Facilitating Design and Use of Effective Visual Languages in Enterprise Modelling and Information Systems

Graham McLeod

inspired.org, South Africa & University Duisburg-Essen, Germany
graham@inspired.org

Supervisor: Dr Prof Ulrich Frank, University Duisburg-Essen

Abstract. Enterprise modelling and information systems work often relies heavily on graphical models expressed in visual languages to concisely capture, rigorously model and effectively convey meaning between stakeholders. Recent research has highlighted problems with the effectiveness of popular modelling notations. A physics of notations (PoN) was proposed to address these issues. Application of the PoN has not proven routinely successful. Models are often constructed by experts, but must be well received by non-experts to achieve their goals. This research contends that recent information from the fields of cognition, visualisation and graphic design can be exploited to enhance the return on modelling effort (ROME) and the value of models. Improved meta models, methods for visual language design and enhanced tools can support the definition and use of effective visual languages and the application of the PoN and derivatives.

Keywords: Graphical Modelling, Enterprise Modelling, Polymetric Modelling, Return on Modelling Effort (ROME), Meta Model, Concrete Syntax, Visual Language, Physics of Notation (PoN).

1 Research Problem

1.1 Area of Interest

Business, public and social enterprises are experiencing accelerated and more significant change than previously [1]. They are also more information systems (IS) and technology intensive to remain competitive and to achieve their aims [2, 82, 83].

J. Ralyté and Y. Wautelet (Eds.): PoEM 2018 Doctoral Consortium Papers, pp. 15-30, 2018.

A variety of techniques assist in enterprise transformation and the delivery of the necessary information technology (IT) and system solutions. These include enterprise modelling (EM), enterprise architecture (EA), requirements engineering (RE) and solution architecture.

EA methods recommend the use of many different forms of models, many graphical. These frameworks include (*inter alia*): Zachman [3], the IFIP Generalised Enterprise Reference Architecture and Methodology (GERAM) [4], the U.S. Department of Defence Architecture Framework (DODAF) [5], The US Government's Federal Enterprise Architecture Framework (FEAF) [6], Cap Gemini's Integrated Architecture Framework (IAF) [7], The Open Group Architecture Framework (TOGAF) [8], the Pragmatic EA Framework (PEAF) [9]. Archimate [10] has been developed specifically to define suitable graphical models to represent EA models.

There are other approaches in the realm of EM and IS analysis. These include (*inter alia):* Multi-perspective Enterprise Modelling (MEMO) [11], Enterprise Engineering (EE) [12] and the Unified Enterprise Modelling Language (UEML) [13].

Requirements Engineering (RE) approaches include (*inter alia)*: Integration DEFinition (IDEF) [14], Structured Systems Analysis/Design (SSA/D), Information Engineering (IE) [15] and Requirements Engineering Board (REB) [16]. These typically address goals, functional, process, service, capability, data, interface and non-functional requirements. They make use of a variety of notations, including (*inter alia*) e.g. IDEF, Unified Modelling Language (UML) [17], Entity Relationship (ER) Models [18], IE Models, Process Models e.g. Business Process Model and Notation (BPMN) [19], Goal Models [20, 11], and Solution Architecture (component/interface) models [21]. Software engineering methods are addressed by graphical methods too e.g. ISO 24744 [85].

Despite the very widespread use of graphical notations in support of enterprise modelling and information systems work, there is often a shortfall in delivery of value versus expectations. This was highlighted by [22] in a paper about Return on Modelling Effort (ROME). Significant effort is expended on the building of models, but the value anticipated may not be realised if the models are inappropriate to the task, unsuitable for the audience or of poor quality.

1.2 Research Problem

Graphical models can definitely enhance modelling effectiveness, but in over 35 years of analysis, development, consulting, strategy, project management and architecture practice in industry, the author has observed first hand that there are also difficulties in achieving value delivery from visual models. While these almost always add understanding for the author and rigour to the analysis and eventual solution, it is often difficult to get acceptance and "buy in" from the non-specialist audience, which typically includes executives, project sponsors and domain experts. Problems which we have observed include the following:

P1 Models and their presentation are not properly matched to stakeholder interests, concerns or expertise

P2 Models are too "technical" - typically meaning that the audience does not easily comprehend the collection of symbols and connections and their meaning. This can be exacerbated by poorly designed or overloaded notations

P3 Models are too homogenous - We refer to "camouflage models" where there are a great many elements that are essentially the same, thus making it difficult to extract anything of significance. An example would be the BAIN Service reference model for banking [23], where there is a nested diagram showing several hundred services just as text names. Another would be a large entity relationship diagram for a domain data model showing several hundred entities and many more relationships

P4 Models are not in a familiar format - Business stakeholders typically relate to documents, presentations and spreadsheets and find it difficult to extract meaning rapidly from an unfamiliar graphical presentation

P5 Models have too many elements without adequate grouping, layering or differentiation

P6 Practitioners spend inordinate effort transposing models from rigorous tools to non-structured formats to overcome the acceptance problem. In the process, the connection to the source repository is lost, thereby destroying integrity, reusability, maintainability and currency of the derived output when the source changes

P7 It may be difficult to get the answers required from the visual models even though the necessary data is in the underlying repository

There are additional challenges, which include the need to support: progressive completion of models as information becomes available; collaborative construction of models by multiple stakeholders; reuse of elements across models and organisational entities; and security and privacy of selected information. These problems can be addressed by improved modelling language and graphical notation design, as well as carefully conceived tooling.

