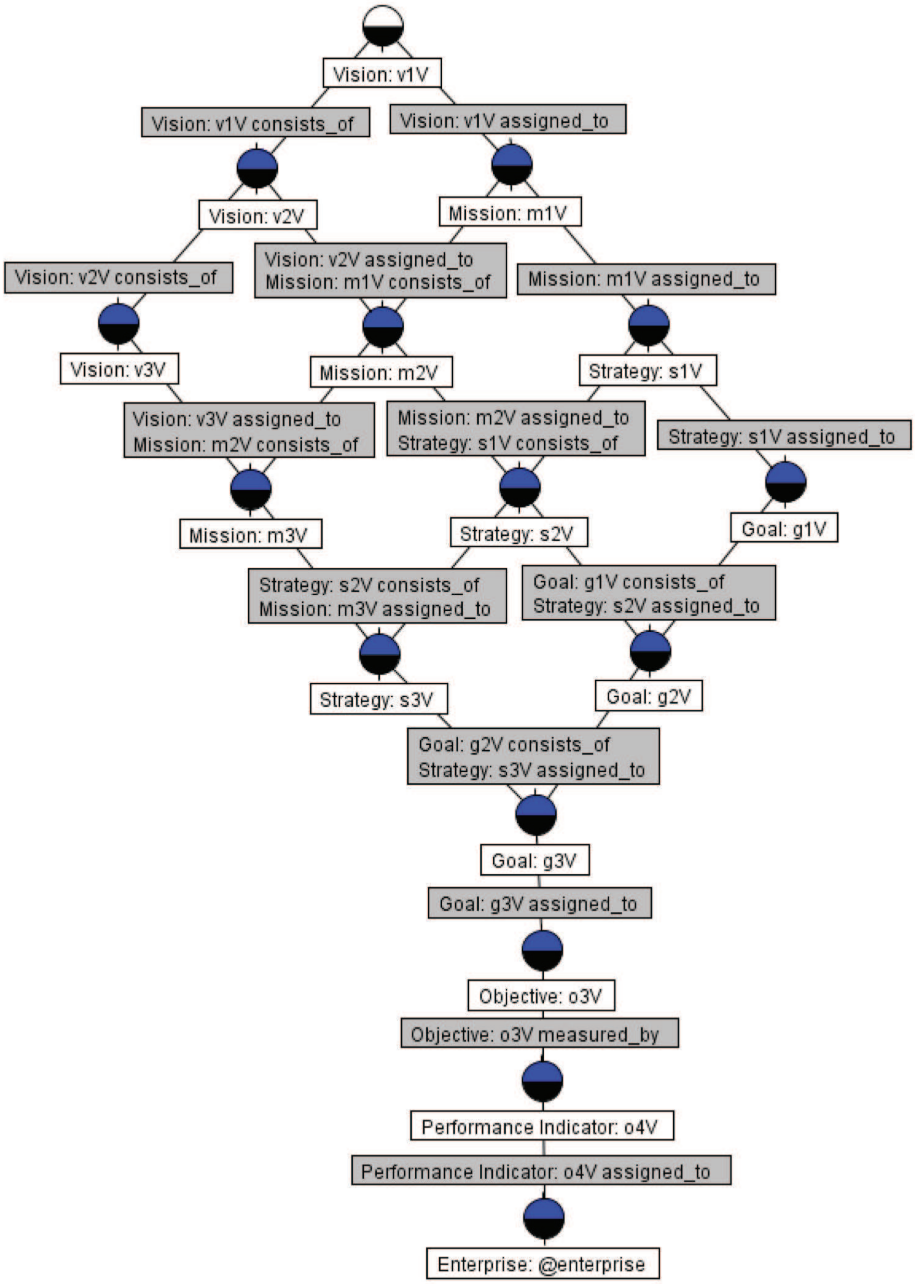


Figure 5. Value, FCL



in the CG of Figure 4 to meta-object relation → meta-object binaries[4]. An example binary is Vision:v1V assigned to→Mission: m1V.

The neatly displayed lattice shows that [Enterprise: @enterprise] is bottommost. It is arguably a semantic unit, as the concept [Vision: v1V] passes transitively through the intermediate concepts and culminating in [Enterprise:@enterprise]. In FCA terminology a CGs concept (that we've mapped to a meta-object) is referred to as an FCA object and, in *CGtoFCA's* case, the meta-object relation is an FCA attribute. A concept in FCA – called a Formal Concept – is the result of when certain conditions are met in a formal *context*. Mathematically:

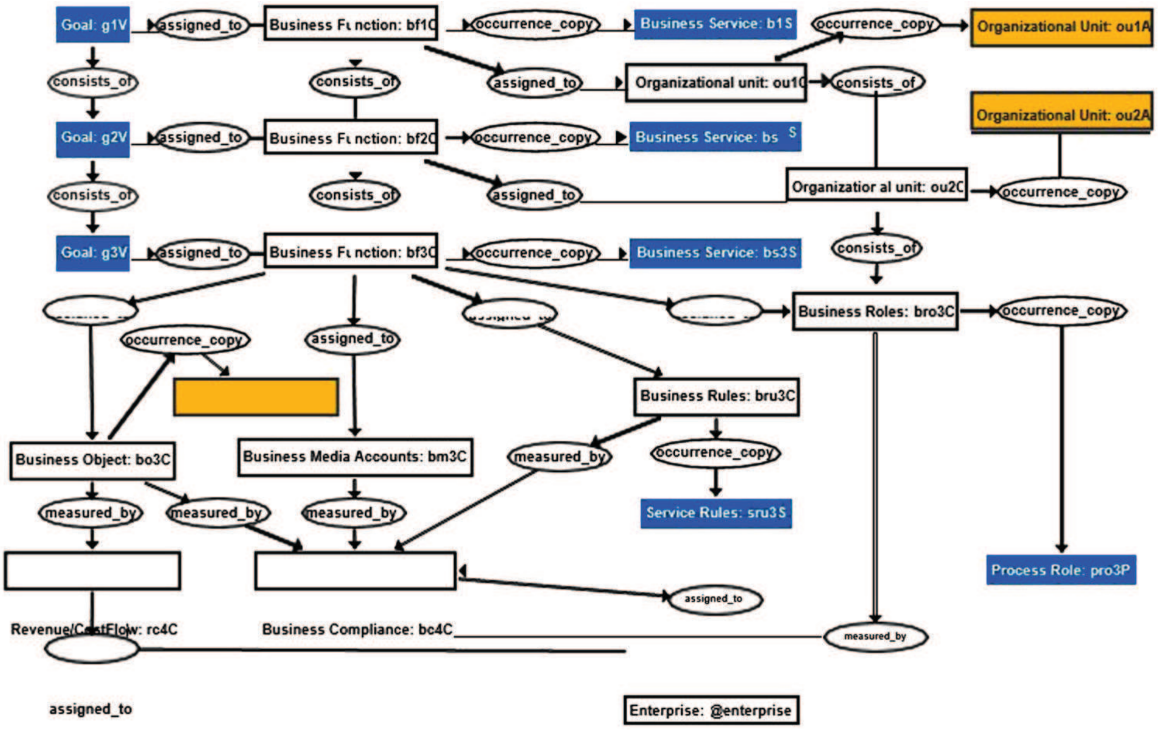
- A formal context is a triple K = (*G, M, I*), where *G* is a set of objects, *M* is a set of attributes, and *I GM* is a binary (true/false) relation that expresses which objects have which attributes.
- (*A, B*) is a formal concept precisely when:
    - Every object in *A* has every attribute in *B*,
    - For every object in *G* that is not in *A*, there is some attribute in B that the object does not have,
    - For every attribute in M that is not in *B*, there is some object in A that does not have that attribute.

