Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of *i** Visual Syntax

Daniel L. Moody
Department of Information Systems & Change Management, University of Twente, The Netherlands
d.l.moody@utwente.nl

Patrick Heymans, Raimundas Matulevicius
PReCISE Research Centre
University of Namur, Belgium
phe@info.fundp.ac.be, rma@info.fundp.ac.be

Abstract

*Goal-oriented modelling is one of the most important research developments in the RE field. This paper conducts a systematic analysis of the visual syntax of **i****, *one of the leading goal-oriented languages. Like most RE notations,* ***i**** *is highly visual. Yet surprisingly, there has been little debate about or modification to its graphical conventions since it was proposed more than a decade ago. We evaluate the notation using a set of evidence-based principles for visual notation design. The paper identifies some serious flaws in the* ***i**** *visual notation together with some recommendations for improvement. A broader goal of the paper is to raise the level of debate and stimulate discussion about visual representation in RE research.*

1. Introduction

1.1 Visual representation: an important but neglected issue in RE research

Visual notations play a critical role in requirements engineering (RE), and have dominated RE practice from its earliest beginnings (e.g. the "structured techniques" in the 1970s). An RE method *without* a visual representation is almost unheard-of. Visual notations play a particularly critical role in communicating with end users and customers, as diagrams are believed to convey information more effectively to non-technical people than text [1].

It is therefore surprising that so little research attention has been given to visual representation in RE. Evaluations of RE notations tend to be conducted based exclusively on their semantics, with issues of visual syntax rarely mentioned. Also, in designing notations, decisions about graphic representation tend to be made in a subjective way, without reference to theory or empirical evidence, or justifications of any kind (*design rationale*) [20, 30].

One explanation for the lack of attention to visual aspects is that methods for analysing visual representations are less mature than those available for analysing verbal or mathematical representations [19, 24, 39]. Another explanation is that researchers consider visual syntax to be relatively unimportant: a matter of aesthetics rather than effectiveness [20]. This view is contradicted by research in diagrammatic reasoning that shows that the *form* of representations has an equal, if not greater, influence on their effectiveness as their *content* [23, 36]. Empirical research in RE contexts confirms that the graphical form of notations significantly impacts their effectiveness for both problem solving and end user communication [27, 32].

1.2 Goal-oriented modelling

Goal-oriented modelling is one of the most important research developments in the RE field, which changes the focus from *what* and *how* (data and processes) to *who* and *why* (the actors and the goals they wish to achieve). ***i**** is one of the leading goal modelling languages: the paper proposing it [42] was awarded the title of "Most Influential Paper of the Past 10 Years" at RE07. It competes with the KAOS [11] for the title of the leading goal-oriented modelling notation.

Like most RE notations, ***i**** is a visual language: almost everything in the language has a graphical representation and it primarily relies on diagrams (*Strategic Dependency Diagrams* and *Strategic Rationale Diagrams*) to document and communicate requirements. Like most RE notations, ***i**** lacks explicit design rationale for visual representation choices: graphical conventions are defined by example, without any explanation for why they were chosen.

1.3 Objectives of this paper

This paper conducts the first systematic analysis of the ***i**** visual notation, with an aim towards improving its effectiveness in practice, especially for communication with end users. In this spirit, rather than simply point-

1090-705X/09 $25.00 © 2009 IEEE
DOI 10.1109/RE.2009.44

ing out deficiencies in the notation, where possible, we suggest ways of overcoming these problems. We believe that research attention to the visual aspects of ***i**** is long overdue: there has been little debate about or modification to the ***i**** visual syntax since it was originally proposed more than 10 years ago. The lack of progress in this area represents a potential barrier to its usability and adoption in practice. A second goal of this paper is to stimulate discussion about visual representation issues in the RE community.

2. Theoretical Basis

One reason for the lack of attention to visual aspects in RE research is the lack of accepted principles for evaluating and designing visual notations. In the absence of such principles, evaluations can only be carried out in a subjective manner.

2.1 A Theory for Visual Notation Design

The analysis in this paper is based on a recently developed theory of visual notation design [30], which defines a set of nine principles for designing cognitively effective visual notations, (Figure 1). The theory is called the *Physics of Notations* as it focuses on the physical (perceptual) properties of notations rather than their logical (semantic) properties as is more commonly the case.