1.2 Research Questions

The overall research question can be formulated as:

> How can we improve the design of modelling languages, their graphical notations and supporting tools to resolve current problems and enhance

their effectiveness in enterprise modelling, information systems and requirements engineering?

This can be decomposed into several sub-questions:

- RQ1 What kinds of visual representation / language are used in enterprise modelling and information systems (abstraction from practice)?
- RQ2 What aspects distinguish effective visual language from ineffective visual language?
- RQ3 What inhibits the effective use of visual language design/usage principles and guidelines in practice?
- RQ4 What aspects of human cognition, perception and information processing can we better exploit to enhance visual languages and their use?
- RQ5 What changes or new capabilities are required in meta models and tools to enhance their support for effective visual language design and use?
- RQ6 What guidance can be provided to designers of visual modelling languages / notations?

2 Prior Work

Many disciplines and theories are relevant to the current study. We will briefly review these and relevant prior contributions within each.

Stakeholder orientation is important to cater for the needs and concerns of authors and model users. Some appropriate techniques are detailed in TOGAF [8].

Semantics is the study of meaning. (from Ancient Greek: σημαντικός sēmantikós, "significant") - Wikipedia. It focusses on the meaning and significance of the knowledge conveyed, rather than the form of presentation. Semantics is important as models are primarily about capturing, organising, analysing and sharing information, knowledge and understanding. A proper semantics ensures that the concepts are appropriate to the goals, understood by the stakeholders and relevant to achieving the purpose of the models. [24] discuss the need for formal semantics in modelling languages. Various approaches are available for defining semantic models. Popular approaches include: the Integration DEFinition for Information Modeling (IDEF1X) , Resource Description Framework (RDF) and The Web Ontology Language (OWL) [25].

Semantic models are capable of describing concepts, relationships, properties and values for virtually any domain or type of model. They are thus suitable as a means to hold the content of models which will be represented using graphical models, as specified by a visual language.

Syntax is used in natural language and grammar to denote the way in which words can be combined to convey meaning. It is used within logic to define the rules for the combination of symbols and operations in legal constructions. It is used in computer science to denote the way in which programming language constructs can be legally combined into valid operations statements or data definitions. [26]. For graphical modelling we talk about concrete syntax [27], meaning the way in which symbols are used to represent concepts, relationships and values and the way in which they may be combined to convey the intended meaning. This is obviously of vital and central interest to our current study since it fundamentally influences the way in which meaning is perceived by the users of graphical models. It is also important to be able to generalise the concepts, relationships, properties and features that are necessary to be able to define visual languages so that these can be supported in tooling. Syntax should be related to semantics to connect the visual representations to the underlying meaning in the model.

Semiotics is the use of symbols to convey meaning. "semiotics, noun: the study of signs and symbols as elements of communicative behaviour; the analysis of systems of communication, as language, gestures, or clothing" - [26]. Graphical models, in particular, make use of a great many symbols in their notations. Some symbols are apprehended naturally based upon evolutionary mechanism of perception. These typically relate to physical entities in the real world, for which the human visual system has been optimised through evolution. These are called "Sensory Codes" and are perceived quickly and with little effort - in effect within the "human hardware". Other "arbitrary codes" are learned through experience or instruction and have to be "decoded" in the brain - in effect through sequential circuitry and "human software", a much slower and more arduous process.

Communications Theory as initially expounded by [28] is relevant to the transfer of ideas from the model author to a model reader. It includes the objective of accurate transfer of message or meaning, via a medium. It has important implications for factors which can inhibit effective communication (e.g. the presence of *noise* which is distracting signal or information which can cause errors at the receiver, or waste capacity).

The related fields of the **Human Visual System, Perception and Cognition** have developed rapidly in recent years. An excellent summary is provided by [29]. We have learned that there is a rapid, broadband, parallel processing initial stage of the visual system that takes in a wide angle view of the visual field and is optimised to find meaningful or interesting elements within that to focus on. Once these are identified, the active process of perception is triggered. This directs attention to the identified elements of interest in the visual field. These are imaged using the much higher resolution fovea in the eye. This sensory input is then processed in a much slower, sequential cognitive system.

Perception is an active process, where further attention is directed to interesting elements in the field of view [30] [31]. This occurs before we are consciously aware of the elements. There is evidence regarding the limitations of the visual short term memory that can provide guidance on how complex visual representations can be before they fatigue a viewer and information is lost.

[32] introduces the concept of **Information Encoding**, identifying 8 variables onto which information can be encoded graphically. The visual variables constitute a kind of visual alphabet for constructing notations. Visual language (notation) designers can use combinations of the variables to create the symbols of the desired language.