To the uninitiated this may be confusing or a little too high-level; however, fuller explications of FCA with formal proofs and lucid worked examples can be found (Wolff, 1993; Ganter et al., 2005; Priss, 2006; Andrews et al., 2011)[5].

Competency

Figure 6 shows the CG for the Competency layer module. Some of the CG concepts in Figure 6 are shaded to highlight where they appear in the other layers. The shading scheme matches that shown by

Figure 6. Competency, GCs



the earlier EIMA overview Figure 3. Again, the same mapping through *CGtoFCA* is applied and Figure 7 shows the resulting FCL. This time [Enterprise:@enterprise] is not bottommost.

An inspection of the CG reveals that there are concepts, such as [Business Service: b1S], [Service Rules: sru3S], [Process Role: pro3P] that have their identical concept in another business layer (e.g. S for Service, P for Process). Likewise [Business Object: bo3A], and [Organizational Unit: ou1A] in A the Application information layer do not transitively end up at [Enterprise: @enterprise] unlike the Value CG Figure 6 above. It is therefore harder to discern that this a semantic unit; it had dependencies with the other layers that will only be resolved when the CGs from those other relevant layers are joined with this layer. If, together, a transitive path to [Enterprise: @enterprise] is discovered they are (interdependent) semantic units.

While a simple inspection of the CG for this layer without the FCL reveals the incomplete transitivity, in the combined form this would be harder especially if the CGs for all the layers are joined. Note also the FCL, which is computer generated rather than hand-drawn, horizontally lays out the meta-objects according to their levels – unless they are all not transitive to [Enterprise: @enterprise], thereby offering another highlight. Compare Figure 7 with Figure 5 for example.

Service

Figure 8 shows the Service layer CG, which highlights similar findings to that of Competency. It exhibits the same [Type-Label: Referent] and (relation) pattern as illustrated by Value and Competency. So as a

Figure 7. Competency, FCL

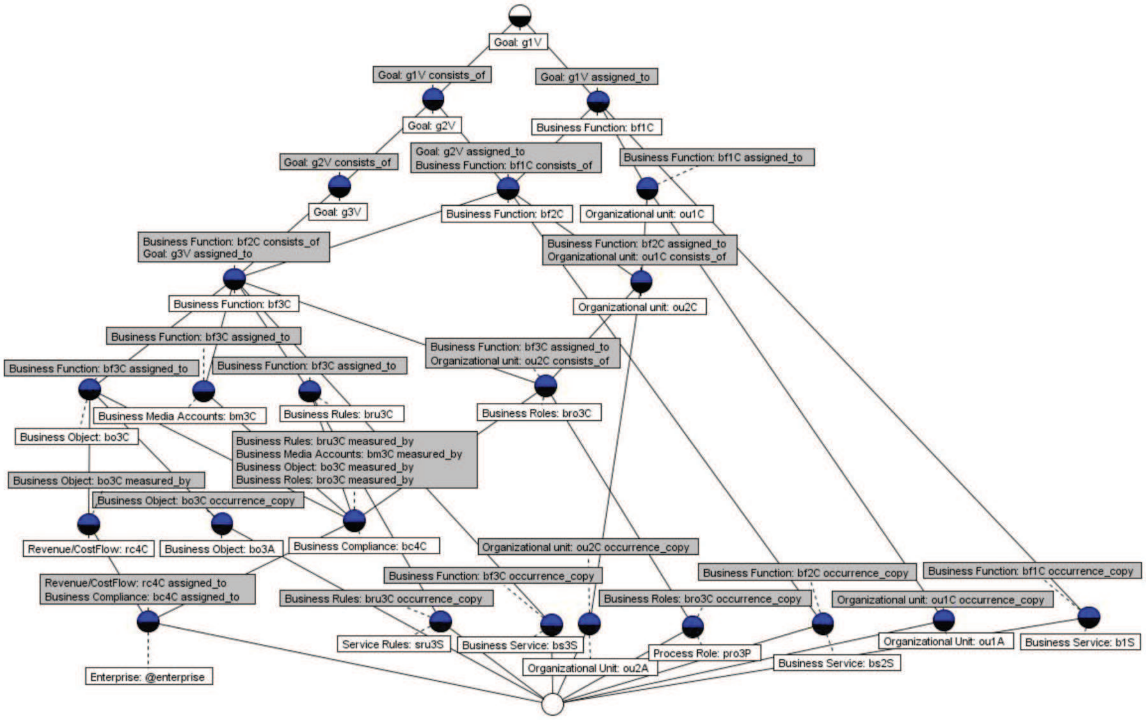
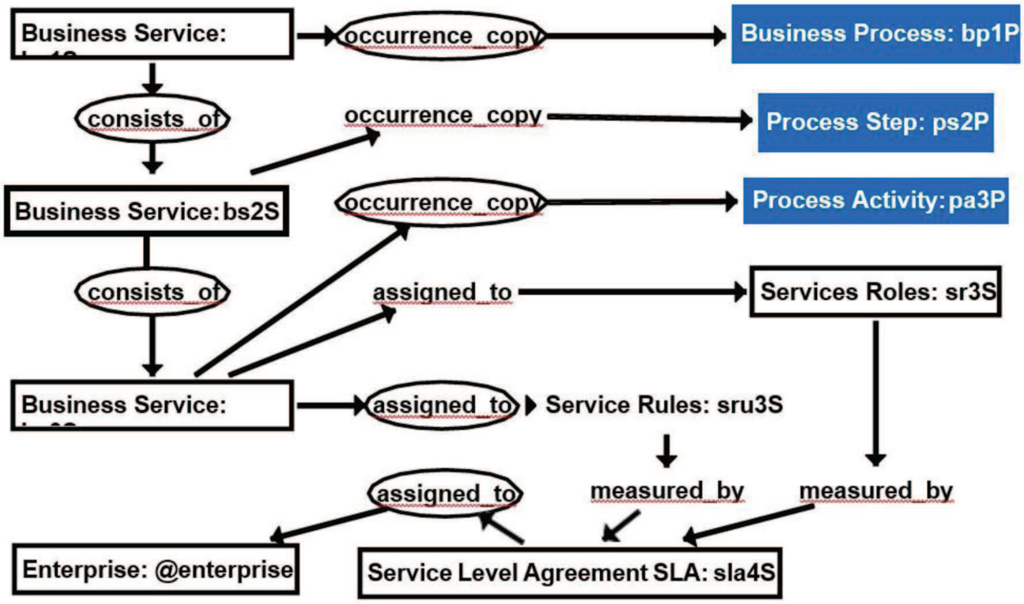


Figure 8. Service, CGs



reminder, bs1S in [Business Service: bs1S] for example, describes bs for Business Service, 1 is Level 1, and S is Service. Certain concepts are also shaded to highlight their occurrence in another layer.

In this layer, there is an (occurrence_copy) relation too. This relation occurred in Competency too but we'll use Service to remark on it further. Essentially, this relation describes two concepts (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[6]. The FCL generated by *CGtoFCA* for the Service layer is shown by Figure 9. 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.