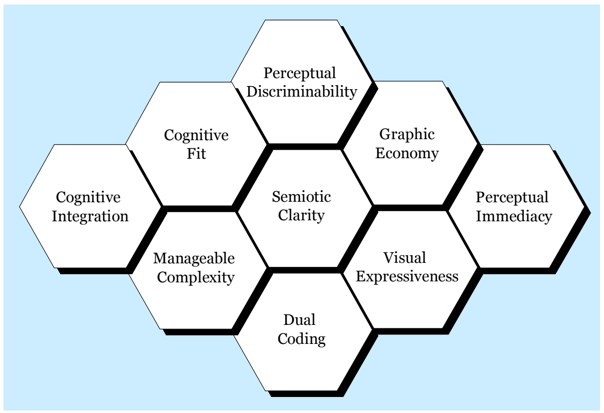

Figure 1. Principles for Cognitively Effective Visual Notations

Importantly, these principles are *evidence-based*: they were synthesised from theory and empirical evidence from a wide range of fields. They also rest on an explicit theory of how visual notations communicate. These provide a scientific basis for evaluating and designing visual notations, which has previously been lacking in the SE field. Rather than defining the principles here, they are defined as they are applied in the analysis (Section 3), so the reader (i.e. you) does not have to take in a lot of new material and remember it (in accordance with principles of *just in time learning*).

2.2 The "Visual Alphabet"

One aspect of the theory that is important to understand the analysis in Section 3 is the concept of **visual variables**. There are 8 elementary visual variables which can be used to graphically encode information [4]. These are categorised into **planar variables** (the two spatial dimensions) and **retinal variables** (features of the retinal image) (**Figure 2**). The visual variables define a **visual alphabet** for designing visual notations: notation designers can create an almost unlimited number of graphical symbols by combining the visual variables together in different ways.

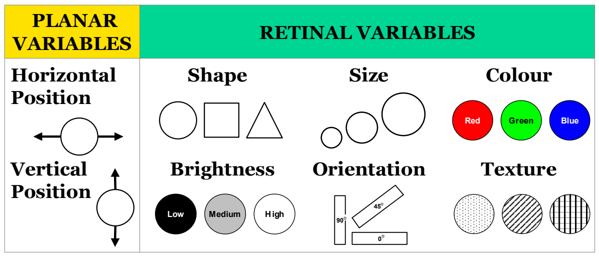

Figure 2. Visual variables [4]

3. Analysis and Results

This section presents the results of a comprehensive, symbol-by-symbol analysis of the ***i**** visual notation. The findings for each principle are presented in a separate subsection (references to principles are indicated in the text by underlining; new terms by **bolding**). Due to space limitations, only the results for 6 of the 9 principles are reported in this paper.

3.1 Unit of Analysis

The first issue is to choose a particular source of ***i**** to use as the basis for our analysis: there is no definitive source as there is, say, for UML. We chose the ***i**** *Guide* [17] as our source for three reasons:

- It is web (Wiki) based so is the most *up-to-date* source. This represents the "living" version of the language, whereas the most often cited sources of ***i**** (Yu's original PhD thesis and his RE97 paper [41, 42]), represent snapshots at a point in time.
- It represents a superset of the conventions defined in most other sources, as the language has expanded over time. It therefore provides the most *complete* source.
- It provides the most *detailed* description of the ***i**** visual syntax.

3.2 Semiotic Clarity

The Principle of Semiotic Clarity states that there should be a 1:1 correspondence between semantic constructs and graphical symbols used in a notation. When there is not a 1:1 correspondence, the following anomalies can occur (Figure 3):

- **Symbol deficit**: when a construct is not represented by any symbol
- **Symbol redundancy (synography)**: when a single construct is represented by multiple symbols
- **Symbol overload (homography)**: when a single symbol is used to represent multiple constructs
- **Symbol excess**: when a symbol does not represent any construct.

Semiotic Clarity maximises the *expressiveness* (by eliminating symbol deficit), *precision* (by eliminating symbol overload) and *parsimony* (by eliminating symbol redundancy and excess) of visual notations.