The **Physics of Notation** (PoN) was defined by [33] towards a scientific basis for constructing visual notations in Software Engineering. We argue that most of the principles are equally relevant for enterprise modelling. His approach leverages information coding to derive a prescriptive theory for the construction of a visual language. Principles are defined which a language should meet to ensure maximum effectiveness and ease of successful interpretation by a viewer. The PoN has been widely applied, although not always well [34]. Limitations are identified by [34], [35]. Some suggested enhancements are provided by [36] and [37]. An alternate approach is espoused by [86] and the Cognitive Dimensions of Notations adherents. A literature review is provided in [38]. In our work, we aim to improve/extend the theory to allow for adaptation to meet analysis goals, using the idea of Polymetric Diagramming.

Polymetric Diagramming is a technique introduced by [39] for the visualisation of complex software systems. The symbols in regular graphical models are modified in multiple ways (using encoding dimensions) to highlight properties of interest of the model elements. For example, the symbol height, width, colour or other properties may be changed. The technique exploits **Pre-attentive Processing** which allows a viewer to rapidly focus on relevant information in a large model using early cognitive processing which is parallel, rapid and low effort. [40]. We aim to adapt the ideas for use with a wider range of models including those for enterprise modelling and more general information systems work. We plan to provide empirically grounded principles for the effective use of the techniques as well as caveats for what should be avoided.

We live in a **highly visual age** and most citizens nowadays are bombarded with a huge volume and variety of information [41]. There is increased competition for attention from all sides, including advertising, news, social media, personal messaging, eMail and other channels [42]. Viewers of models are highly visually literate and now demand much better input to attract and hold their attention. [43] highlight the roles of viewer bias and competition for attention.

Modelling Language Design is addressed by [44], [45], [46] and [47] who provide good principles and guidelines.

The fields of **Graphic Design, Visualisation** and **User Experience** have evolved to address these demands. There is now a body of knowledge to assist designers in producing good visual products based upon empirical evidence of how we perceive information as humans. Some outstanding contributions in these areas include: [48],

[29], [49], [50], [51]. [52] introduce the field of **Knowledge Visualization** to enhance communication with executive stakeholders and non-specialists. [53] address communication of strategy.

Meta Modelling is vital to structure the environment which will hold models (both semantic and visual). Meta modelling can be achieved in UML (with extensions), in Semantic Models and in a comprehensive notation such as Concept and Object Modelling Notation (COMN) espoused by [54] which allows conceptual, logical and physical modelling in one notation. The physical layer can target various technologies including SQL and NoSQL data stores.

Model Quality is addressed by various authors [including 84], most comprehensively by [55].

Standards are useful for information interchange between tools in a tool eco-system. Commonly used formats include eXtensible Markup Language (XML) and JavaScript Object Notation (JSON). An application of XML, viz. XML Metadata Interchange (XMI) [56] is promoted for inter-operation between tools, but may not have the richness required for our needs. RDF and OWL could be used, but these are less commonly supported outside the semantic community. A language which has proven competent in exchanging semantic information across programming and data definition languages, as well as UML and other tooling, is the FAMOOS Information Exchange Model (FAMIX) as implemented in the Moose platform [57].

Solution Architecture using a service oriented and layered approach is relevant to defining the architecture for a toolset / ecosystem which can support the range of facilities required. A Development Method which can support the evolution of current assets and the definition of new assets towards the end vision is required. The Service Oriented Modeling and Architecture (SOMA) approach from IBM [21] seems most promising.

Exemplar Tools were examined, especially those capable of managing a variety of model types and allowing non-programmers to define meta models and associated notations. We sought to identify concepts and meta model features that allowed the necessary flexibility as well as architectural approaches which have proven practical. Tools examined include: The author and colleagues Enterprise Value Architect (EVA), MEMO [27], Meta Edit+ [58], XModeler [59] and Eclipse [60]. [61] illustrates generation of visual models from semantic models. Lyra [62] illustrates interactive visualisation design.

Domain Specific Language (DSL) may be an appropriate technique to interact with stakeholders and to invoke services required of tool elements. The author previously defined a Meta Enterprise Architecture Language (MEAL) [63] in this vein. Other tools which serve as examples include: [64] while [65] and [66] provide guidance on use in enterprise modelling. [67] and [65] provide guidance on DSL design. [47] and [68] discuss development of visual language syntax.

Development Tools and Libraries which we have chosen for prototyping and proof of concept work include: The **Pharo** open source Smalltalk language and

Integrated Development Environment (IDE) [69], the **Roassal** and **Mondrian** visualisation libraries [70]. These were chosen for openness, familiarity, expressiveness and the "live programming" dynamic bound nature of the environments. Mondrian was also the original environment in which Polymetric Diagramming was implemented.

