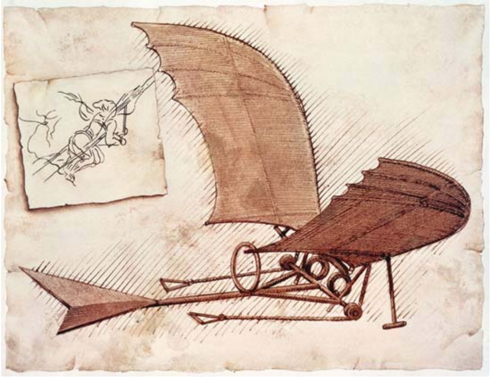
The “Physics” of Notations: A Scientific Approach to Designing Visual Notations in Requirements Engineering[*]

Visual representations form an integral part of the “language” of requirements engineering. Most RE techniques rely heavily on diagrams to document and communicate user requirements: an RE notation *without* a visual representation is almost unheard-of. Visual representation aspects critically determine the effectiveness of RE notations, both for communication with business stakeholders and in supporting problem solving by requirements engineers.

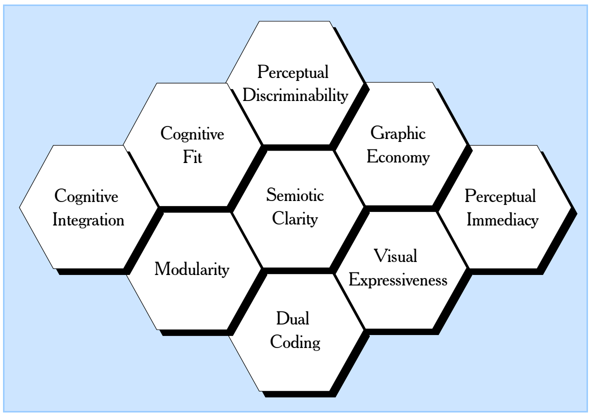
Currently, RE visual notations are designed in an *ad hoc* and unscientific manner. Decisions about graphic representation are typically made in a subjective way, without reference to theory or empirical evidence, or justifications of any kind (*design rationale*). The majority of effort is spent designing the *semantics* of notations (what constructs to include and what they mean), with the design of *visual syntax* (how to visually represent these constructs) taking place largely as an afterthought.



Visual notations play an important role in all engineering disciplines, but we lack explicit principles for designing them

While RE now has mature methods for evaluating and designing semantics of notations (e.g. ontological analysis, formal semantics), equivalent methods for visual syntax are notably absent. Currently, in evaluating, comparing and constructing visual notations, we have little to go on but intuition and rule of thumb – we have neither theory nor a systematic body of empirical evidence to guide us.

The aim of this tutorial is to establish the foundation for a science of visual notation design, to help it progress from a “craft” (as it currently exists) to a design discipline. It defines a set of principles for designing *cognitively effective* visual notations (summarised below): ones that are optimised for human communication and problem solving. The principles have been successfully used to evaluate and improve several modelling notations [e.g. 2, 4] as well as to design visual notations from first principles [e.g. 3].



Principles for Designing Cognitively Effective Visual Notations

Importantly, these principles are *evidence-based*: they are based on theory and empirical evidence from a wide range of fields. They also rest on an explicit theory of how visual notations communicate: only by understanding *how* and *why* visual notations communicate can we improve their ability to communicate. The principles provide a scientific basis for evaluating, comparing and constructing visual notations, which has previously been lacking in the RE field. A range of examples (both *exemplars* and *counter-exemplars*) are used to illustrate the principles.

[*] We call this the “physics” of notations because we focus on the physical (perceptual) properties of notations rather than their logical (semantic) properties (as is more commonly the case).

Intended audience

The tutorial is targeted at the following groups:

- RE researchers: the tutorial provides a theoretical basis for evaluating, comparing and designing visual notations.
- Notation designers and method engineers (e.g. members of groups like OMG): it provides practical guidelines for constructing and improving visual notations.