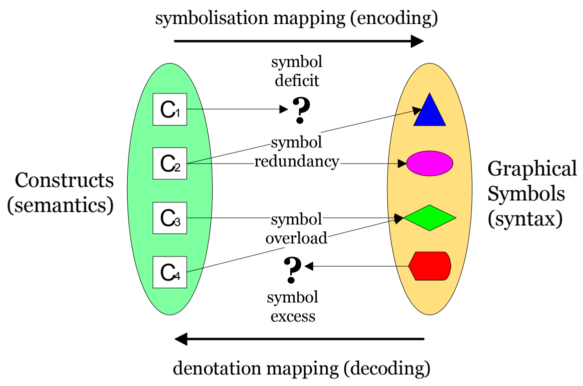

Figure 3. Principle of Semiotic Clarity

Evaluating the semiotic clarity of a notation involves conducting a mapping between its metamodel and symbol set (**visual vocabulary**). This is problematic as the ***i**** Guide does not define an explicit metamodel. To conduct the analysis, we reverse engineered a metamodel (or more precisely, a *metaclass hierarchy*) from the text, drawing on previous research in this area [2, 5, 6]. However, without a metamodel, semiotic analysis can only be conducted in an approximate manner. A clear recommendation from this research is that ***i**** needs an explicit metamodel, represented in standard format (e.g. using MOF [33]). While Yu's PhD thesis [41] includes a metamodel, it is represented in non-standard format and has not been updated to reflect subsequent changes to the language.

Symbol redundancy (synographs)

There is only one case of symbol redundancy in ***i****: two different symbols can be used to represent Beliefs (Figure 4).

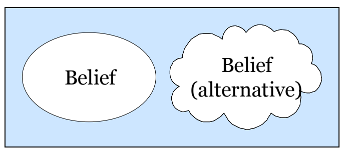

Figure 4. Symbol redundancy (synographs)

No explanation is given for why a choice is provided, which is not provided for any other construct. Symbol redundancy places a burden of choice on the language user to decide which symbol to use and an additional load on the reader to remember multiple representations of the same construct. To resolve this, one of the symbols should be chosen to represent the construct and the other removed from the notation. As we will see, other principles provide clear guidelines for choosing between the two alternatives.

Symbol overload (homographs)

This is the worst type of anomaly as it results in ambiguity and the potential for misinterpretation [16]. It also violates one of the basic properties of the symbol system of graphics: **monosemy** – this means that all symbols should have a single meaning, defined in advance and independent of context [4]. Symbol overload is a serious problem in i*: most of its symbols are technically homographs. All of these occur among relationship types: there are 27 different types of semantic relationship but only 5 visually distinct graphic links[2] (Table 1). This means that, on average, each graphical link has to represent more than 5 different types of relationships. The level of symbol overload is primarily due to the use of **textual differentiation**. Labels are used to distinguish between 6 types of Actor Relationships, 9 types of Softgoal Contributions and 9 types of Softgoal Correlations[3]. **Contextual differentiation** is also used: Actor Relationships and Contribution Relationships use the same graphical link but connect different types of elements. Symbol overload is a common (but suboptimal) way of dealing with excessive graphic complexity (Principle of Graphic Economy).

Table 1. Symbol overload in *i**

Graphic link	Semantic relationship	Overload
——>	Actor association (6 types) Contribution (9 types)	14
— —>	Correlation (9 types)	8
+——	Decomposition	0
——▷	Means-end	0
—D—	Strategic dependency	0
5	**27**	**22**

Symbol excess and symbol deficit

There is no symbol excess or symbol deficit in i*. However, absence of symbol deficit is not necessarily a

2 Two symbols are **visually distinct** if they have a different value for at least one **visual variable**. This is based on the concept of **visual distance** (see Perceptual Discriminability).

3 Alternatively, the different types of contributions and correlations could be considered as *properties* of relationships (strength and direction) so would be considered under Visual Expressiveness.

good thing as this is a useful technique for keeping graphic complexity manageable. Currently, ***i**** has excessive graphic complexity, especially for a notation intended for use in early analysis (see Principle of Graphic Economy).

3.3 Perceptual Discriminability

Perceptual Discriminability refers to the ease and accuracy with which different symbols can be differentiated from each other. Accurate discrimination between symbols is a prerequisite for accurate interpretation of diagrams [39]. Discriminability is determined by the **visual distance** between symbols, which is measured by the number of visual variables on which they differ and the size of these differences (number of **perceptible steps**). In general, the greater the visual distance between symbols, the faster and more accurately they will be recognised [40]. If differences are too subtle, errors in interpretation can result. Requirements for discriminability are much higher for novices than for experts [7].