3 Research Goals and Questions

RG1	Define a meta model capable of holding formal semantic description, syntax and semiotics of common graphical model types
RG2	Define extensions to the above capable of supporting multiple model types relating to the same semantic models and multiple visual languages
RG3	Define extensions to allow the design time specification of and runtime application of modifications to support more effective model use (e.g. by polymetric diagramming, graphical model layout and other techniques)
RG4	Define tool characteristics and architecture to support the more effective definition of visual languages and the more effective use of the visual language principles and guidelines at model use time (e.g. by modifying representation and/or layout to suit purpose, audience and medium)
RG5	Demonstrate efficacy by exemplar visual language support in prototype tooling (e.g. of typical enterprise and IS requirements visual models with emphasis of relevant information)

Media for model consumption may include (*inter alia*) interactive computer screen; small screen (phone or tablet); printed document (A4); presentation (e.g. landscape PowerPoint or Keynote); Poster (A2 or larger for wall display or display on LCD monitors); Wall (larger than A0).

4 Research Method

The primary goal of the research is to produce improvements in the efficacy of graphical models with respect to the ability of a modeller, analyst or viewer to apprehend and interpret useful information. What information is regarded as useful will vary depending upon the role and perspective of the person, the context and the goals of the modelling or interpretation of the model.

The research falls in the field of information systems, since it involves a socio-technical system i.e. artefacts and human interaction with them [71]. More specifically it involves the interaction of analysts and modellers with information technology producing models and the resultant models.

IS is a multidisciplinary field with two major branches of research. The naturalistic branch seeks to find universal truth in observed phenomena. Design (science) research seeks to create utility in artificial creations (artefacts). This research falls in the second camp. There is no existing phenomenon to examine in a naturalistic way. On the other hand, the intent is to create a technology and artefacts of use. Using the guide provided by [72], it falls in the Improvement and Exaptation quadrant: new solutions for known problems; adapt solutions from other fields.

The research aims to produce a technology-based solution to an important business problem. This is essentially the definition of the goal of Design Science Research (DSR) as defined by [72]. The research will produce a number of artefacts, which is also a characteristic of DSR. These include:

A1 Exemplar Models which demonstrate the application of the techniques with a variety of relevant model types

A2 A (Meta)Meta Model which includes the necessary concepts, relationships and attribute structures to describe the desired model features, modifications and their mapping to the underlying data of represented objects

A3 A Documented Process which describes how the modifications are conceived, described and applied

A4 Guidelines for Effective Modification Types to assist with the effective use of the approach and avoiding bad practices

A5 Software Prototype to evaluate the potential for automated support leading to a

A6 Tool Ecosystem Architecture to facilitate seamless interoperation of tools in support of better visual language design and use

The research will also contribute to theory building, which is another goal of DSR. Specifically, we aim to develop two theories which extend the "Physics of Notations" [33]:

T1 Descriptive Theory - A category four theory [73] which explains how and why polymetric modelling works and predicts effect of its application

T2 Prescriptive Theory - A category five theory [73] which provides guidance for design and action

Additional artefacts, beyond the scope of this research, but contemplated for further work, include:

A7 Software system implementing the approach and the meta model defined

A8 User interface to simplify specification for the user, further increasing efficiency and enabling unaided analysis-time use by a much wider (non developer) audience

4.1. Selection of Research Method

There are many flavours of design science research, as it is still a maturing field. The author looked in particular at [72], [74], [75], [76], as well as [77], [78] before settling on an integrated approach that (a) combined all relevant elements of the preceding approaches (b) represented current best practice (c) fit our problem space well. The approach selected is articulated by [79]. Their roadmap is a verified consensus and presents a capable view of how to conduct Design Science Research, summarised in Figure 1.

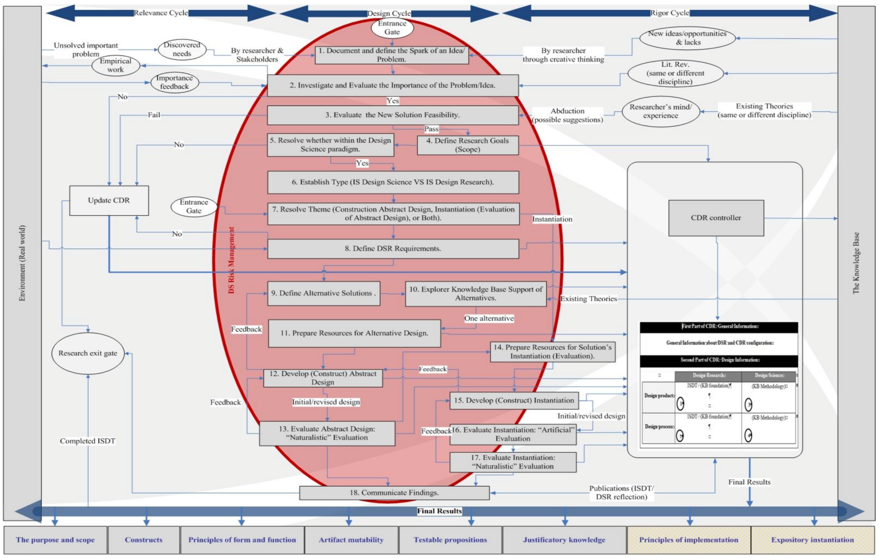

Figure 1 - Design Science Research Framework - [79]

4.2. Application of Method

The method recommended by [79] was followed fairly closely.