Figure 9's FCL evidences that [Enterprise: @enterprise] again is not bottommost. Looking at the reason that we already know from Competency but from another perspective, this 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. Intent here is an FCA term that reading upwards from a given Formal Concept towards the top of the lattice shows all the attributes that the concept has. Thus, [Enterprise: @enterprise] – given all the other concepts in the layer (as in Value) transitively arrive to it – has *all* the attributes in the lattice. Therefore, it is clearly shown that [Enterprise: @enterprise] captures all the features (attributes) that make up the given layer and nothing is left out. Unless they are out of its intent, as evident in Competency and Service.

Process

The Process layer is described by Figure 10 for the CG and Figure 11 for the FCL. [Enterprise: @enterprise] again is not bottommost. By now the behaviour of the transformation of the CG to FCL using CGtoFCA and the associated tools described earlier should be self-explanatory.

Figure 9. Service, FCL

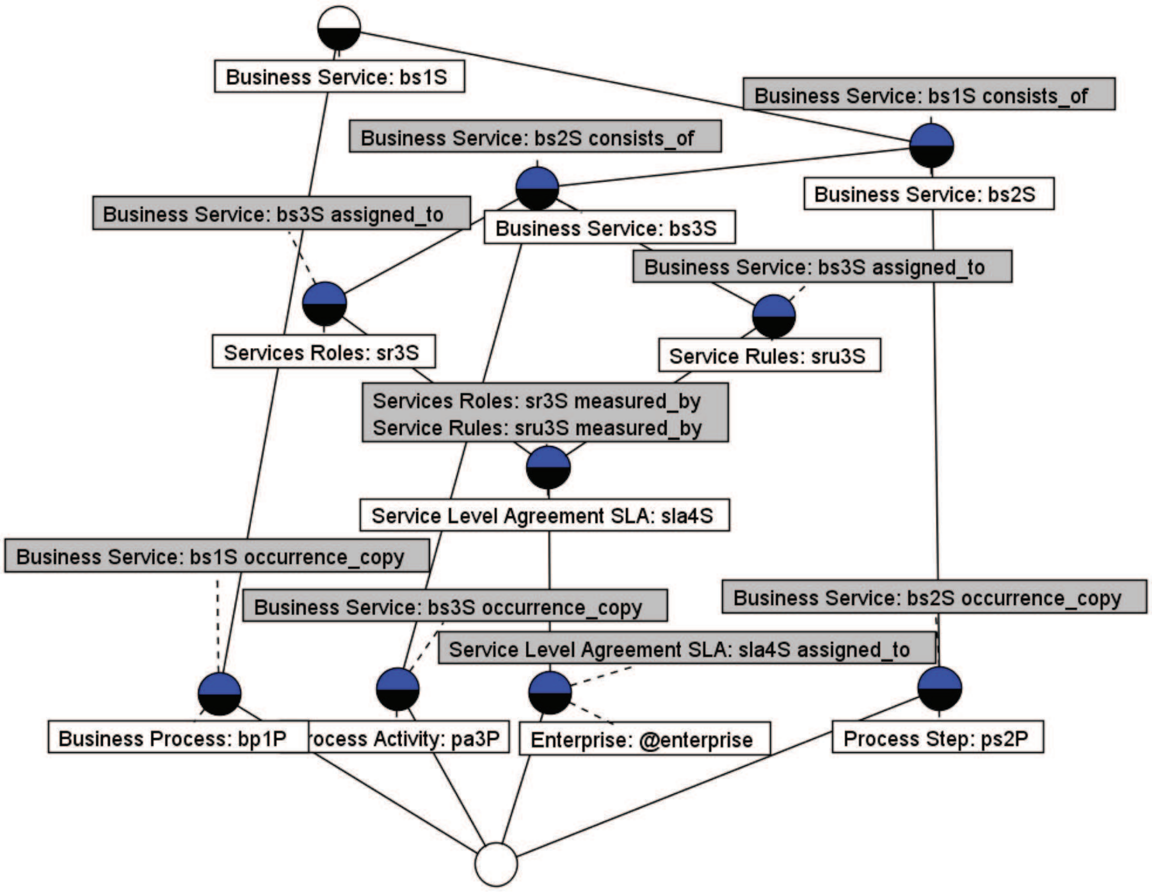


Figure 10. Process, CGs

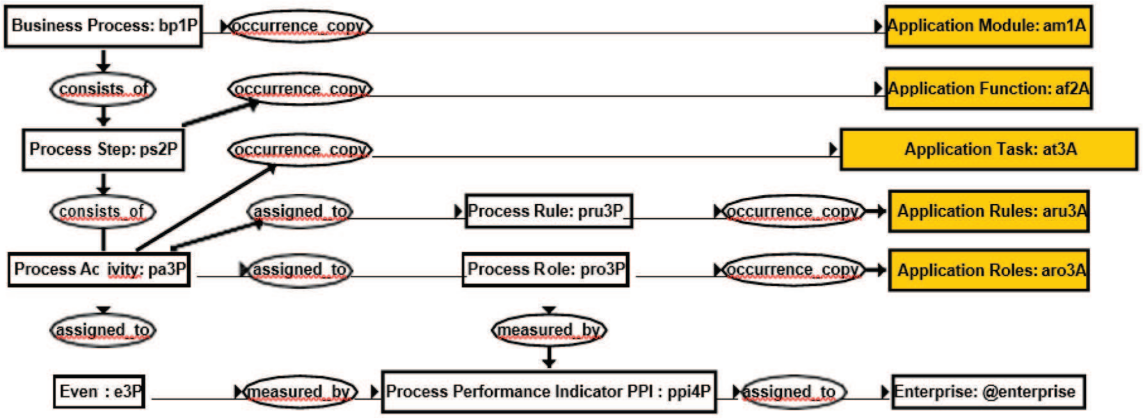
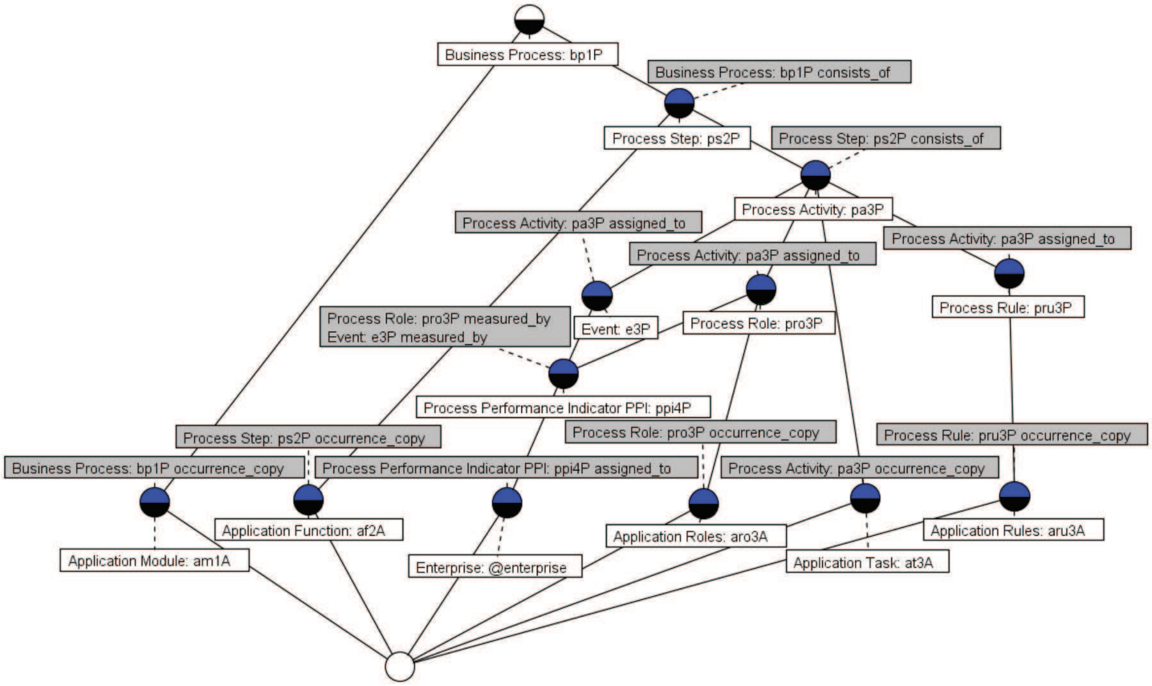