Discriminability of Shapes

Of all visual variables, **shape** plays a privileged role in perceptual discrimination, as it represents the primary basis on which we classify objects in the real world. This means that more than any other variable, shapes used to represent different constructs should be clearly distinguishable. Figure 5 shows the different types of **nodes** used in ***i****. All are 2 dimensional geometric shapes, with sometimes only very subtle differences between them. These differences are likely to be diluted even further when diagrams are drawn by hand.

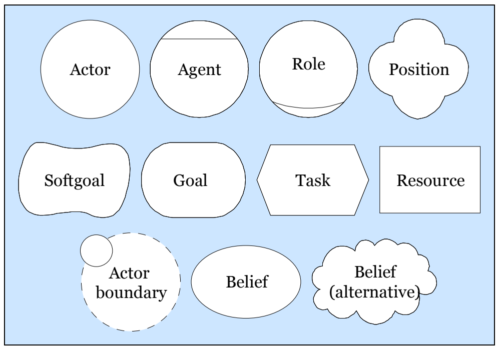

Figure 5. Node types in *i**

In particular, the shapes used to represent Goal and Belief are highly similar. This provides a strong reason to use the alternative symbol (cloud) for Beliefs to resolve the problem of symbol redundancy (Semiotic Clarity). A similar problem exists between Agents and Roles: the difference between the symbols is very subtle (a straight or curved line at the top or bottom of the symbol) and not at all intuitive.

Because of the privileged role of shape in object recognition, differences in shape tend to be interpreted as differences in *kind*. For this reason, the same or similar shapes should be used to represent the same or similar constructs. This principle is violated where one of the subtypes of Actor (Position) is represented by a shape from a different shape family. To avoid giving misleading impressions about the relationship among these constructs, similar shapes should be used to represent all actor subtypes. A possible solution to this problem is discussed under Graphic Economy. A similar issue exists with Goals and Softgoals: shapes from different families are used to represent these concepts when Softgoal is a subtype of Goal (or in TROPOS, Hardgoal and Softgoal are specialisations of an abstract metaclass Goal [6]). A better solution would be to use the same shape with a secondary visual variable to distinguish between them (a possible solution to this problem is discussed in Perceptual Directness).

Strategic Dependencies

Strategic dependencies are represented by lines with the letter "D" attached to each side of the dependency (Figure 6). The orientation of the letters defines the direction of the dependency. This convention is one of the most distinctive (and peculiar) characteristics of the ***i**** visual notation and makes ***i**** diagrams almost immediately recognisable. However, it is not particularly effective as a graphical representation technique:

- The shape of the letter "D" is too symmetrical, making it perceptually difficult to identify the direction of the dependency. The fact that "D"s are attached to both sides of the dependency exacerbates this (e.g. compare Figure 6 and Figure 7).
- Using "D"s on both sides of each dependency creates **visual noise**: ***i**** diagrams are unnecessarily cluttered by the number of "D"s.

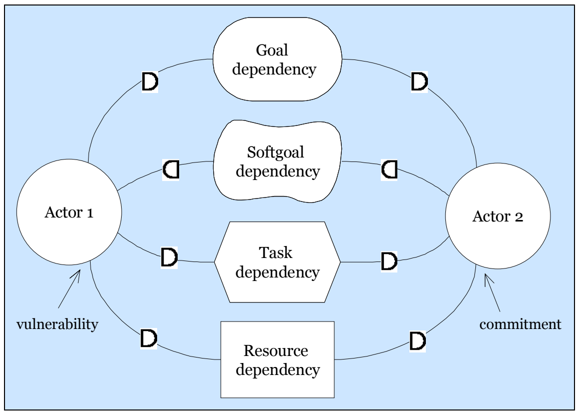

Figure 6. Strategic Dependencies in *i**:** how many vulnerabilities does each actor have? Conscious effort is required to determine the orientation of the "D"s.

Dependencies could be represented more clearly using conventional arrows, making sure to use a different type of arrow to those already used in ***i**** (Figure 7). This resolves all the discriminability problems identified above and also supports Perceptual Directness as arrows are universally understood as a symbol of directionality dating back to primitive times [14, 37] whereas the "D"s always require explanation to new users.