- The **Relevance Cycle** (left 1/3) is concerned with clarifying the problem, checking its relevance, seeing if there is a potential (designable) solution, identifying sources of knowledge and setting criteria for completion of the research. It ensures the research engages with the real world context and issues
- The **Design Cycle** (middle 1/3) sets out a cycle of activities to be carried out to progress the research. It ensures that a structured approach is taken to the research and that important steps are carried out
- The **Rigour Cycle** (right 1/3) provides guidance on ensuring rigour in the performance of the research. It includes creative thinking (to find ideas and opportunities and potential solutions); Literature survey to ensure familiarity with prior research of relevance and a sound theoretical grounding and the management of the Central Design Repository (CDR). The repository has components to record

details about the design and the artefacts as work progresses, as well as to record the process, progress and important decisions and outcomes during the research.

- At the base of the diagram, the **Final Results** section documents outputs which should be created by the research, including: Purpose and Scope; Constructs; Principles of Form and Function; Artefact Mutability; Testable Propositions; Justificatory Knowledge; Principles of Implementation; and Expository Instantiation(s)

For further guidance on what should be written up in communicating design science research we consulted [80] albeit with the caveat from [78] that not all recommended elements are relevant in every DSR project.

Since the research will create models, software and methods, it is also aligned with the German tradition of Wirtshaftsinformatik (research through development: possible future usage patterns) [81].

5 Progress to Date

Much effort has been expended in clarifying the problems, surveying relevant literature and selecting and refining an appropriate research method. The study started with the University of Cape Town, but was moved to Duisburg-Essen which presented a more appropriate home for the topic.

Model types commonly used and their generic characteristics have been identified. (RQ1)

Familiarity with PoN and its application has been established. We have catalogued difficulties in use of models in practice and in use of PoN in various applications. (RQ2, RQ3)

Literature on the design of visual languages, graphical notations and their efficacy in practice has been reviewed to identify inhibitors of effectiveness. (RQ3)

Literature on human visual system, cognition, use of symbols and associated biology and neurology has been reviewed to identify aspects which could be better utilised and improved. (RQ4, RQ6)

Requirements in support of polymetric diagramming, visual language design and usage time interaction have been identified. (RQ5, RQ6)

Tools have been examined to glean support mechanisms for generic modeling and how they handle notation. (RQ5)

Consideration of how graphics are defined in other environments and in our chosen development environment. (RQ5)

Definition of meta model to support semantics, model types, basic visual language and selected polymetric modifications is substantially complete. The highest level of this is shown as Figure 2. (RQ5) Space does not permit more detail.

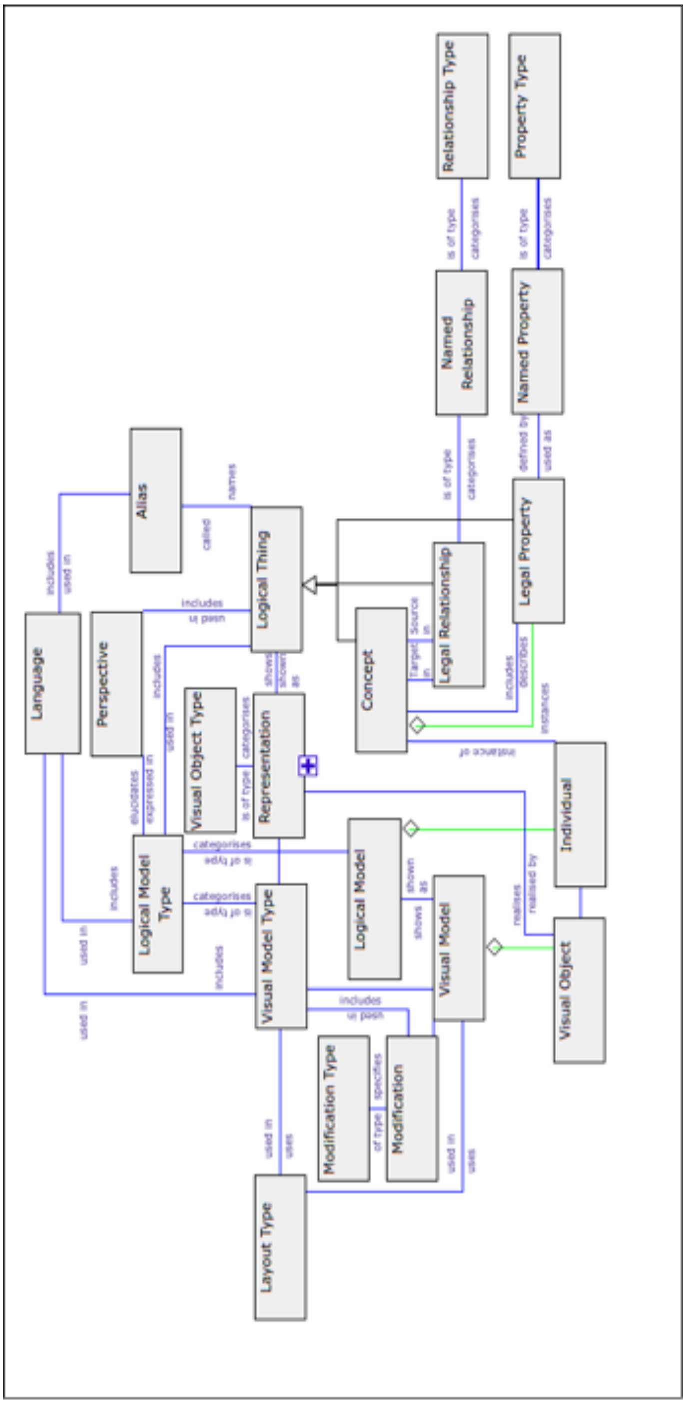

Figure 2 - High Level Draft Meta-Meta Model for Tool Support

A draft layered architecture and API requirements has been created for a proposed toolset / ecosystem, highest level shown as Figure 3. Space does not permit more detail (RQ5)