Figure 11. Process, FCL



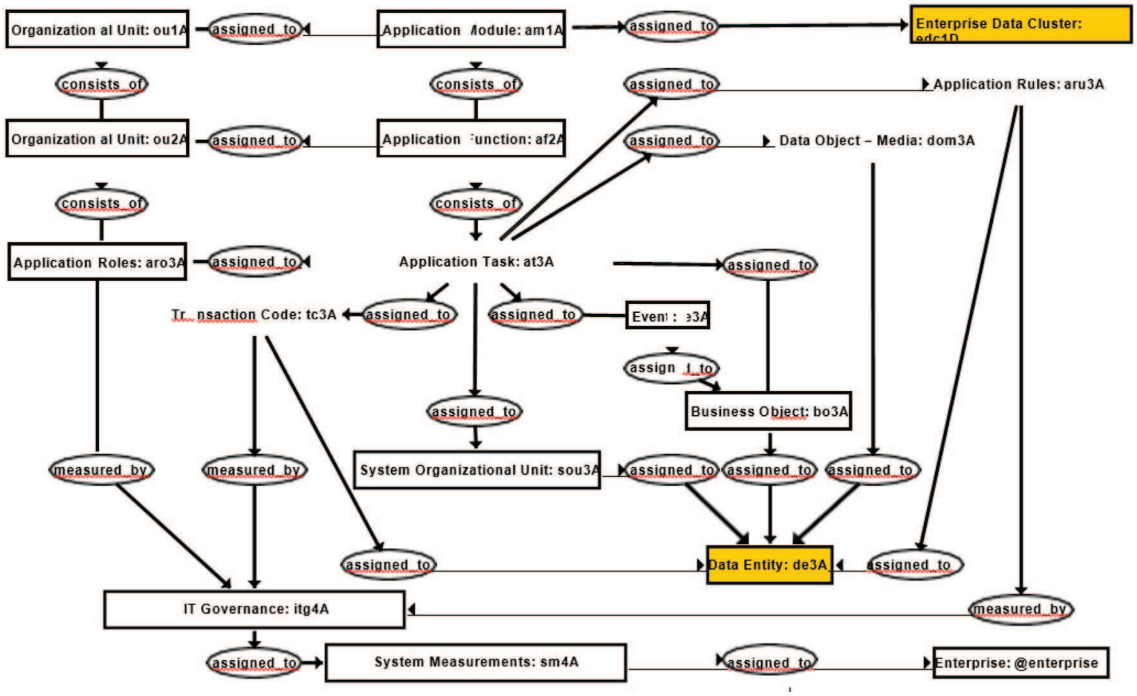
To aid our understanding however, we can consider FCA in more detail. In FCA, the bottommost concept is known as the *infimum* and the top most formal concept in a FCL is the *supremum*. In the Process layer the supremum is [Business Process: bp1P]. The FCA intent of a lattice through its attributes (e.g. Business Process: bp1P consists_of) has already been described; the *extent* is all the objects in the path of a given object down to the infimum. (The inverse of which is that the intent goes the other way up to the top of the lattice through the attributes to the supremum.) Thus, for example the extent of Process Activity: pa3P is Process Role: pro3P, Application Roles: aro3A, Application Task: at3A, Application Roles: aro3A, Process Rule: pru3P, Application Rules: aru3A, Event: e3P, Process Role: pro3P, Process Performance Indicator PPI: ppi4P, and Enterprise: @enterprise. Enterprise: @enterprise is *not* however in the extent of Application Roles: aro3A, Application Task: at3A, Application Roles: aro3A. In this layer as it stands these concepts (meta-objects) do not extend to the enterprise, when they ought to be given their expected impact on it!

The Information Systems Layer

Application

Figure 12 depicts the Application layer CG. Figure 13 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 [System Measurements: sm4A], which evaluates the Application layer. Also emerged in the middle of the lattice is another formal concept without its own

Figure 12. Application, CGs



object. So far this has only occurred at the infimum, with Enterprise: @enterprise. We can follow the intent and extent from and to this concept for example, to:

- Get a sense of what name we might give this meta-object,
- Identify a structural issue in this layer, or
- Confirm that it's simply warranted, and left simply without a name.

It thus reveals a focus for further investigation.