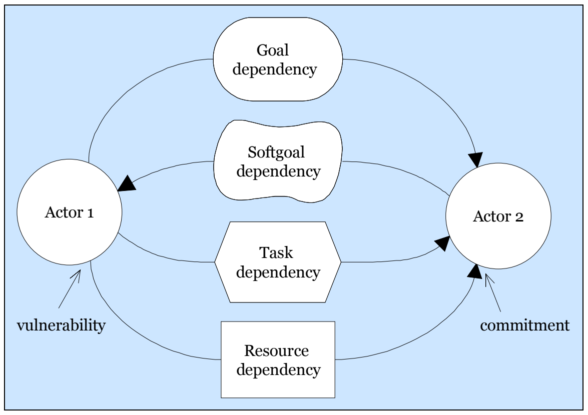

Figure 7. Strategic Dependencies (suggested improvement): how many vulnerabilities or commitments does each actor have? This can be easily determined by the presence or absence of arrowheads.

Textual differentiation

Textual differentiation is commonly used in UML to distinguish between relationship types. However, this is not a graphic design practice that should be emulated, as it is less cognitively effective [31]. As discussed under Semiotic Clarity, textual differentiation results in symbol overload: symbols that are differentiated only by labels have zero visual distance and are technically homographs. It also reduces discriminability as it relies on slower, sequential, cognitive processes [31]. To maximise discriminability, symbols should be differentiated using visual variables so that differences can be detected automatically using perceptual processes.

i* uses textual differentiation of relationships to an even greater extent than UML: around 90% of relationship types are differentiated in this way (see Figure 8 for an example). Textual differentiation of symbols is a common symptom of excessive **graphic complexity**, a problem that ***i**** and UML both suffer from: however there are better ways of dealing with this than resorting to such measures (Graphic Economy).

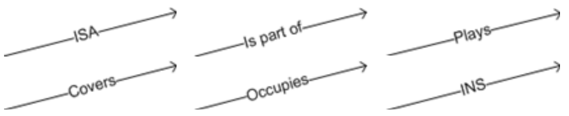

Figure 8. Actor relationships in *i [17]**

Labels play a critical role in diagrams, as they are used to differentiate between **symbol instances** (**tokens**) and define their correspondence to the real world domain. Using labels to differentiate between **symbol types** (language level) rather than symbol instances (diagram level) therefore confounds their role in diagrams. It also precludes the use of user-meaningful and domain-relevant labels for relationships. This is not to say that text should never be used in visual notations. Text can be usefully used for **redundant coding**, to reinforce and clarify meaning (Principle of Dual Coding), but should never be used as the sole basis for distinguishing between symbols.

3.4 Complexity Management

Managing complexity is an important issue in RE [9, 22] and also in visual representation: a well-known problem with visual notations is that they do not scale well [10]. It is especially important for notations designed for communication with end users to have effective complexity management mechanisms as non-experts are less equipped to deal with complexity. Currently, ***i**** lacks effective complexity management mechanisms, which means that problems must be represented as single monolithic diagrams, no matter how complex they become (Figure 9). Without this, ***i**** stands little chance of being adopted in industrial projects, where complexity management represents one of the greatest challenges [12, 15]. One of the few empirical evaluations of ***i**** [13] identified this as one of its most serious limitations in practice.

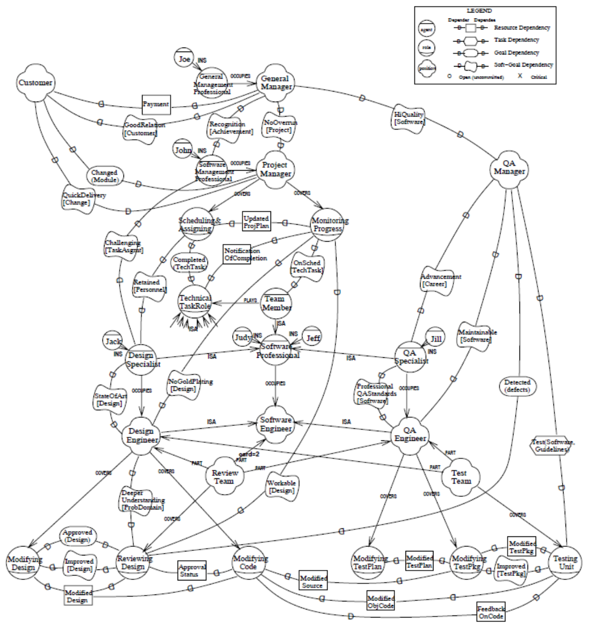

Figure 9. Complexity Management: *i** lacks complexity management mechanisms (diagram from [41]).