A selection of prototype tool components have been constructed using the chosen tooling to validate ideas. (RQ5)

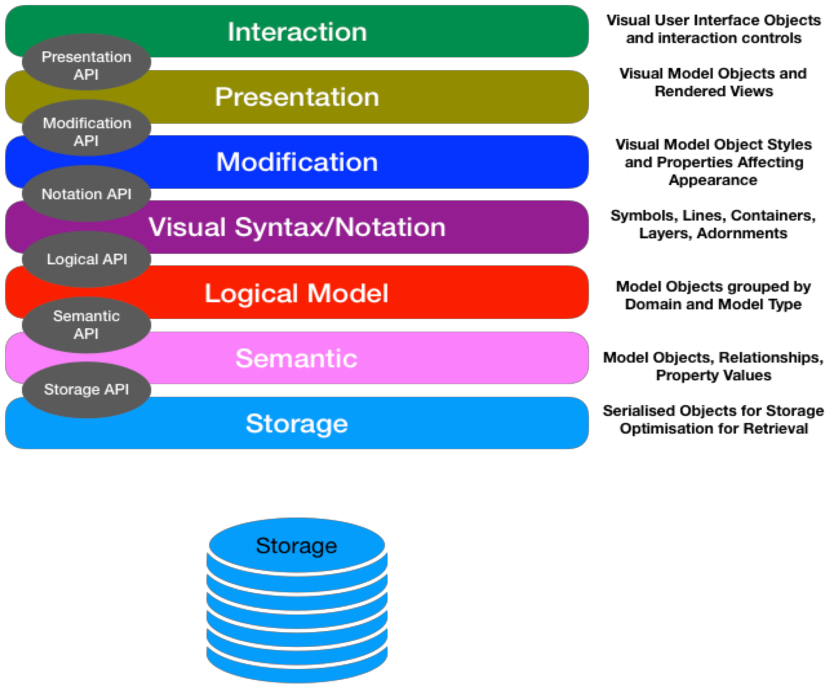

Figure 3 - Layered Tool Architecture (high level)

6 Contribution

C1 Extend PoN to deal with Polymetric Diagramming via descriptive and prescriptive theories

C2 Extend the application of the PoN to enterprise models

C3 Apply Polymetric Diagramming to enterprise models

C4 Define meta model to facilitate construction of language engineering tools and tools which are more effective at model use time

C5 Provide layered model to deal with different concerns architecturally and support modelling tool eco-system

C6 Provide guidance in method and principles for visual language design

The above will collectively contribute to value of models, reduced effort in model translation and interpretation and ultimately, improved ROME.