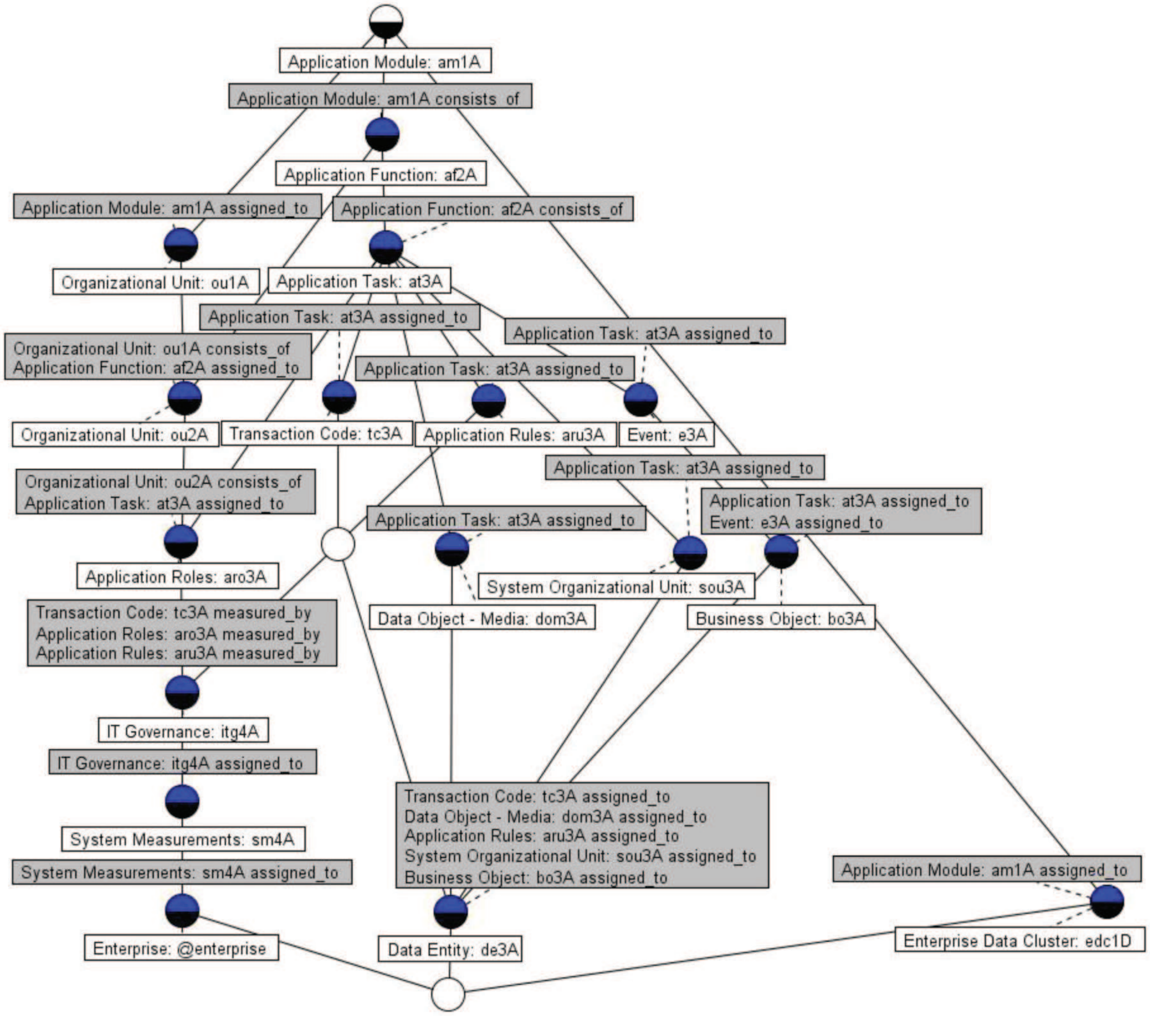
Data

Figure 12 depicts the Application layer CG. Figure 13's FCL evidences that [Enterprise: @enterprise is bottommost i.e. at the infimum. Like Value, the extent of all the attributes from the topmost formal concept i.e. the supremum is [Enterprise: @enterprise] including from all the relevant KPIs (level 4 meta-objects) including [System Measurements: sm4A]. In this layer as it stands all its concepts (meta-objects) extend to the enterprise, demonstrating that they all impact on the enterprise as expected!

A MODULARISED, HOLISTIC META-MODEL

While most of the individual modules do not have [Enterprise: @enterprise] as their bottommost formal concept, modularising the EIMA makes them more readable by a human reviewer. Given they are (or

Figure 13. Application, FCL



should be) semantic units also reinforces their need to be modular, rather than in one heterogeneous mass. However apart from Value and Data, the layers are not evidently semantic units without resolving their interdependencies with the other layers. Therefore, the next stage is to combine the modules according to their co-referent links thereby discovering that when so combined whether [Enterprise: @ enterprise] emerges to be bottommost or not. Through CGs join operation the co-referent links enable the CGs for each layer to be joined into one, large CG. When that joined CG is passed to *CGtoFCA*, the resulting FCA is shown by Figure 16.

The attributes and objects are not in this figure for convenience, but it shows (although not labelled for this reason) that [Enterprise: @enterprise] remains at the infimum (bottommost). That clarifies Competency, Service, Process and Application as se mantic units. Also noticeable are other formal concepts that do not have their own meta-objects in the lattice, and as before warranting further investigation as discussed above for the Application layer.