Tutorial Format

The tutorial is designed to be highly interactive, with practical exercises on designing and improving visual notations as well as group discussions and brainstorming sessions on some of the finer points of visual notation design. Examples and exercises feature some of the most widely used RE notations (e.g. ***i****, UML, ER, EPCs).

Learning Objectives

Visual notations form a critical part of RE research and practice, yet currently, we lack sound principles for designing them. This tutorial addresses the following problem: given a particular set of notation semantics (as defined by a metamodel and formal semantics), how can we design a visual notation that optimises human communication and problem solving? The tutorial also addresses the related problem of how to evaluate, compare and improve existing visual notations.

Topic Outline

1. Introduction

- The Nature of Visual Notations
- What Makes a Good Visual Notation: Cognitive Effectiveness

2. Current Practice in Visual Notation Design

- A job for amateurs
- Lack of justification (design rationale)
- Limited search of the graphic design space
- Examples: UML, ER, ***i****

3. Previous Research

- Ontological analysis
- The Cognitive Dimensions of Notations (CDs) Framework

4. How Visual Notations Communicate (descriptive theory)

- Communication Theory
- Semiotic Theory
- Graphic Design Theory
- Visual Perception Theory
- Cognitive Theory

5. Principles for Designing Effective Visual Notations (prescriptive theory)

- Principle of Semiotic Clarity: There should be a 1:1 correspondence between semantic constructs and graphical symbols
- Principle of Perceptual Discriminability: Symbols used to represent different constructs should be clearly distinguishable
- Principle of Perceptual Immediacy: Use graphical representations whose appearance suggests their meaning
- Principle of Modularity: Include explicit complexity management mechanisms
- Principle of Cognitive Integration: Include explicit mechanisms to integrate separate diagrams
- Principle of Visual Expressiveness: Fully utilise the graphic design space
- Principle of Dual Coding: Use text to clarify and refine meaning of diagrams
- Principle of Graphic Economy (less is more): The number of different graphical conventions should be cognitively manageable
- Principle of Cognitive Fit: Different visual dialects should be used for different audiences
- Trade-offs and synergies: interactions among the principles

6. Conclusion

- Applying the principles: evaluating, comparing, improving and designing visual notations
- Anatomy of a design theory

Note: The content of this tutorial is based on a paper to appear in the October issue of *IEEE Transactions on Software Engineering* [1]: this will be available for distribution to participants at the time of the conference. This tutorial will be the first time this work has been presented at a public forum and includes much material excluded from the journal paper for reasons of space.

Presenter Biography

Daniel Moody is a Visiting Professor in the Department of Information Systems at the University of Twente (the Netherlands). He holds a PhD in Information Systems from the University of Melbourne, but his experience spans both research and practice. Daniel has held academic positions at universities in Australia, Norway, Spain, Czech Republic, Slovenia, Iceland and the Netherlands, has published over 100 scientific papers and chaired several international conferences. He has also held senior positions in major corporations and IT consulting firms. He has conducted consulting assignments in 12 different countries in a range of industries including banking, law enforcement, television broadcasting, pharmaceuticals, biotechnology, airline travel, emergency services, healthcare, education and forestry. He is the current President of the *Australian Data Management Association* (*DAMA*) and is listed in *Who's Who in Science and Engineering*.

References

[1] Moody, D.L. (2009): The "Physics" of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, **October**.

[2] Moody, D.L., P. Heymans, and R. Matulevicius (2009): Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of the i* Visual Notation. *17th IEEE International Conference on Requirements Engineering (RE09)*, Atlanta, Georgia.

[3] Moody, D.L. and R.M. Mueller (2009), From Verbalisation To Visualisation: A Unified Language for Theory Modelling (UTML). *MIS Quarterly*, **(under review)**.

[4] Moody, D.L. and J. van Hillegersberg (2008): Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Suite of Diagrams. *Proceedings of the 1st International Conference on Software Language Engineering (SLE)*, Toulouse, France: Springer Lecture Notes in Computer Science.