References

1. Valerdi R, Nightingale D (2011) An Introduction to the Journal of Enterprise Transformation. Journal of Enterprise Transformation 1:1–6
2. Brynjolfsson E, Hitt L (1994) Computers and Economic Growth: Firm-Level Evidence. Sloan Working Paper 3714
3. Zachman JA (1987) A framework for information systems architecture. IBM Systems Journal 26:276–292
4. IFIP-IFAC (1999) GERAM: Generalised Enterprise Reference Architecture and Methodology. IFIP,
5. McDaniel D (2009) DoDAF 2.0 Meta Model (DM2) walkthrough. DOD Enterprise Architecture Conference
6. CIO_Council:_US_Government (1999) Federal Enterprise Architecture Framework Version 1.1.
7. van't Wout J, Waage M, Hartman H et al. (2010) The Integrated Architecture Framework Explained. Springer-Verlag, Berlin Heidelberg
8. Group O (2011) TOGAF Version 9.1 Document Number: G116. Open Group,
9. Smith KL, Graves TS (2011) An Introduction to PEAF: Pragmatic Enterprise Architecture Framework. Pragmatic EA Limited,
10. Group O (2012) Archimate 2.1 Specification.
11. Frank U (2014) Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. Software & Systems Modeling 13:941–962
12. Dietz JLG, Hoogevorst JAP, Albani A et al. (2013) The Discipline of Enterprise Engineering. International Journal of Organisational Design and Engineering 3:86–114
13. Anaya V, Berio G, Harzallah M et al. The Unified enterprise modelling language - overview and further work. Computers in Industry 61:99–111
14. Menzel C, Mayer RJ (1998) The IDEF Family of Languages. In: P B, K M, G S (eds) Handbook on Architectures of Infromation Systems. Springer, Berlin, Heidelberg
15. Martin J (1989) Information Engineering Book 1: Introduction. Prentice Hall, New Jersey
16. Pohl K, Rupp C (2016) Requirements Engineering Fundamentals: a study guide for the Certified Profesional for Requirements Engineering exam: foundation levelO. O'Reilly Media, Sebastopol, CA
17. Group OM (2011) OMG Unified Modelling Language (OMG UML), Infrastructure Version 2.4. Object Management Group,
18. Chen PPS (1976) The entity-relationship model - toward a unified view of data. ACM Transactions on Database Systems (TODS) 1:9–36
19. Group OM (2011) Business Process Model and Notation (BPMN) Version 2.0. Object Management Group,
20. Bochmann G (2010) Goal Modelling and GRL. Course SEQ3101
21. Arsanjani A, Ghosh S, Allam A et al. (2008) SOMA: A method for developing service-oriented solutions. IBM Systems Journal 47 No 3
22. de Kinderen S, Proper HA (2013) e 3 RoME: a value-based approach for method bundling. 28th Annual ACM Symposium on Applied Computing 1469–1471
23. BIAN.org (2015) Banking Industry Architecture Network Services Landscape.
24. Bryant BR, Gray J, Mernik M et al. (2011) Challenges and directions in formalizing the semantics of modeling languages. Computer Science and Information Systems 8:225–253
25. Allemang D, Hendler J (2011) Semantic Web for the Working Ontologist 2nd Edition. Morgan Kaufmann,
26. Dictionary.com (2015) Terminology Dictionary at Dictionary.com.
27. Frank U (2011) The MEMO Meta Modelling Language (MML) and Language Architecture. ICB Research Report 43
28. Shannon CE (2001) A Mathematical Theory of Communication. ACM SIGMOBILE Mobiel Computing and Communications Review 5:3–55
29. Ware C (2008) Visual Thinking for Design. Morgan Kaufmann,

30. Treisman A Preattentive processing in vision. Computer vision, graphics and image processing 31:156–177
31. li Z (2000) Pre-attentive segmentation in the primary visual cortex. Spatial Vision 13:25–50
32. Bertin J (1983) Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press, Madison, Wisconsin, USA
33. Moody DL (2009) The physics of notations: toward a scientific basis for constructing visual notations in software engineering. Software Engineering, IEEE Transactions on 35:756–779
34. Linden DVD, Zamansky A, Hadar I (2016) How cognitively effective is a visual notation? On the inherent difficulty of operationalizing the physics of notations. International Workshop on …
35. van der Linden D, Zamansky A, Hadar I (2016) Cognitive Effectiveness of Conceptual Modeling Languages: Examining Professional Modelers. Lecture Notes in Business Information Processing 248:448
36. van der Linden D, Hadar I, Zamansky A (2016) Towards a Marketplace of Visual Elements for Notation Design. 2016 IEEE 24th International Requirements Engineering Conference (RE) Beijing, China:353–358
37. Caire P, Genon N, Heymans P et al. (2013) Visual notation design 2.0: Towards user comprehensible requirements engineering notations. Requirements Engineering Conference:115–124
38. van der Linden D, Hadar I (Forthcoming) A Systematic Literature Review of Applications of the Physics of Notations. IEEE Transactions on Software Engineering 2018:NA
39. Lanza M (2003) Object-Oriented Reverse Engineering Coarse-grained, Fine-grained, and Evolutionary Software Visualization. Dissertation, Universitat Bern
40. Ware C (2013) Information Visualisation - Perception for Design (3rd Edition). Morgan Kaufmann,
41. Mirzoeff N (2012) The Visual Culture Reader - 3rd Edition. Routledge,
42. Persson P Attention manipulation and information overload. IFN Working Paper 995
43. Van Knippenberg D, Dahlander L, Haas MR et al. (2015) Information, Attention and Decision Making. Academy of Management Journal 58:3:649–657
44. Paige RF, Ostroff JS, Brooke PJ (2000) Principles for modeling language design. Information and Software Technology
45. Frank U (2011) Some Guidelines for the Conception of Domain-Specific Modelling Languages. In: Nuttgens M, Thomas O, Weber B (eds) Enterprise Modelling and Information Systems Architectures (EMISA 2011).
46. Pedro L, Risoldi M, Buchs D (2009) Composing Visual Syntax for Domain Specific Languages. HCI 2009:889–898
47. Plauska I, Damaksevicius R (2013) Design of Visual Language Syntax for Robot Programming Domain. ICIST 297–309
48. Agrawala M, Li W, Berthouzoz F (2011) Design principles for visual communication. Commun ACM 54:60
49. Malamed C (2011) Visual Language for Designers: Principles for Creating Graphics that People Understand. Rockport Publishers, Beverley, Massachusetts
50. Malamed C (2013) Visual Design Solutions - Principles and Creative Inspiration for Learning Professionals. John Wiley & Sons, Hoboken, New Jersey
51. Bostock M, Ogievetsky V, Heer J (2011) D3: Data-Driven Documents. IEEE Transactions on Visualization and Computer Graphics 17:2301–2309
52. Eppler MJ, Burkhard RA (2004) Knowledge Visualization. Towards a New Discipline and its Fields of Application.
53. Kernbach S, Eppler MJ, Bresciani S The Use of Visualization in the Communicaiton of Business Strategies: An Experimental Evaluation. International Journal of Business Communication
54. Hills T (2016) NOSQL and SQL Data Modeling. Technics Publications, Basking Ridge, NJ
55. Krogstie J (2017) Quality of Conceptual Models in Model Driven Software Engineering. In: Cabot J, Gomez C, Pastor O et al. (eds) Conceptual Modeling Perspectives. Springer,
56. OMG (2007) MOF 2.0/XMI Mapping, Version 2.1.1. Object Management Group, Cambridge, MA
57. Girba T (2015) The Moose Book. Girba (Creative Commons),
58. Kelly S, Lyytinen K, Rossi M et al. (2013) MetaEdit+ at the Age of 20. Seminal Contributions to … Seminal Contributions to Information Systems Engineering