Figure 14. Data, CGs

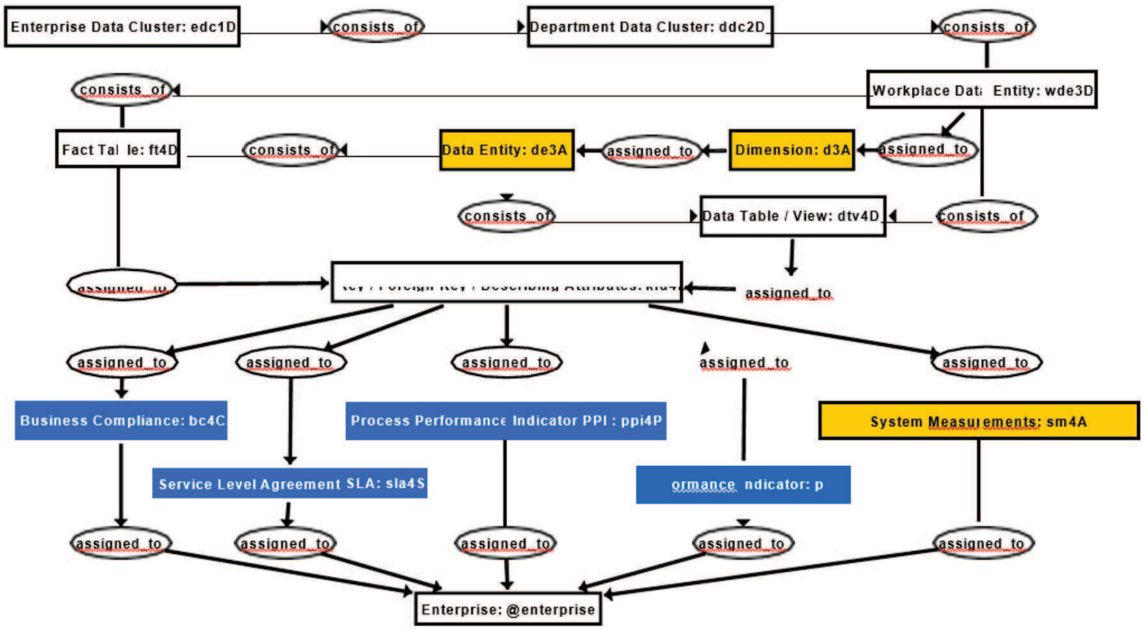


Figure 15. Data, FCL

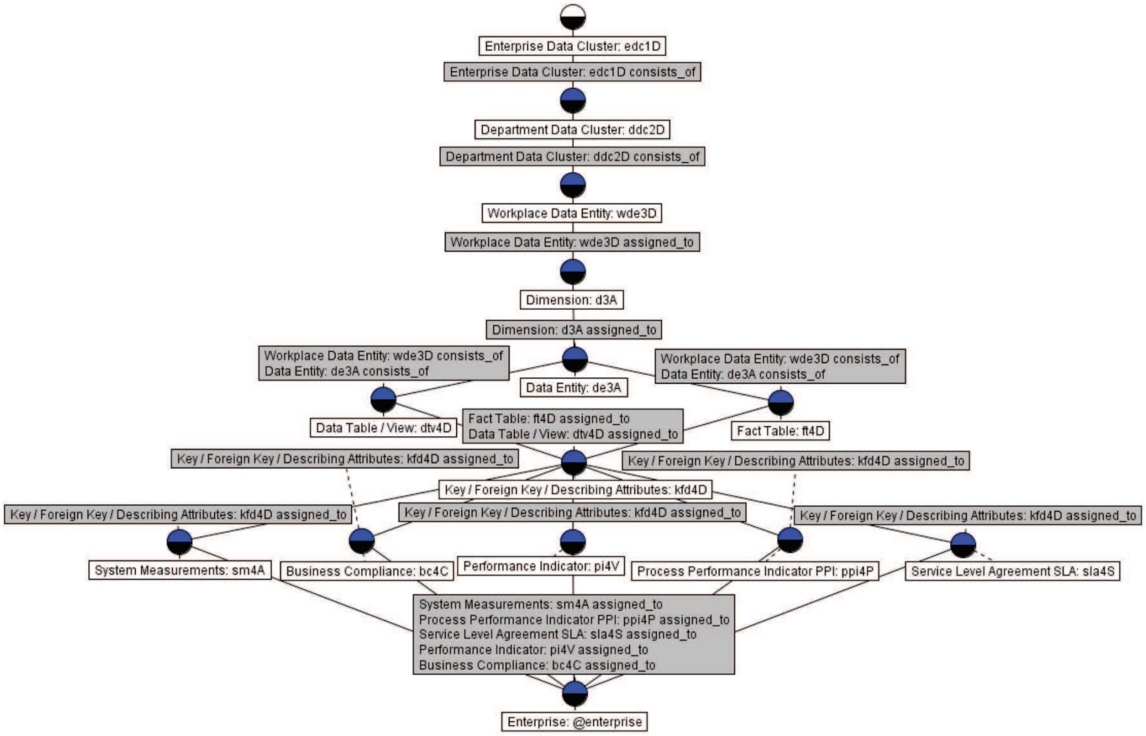


Figure 16. Combined EIMA, FCL

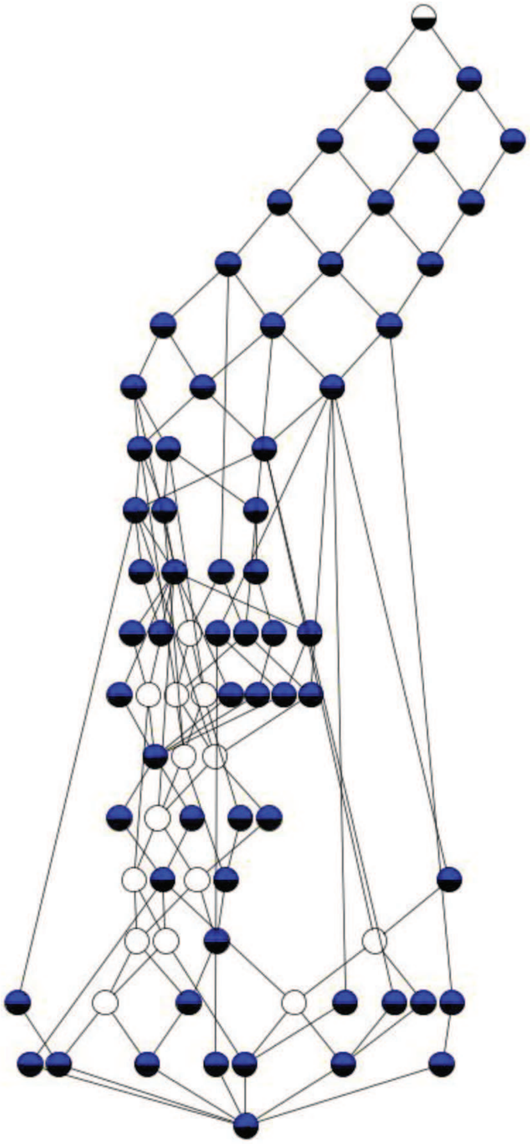
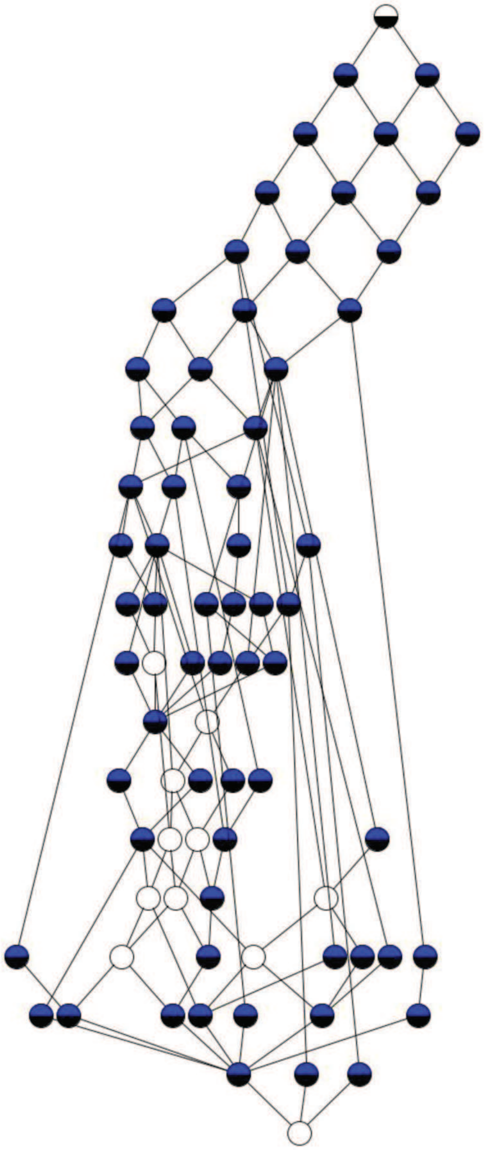


Figure 17. Combined EIMA with disjoin, FCL



What's happens if a co-referent doesn't join? That might be because there is a disagreement between the human modelling team for one layer viewing a given concept as having a different meaning thus not harmonised across the whole meta-model. We can reflect this discord by simply giving it a different co-referent. For example [Organizational Unit: not-ou1A] in the Competency layer and [Organizational Unit: ou1A] in the Application layer have a different referent because the modellers for each respective layer do not (currently) agree they are the same meta-object even though they share the same name[7]. Also, in this example there is disagreement over [Organizational Unit: not-2A] in the Competency layer and [Organizational Unit: ou2A] in the Application layer. This latter example is compounded by a typo i.e. not-2A in the Application layer should be not-ou2A following the (mis)naming convention in this example?

In any event, a differing FCL results as Figure 17 demonstrates. Not surprisingly, [Enterprise: @ enterprise] is no longer at the infimum. As [Organizational Unit: ou1A] and [Organizational Unit: ou1A] no longer have semantic relations directed from them to a target concept, they are the other two formal concepts along with [Enterprise: @enterprise] directly above the infimum. Figure 18 shows an extract of the CGs involved and Figure 19 these three formal concepts and where they are situated in the FCL.

From a simple visual inspection Figure 17's shape has also altered from Figure 16 including the formal concepts without their own meta-objects, suggesting other impacts of the discord. We can in any event see how Figure 17's altered structure captures the nature of the discord i.e.:

Figure 18. Combined EIMA CGs Extract, FCL

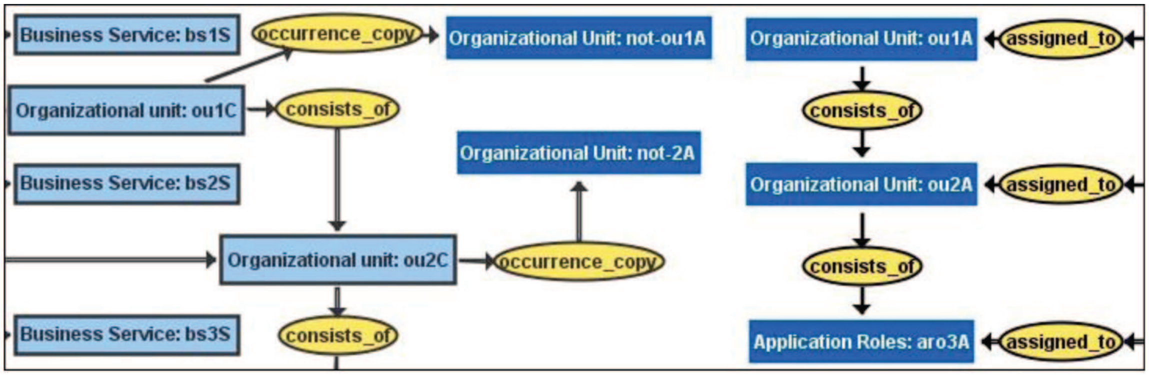
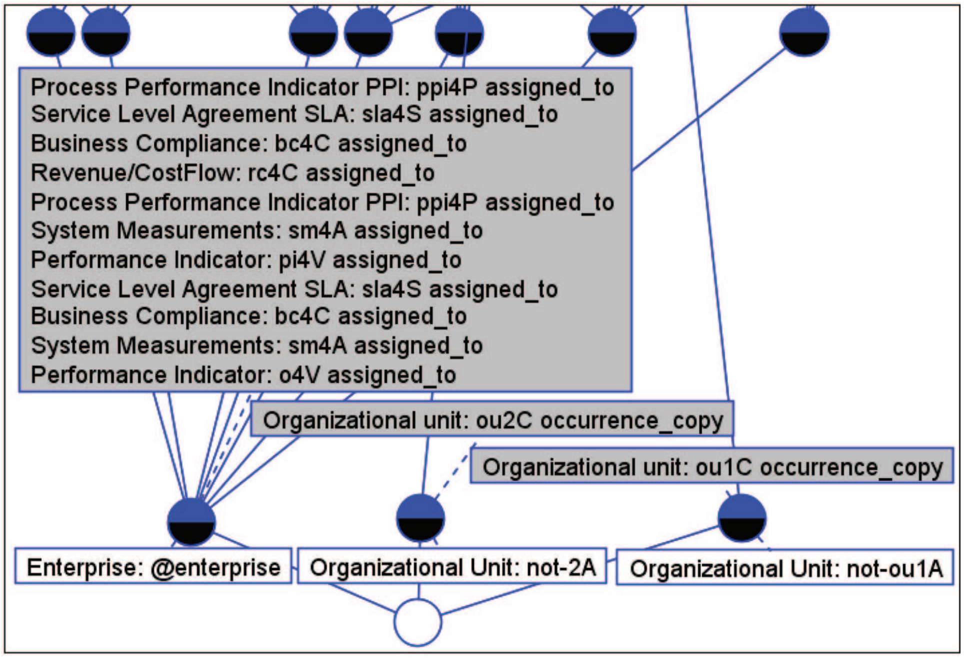


Figure 19.



1. *Syntactic* (the typo),
2. *Semantic* (the different meanings of the same meta-object), and
3. *Pragmatic* (the process of sharing meaning that in this case hasn't been achieved yet).

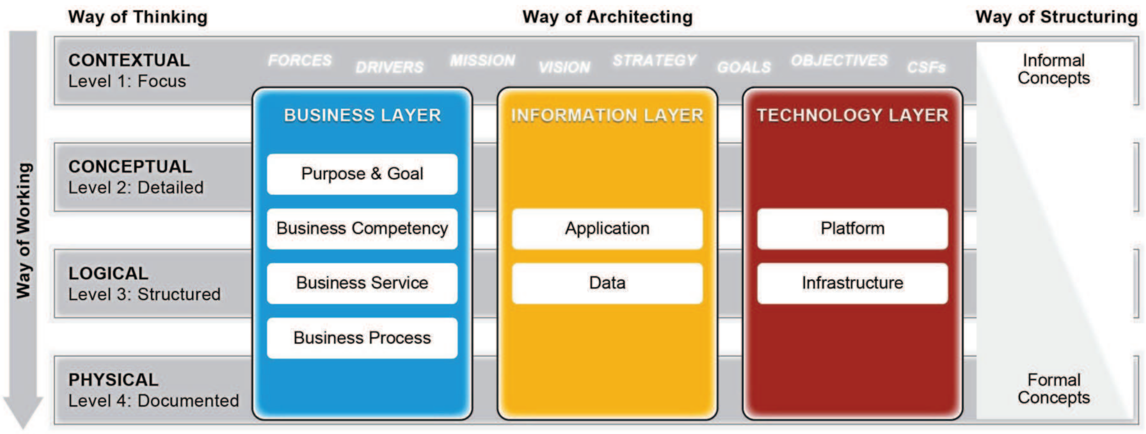
Towards the beginning of this paper we pointed out that semantics are an aspect of semiotics, like syntax, which distinguishes valid from invalid symbol structures, and like pragmatics, which relates symbols to their meaning within a context e.g. the community in which they are shared (Cordeiro & Filipe, 2004). We also brought in Conceptual Structures, which Sowa describes as "Information Processing in Mind and Machine" (Sowa, 1984). FCA adds a mathematical dimension to logic, depicted by the CGs. The productivity of the computer through *CGtoFCA* augments the human creativity from which enterprises emerge. That contextual way of thinking as it traverses though the conceptual, logical and physical way of working is conceptually structured through the interplay between informal and formal concepts. Figure 20 depicts this added dimension to the earlier Figure 2.

Whilst focusing on EIMA for the purposes of our discussion, there is the potential for our approach to be applied as a general vehicle for harmonising meta-models (Henderson-Sellers, 2012). Formal concepts can pinpoint the disharmonies ranging from the simply syntactic cases, the meaning-driven semantic cases and – eventually – sharing meaning (pragmatcs) based on where the differences actually lie.

CONCLUSION

Using EIMA as the illustration, we have portrayed how the layers-and-levels meta-models of the GUA and other bodies adopting this approach can be enhanced by Formal Concepts. The use of CGs (Conceptual Graphs) and FCA (Formal Concept Analysis) through the *CGtoFCA* algorithm provide a rigorous unification of modularised meta-models. That included the validation of each layer (module) as semantic units where they have to have the necessary interdependencies with other layers (modules). In EIMA's case, through the co-referent links it revealed the cross-layer levelling of its information content specific meta-objects in its Business and Information Systems layers. As such it also revealed the syntax, semantics

Figure 20. The layered enterprise way of working, conceptually structured



and pragmatics in the levels (Contextual, Conceptual, Logical and Physical) that in EIMA's case is the Enterprise, Department, Workplace, and Document (with its associated Key Performance Indicators).

The integration of a technical layer in the next version of EIMA will also apply the approach taken in this paper, so that it is better integrated at the outset. Moreover, EIMA will be rolled back into the overarching GUA meta-models. For historic and expediency reasons, there has been a little divergence of EIMA from GUA's (thus LEADing Practices') meta-models. Again, the approach described in this paper will ease that process, bringing together EIMA's valuable experience with SAP's GBI set of case studies with the developments that have since happened with GUA's meta-models. Naturally this harmonisation can extend into the meta-models of other standards or recommendations bodies such as the OMG, the Open Group, ISO, Web and others[8].

The findings of this paper may also outline mismatches in the "supporting work products" by offering support mechanisms from the highest contextual level to a system design. The resulting identification of any such gaps in the physical layer is beneficial to system builders so as to prevent running into unforeseen interoperability issues during implementation. In our vision, the mathematical interpretation through formal concepts that are enacted by the computer support the transition into the physical layer along with the informal concepts that characterise existing approaches. Essentially there a rich interaction between the computer and human modeller or designer in reconsidering their CG (Conceptual Graph) models from the FCL (Formal Concept Lattice). It thus acts as a supporting tool to the logical and other layers, and actualises the modularising of complex meta-models in enterprise systems using conceptual structures.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Global University Alliance for this work, as well as access to the practitioner resources of LEADing Practice including its CEO Henrik von Scheel. Acknowledgements also to the SAP University Alliances program, particularly Stefan Weidner of the SAP University Competence Center & School of Computer Science, University of Magdeburg, Germany.