59. Clark T, Sammut P, Willans J (2015) Applied Metamodelling - A Foundation for Language Driven Development. Third Edition. arXiv preprint arXiv:1505.00149.,
60. Foundation E (2017) Eclipse Modeling Tools.
61. Buckl S, Ernst AM, Lankes J et al. (2007) Generating Visualizations of Enterprise Architectures using Model Transformations. Enterprise Modelling and Information Systems Architectures EMISA 2007 33–46
62. Satyanarayan A, Heer J (2014) Lyra: An Interactive Visualization Design Environment. Eurographics Conference on Visualization (EuroVis) 33
63. McLeod G (2009) Cooking up a MEAL: Creating a Meta Engterprise Architecture Language.
64. Baetens N (2011) Comparing graphical DSL editors: AToM3, GMF. MetaEdit+ Technical report …
65. Frank U (2010) Outline of a method for designing domain-specific modelling languages. ICB Research Report 42
66. Sonnenberg C, Huemer C, Hofreiter… B (2011) The REA-DSL: A domain specific modeling language for business models. Advanced Information …
67. Karsai G, Krahn H, Pinkernell C et al. (2014) Design guidelines for domain specific languages. 9th OOPSLA Workshop on Domain Specific Modeling (DSM'09) Preprint
68. Guizzardi G, Pires LF, van Sinderen MJ On the role of Domain Ontologies in the design of Domain-Specific Visual Modeling Languages.
69. Pharo (2015) The Immersive Programming Experience.
70. Pena-Araya V (2015) Agile Visualization with Roassal. In: (ed) Agile Visualization. agilevisualization.com,
71. Benbasat I, Zmud RW (2003) The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. MIS quarterly
72. Hevner A, Chatterjee S (2010) Design science research in information systems.
73. Gregor S (2006) The nature of theory in information systems. MIS quarterly 611–642
74. Peffers K, Tuunanen T, Rothenberger MA et al. (2007) A Design Science Research Methodology for Information Systems Research. Journal of Information Systems 24:45–78
75. Gregor S, Jones D (2007) The anatomy of a Design Theory. Journal of the Association of Information Systems 8:312–335
76. Vaishnavi V, Kuechler WJ (2008) Design science research methods and patterns: innovating information and communication technology. Auerbach Publications, Taylor & Francis Group LLC, Boca Raton, New York
77. Venable JR (2010) Design Science Research Post Hevner et al: Criteria, Standards, Guidelines, and Expectations. In: (ed) Global Perspectives on Design Science Research. Springer, Berlin Heidelberg
78. Perjons E, Rossi M, Alter S et al. (2015) Panel on Design Science Methods at CAiSE'15 Stockholm.
79. Alturki A, Gable GG, Bandara W et al. (2012) Validating The Design Science Research Roadmap: Through The Lens Of "The Idealised Model For Theory Development". PACIS 2012
80. Gregor S, Hevner AR (2013) Positioning and presenting design science research for maximum impact. MIS quarterly 37:337–356
81. Frank U (2006) Towards a Pluralistic Conception of Research Methods in Information Systems Research. ICB Research Report 7:1–69
82. Bardhan Indranil, Krishnan Viswanathan, Lin Shu (2013) Research Note - Business Value of Information Technology: Testing the Interaction Effect of IT and R&D on Tobin's Q. Information Systems Research 24(4):1147-1161
83. Brynjolfsson Erik, Hitt Lorin M (2000) Beyond Computation: Information Technology, Organizational Transformation and Business Performance. Journal of Economic Perspectives 14(4):23-48
84. Frank, U. (2007). Evaluation of Reference Models. W: P. Fettke & P. Loos (red.), Reference Modeling for Business Systems Analysis (s. 118–140). Hershey: Idea Group.
85. Sousa Kenia, Vanderdonckt Jean, Henderson-Sellers Brian, Gonzalez-Perez Cesar (2012) Evaluating a Graphical Notation for Modelling Software Development Methodologies. Journal of Visual Languages and Computing 23:195-212
86. Green TRG, Petre M (1996) Usability Analysis of Visual Programming Environments: a 'cognitive dimensions' framework. Journal of Visual Languages and Computing 7:131-174