REFERENCES

Andrews, S., Orphanides, C., & Polovina, S. (2011). Visualising Computational Intelligence through Converting Data into Formal Concepts. In Next Generation Data Technologies for Collective Computational Intelligence (pp. 139–165). Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-20344-2_6

Andrews, S., & Polovina, S. (2011). A Mapping from Conceptual Graphs to Formal Concept Analysis. In Conceptual Structures for Discovering Knowledge, the 19th International Conference on Conceptual Structures, LNCS (Vol. 6828, pp. 63–76). Springer. doi:10.1007/978-3-642-22688-5_5

Bork, D. (2015). *Development Method for the Conceptual Design of Multi-View Modeling Tools with an Emphasis on Consistency Requirements* [Doctoral Dissertation]. University of Bamberg, Germany.

Bork, D., Buchmann, R., & Karagiannis, D. (2015). Preserving multi-view consistency in diagrammatic knowledge representation. In *Proceedings of the International Conference on Knowledge Science, Engineering and Management* (pp. 177–182). Springer.

Cordeiro, J., & Filipe, J. (2004). The semiotic pentagram framework – a perspective on the use of semiotics within organisational semiotics. In *Proceedings of The 7th International Workshop on Organisational Semiotics*.

Floyd, R. W. (1967). Assigning Meanings to Programs. In *Mathematical Aspects of Computer Science* (Ch. 19, pp. 19–32). American Mathematical Society.

Ganter, B., Stumme, G., & Wille, R. (2005). Formal Concept Analysis: Foundations and Applications. Springer.

Group, T. O. (2011). 34. content metamodel. Retrieved from http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap34.html

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, *43*(5), 907–928.

Henderson-Sellers, B. (2012). Standards harmonization: Theory and practice. *Software & Systems Modeling*, *11*(2), 153–161. doi:10.1007/s10270-011-0213-0

Malik, N. (2009). Why the Zachman framework is not an ontology.

Mayall, A. and Carter, J. (2015). The essential project: Harnessing conceptual structures to expose organizational dynamics. *International Journal of Conceptual Structures and Smart Applications*, *3*(2), 1–11.

Oberle, D. (2013). *How ontologies benefit enterprise applications*. Semantic Web Journal.

Polovina, S. (2007). An introduction to conceptual graphs. In *Proceedings of the 15th International Conference on Conceptual Structures: Knowledge Architectures for Smart Applications ICCS '07*. Springer-Verlag.

Polovina, S., Scheruhn, H.-J., Weidner, S., & von Rosing, M. (2016a). Discovering the gaps in enterprise systems via conceptual graphs & formal concept analysis. In *Poster Proceedings, the 22nd International Conference on Conceptual Structures (ICCS)* (pp. 5–8).

Polovina, S., Scheruhn, H.-J., Weidner, S., & von Rosing, M. (2016b). Highlighting the gaps in enterprise systems models by interoperating CGS and FCA. In S. Andrews & S. Polovina (Eds.), *Proceedings of the Fifth Conceptual Structures Tools & Interoperability Workshop (CSTIW '16)* (pp. 46–54). CEUR-WS.

Priss, U. (2006). Formal concept analysis in information science. *Annual Review of Information Science & Technology*, *40*(1), 521–543. doi:10.1002/aris.1440400120

Scheruhn, H., Fallon, R., & Rosing, M. (2015). Information Modelling and Process Modelling. In The Complete Business Process Handbook (Vol. 1, pp. 511–550). Elsevier.

Scheruhn, H.-J., Ackermann, D., Braun, R., & Förster, U. (2013). Human-Computer Interaction. Users and Contexts of Use. In *Proceedings of the 15th International Conference, HCI International 2013* (pp. 446–455). Springer Berlin Heidelberg, Berlin, Heidelberg.

Scheruhn, H.-J., Rautenstrauch, C., Pegnetter, R., and Weidner, S. (2006). Strategische ausrichtung eines internationalen masterprogramms mit dem schwerpunkt integrationskompetenz am beispiel von mysap. *Die neue Hochschule DNH*.

Sowa, J. F. (1984). *Conceptual Structures- Information Processing in Mind and Machine*. Addison-Wesley.

Sowa, J. F. (2008). Conceptual Graphs. In Handbook of Knowledge Representation, Foundations of Artificial Intelligence (Vol. 3, pp. 213–237). Amsterdam: Elsevier.

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, *31*(3), 590–616. doi:10.1147/sj.313.0590

von Rosing, M. (2016). What are artefacts & how can they be used. Retrieved from http://www.leadingpractice.com/knowledge-center/recorded-webinars/what-are-artefacts-2/

von Rosing, M., & Laurier, W. (2015). An introduction to the business ontology. *International Journal of Conceptual Structures and Smart Applications*, *3*(1), 20–41.

von Rosing, M., Urquhart, B., & Zachman, J.A. (2015). Using a business ontology for structuring artefacts: Example- northern health. *International Journal of Conceptual Structures and Smart Applications*, *3*(1), 42–85.

von Rosing, M., & von Scheel, H. (2016). Using the business ontology to develop enterprise standards. *International Journal of Conceptual Structures and Smart Applications*, 4(1), 48–70.

Zivkovic, S., & Karagiannis, D. (2015). Towards metamodelling-in-the-large: Interface-based composition for modular metamodel development. In *International Conference on Enterprise, Business-Process and Information Systems Modeling* (pp. 413–428). Springer.

Wolff, K. E. (1993). A first course in Formal Concept Analysis. In Proceedings of StatSoft '93 (pp. 429–438). Gustav Fischer Verlag.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal*, *26*(3), 276–292. doi:10.1147/sj.263.0276

Zachman, J.A. (2015). John Zachman's concise definition of the Zachman Framework.

Zhao, F., Scheruhn, H.-J., & Rosing, M. (2014). Human-Computer Interaction. In *Applications and Services* (pp. 776–785). Cham: Springer International Publishing.

ENDNOTES

[1] GUA (www.globaluniversityalliance.net) is a non-profit body consisting of over 450 universities, professors and researchers.

[2] www.leadingpractice.com

[3] http://uac.sap.com

[4] The CGs are drawn in CharGer (http://charger.sourceforge.net/) as it has support for the ISO/IEC24707 CGIF (CG Interchange Format). At http://www.jfsowa.com/cg/cgdpansw.htm there is

more information on the standard. Concept Explorer (http://conexp.sourceforge.net/) was used to generate the FCL.

[5] On this occasion https://en.wikipedia.org/wiki/Formal_concept_analysis is also a good starting point.

[6] In passing it is worth remarking in the ARIS software (www.aris.com) that is used to model GBI, these occurrences are the same object, so in that sense the problem may appear to go away. But it doesn't as they are 'pseudo-synonyms', evidenced by them not being co-referent semantically.

[7] To clarify, EIMA itself would not be 'broken' in this way; rather it would be some derivative of this or any other well-formed meta-model for a particular enterprise context.

[8] www.omg.org,www.opengroup.org,www.iso.org, www.w3c.org