To effectively represent complex situations, visual notations need to provide mechanisms for *modularising* and *hierarchically structuring* diagrams. Experimental studies show that this can improve end user understanding of RE diagrams by more than 50% [29]. DFDs provide one of the earliest examples of how to use such mechanisms to manage complexity of diagrams, which may partly explain their remarkable longevity in practice despite their well-known semantic limitations. UML Statecharts and Activity Diagrams also provide examples of modularisation and hierarchical structuring in visual notations.

i* currently lacks modularisation mechanisms of any kind. It provides limited hierarchical structuring (Strategic Rationale and Strategic Dependency diagrams define two levels of abstraction), but to be most effective, multiple levels of decomposition should be provided depending on the complexity of the underlying model.

3.5 Perceptual Directness

Perceptual directness refers to the use of graphical representations whose appearance suggests their meaning. While Perceptual Discriminability simply requires that symbols be *different* from each other, this principle requires that they provide clues to their meaning. Perceptually direct representations reduce cognitive load through built-in *mnemonics*: their meaning can be either perceived directly or easily learnt [35]. Such representations speed up recognition and improve intelligibility, especially to naïve users [7, 27].

i* currently makes very little use of perceptual direct representations. This is surprising for a notation intended for early analysis, given the well-known advantages of such representations for communicating with novices. Most symbols in ***i**** are abstract geometrical shapes that don't convey anything about the constructs they represent: their meaning is purely *conventional* and must be learnt. Ironically, the only exception is a synograph: the cloud (the alternative symbol for Belief) is a widely recognised convention for expressing inner thoughts and feelings (the ubiquitous "thought bubble"). This provides another reason to choose this as the standard symbol for Belief to resolve the problem of symbol redundancy (Semiotic Clarity).

The central constructs in ***i**** are the four dependency types: Goals, Softgoals, Tasks and Resources. Currently, the symbols used for these are neither discriminable nor mnemonic. Figure 10 shows some more perceptually direct symbols that could be used: the Task shape suggests motion or action, the (3D) Resource shape suggests a tangible object, while a football is used to represent both Goals and Softgoals[4]. The dotted outline of Softgoals suggests "fuzziness". These are not **perceptually immediate** in the strict sense that a novice reader could guess their meaning, but are more mnemonic than the abstract geometrical shapes currently used. They are also more discriminable and solve the problem of shape inconsistency identified in Perceptual Discriminability: Goals and Softgoals now have the same shape and are differentiated by a secondary visual variable (brightness).

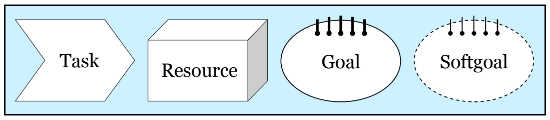

Figure 10. More perceptually direct symbols for dependency types

Iconic (pictorial) representations

Icons (mimetic symbols, pictographs) are symbols which perceptually resemble the concepts they represent [34]. Empirical studies show that replacing abstract shapes with icons improves understanding of RE models by novices [27]. They also improve likeability and accessibility: a pictorial representation appears less daunting to novices than comprised only of abstract symbols [3, 35]. ***i**** currently includes no iconic representations which makes diagrams look rather dull and technical compared to other techniques used in early analysis. For example, UML Use Cases make limited use of icons while *rich pictures* [8] consist entirely of pictorial representations (Figure 11).

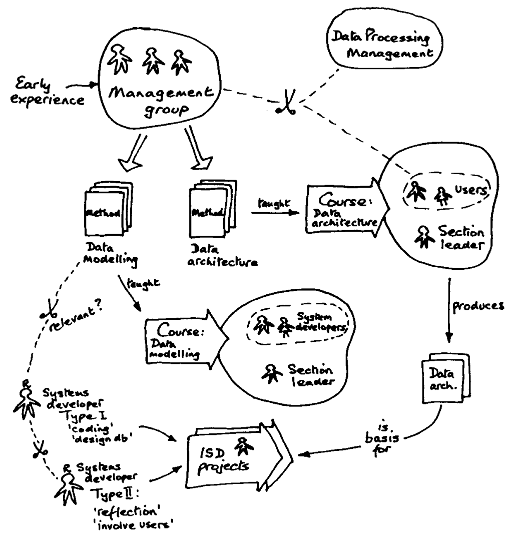

Figure 11. Rich pictures [8]

An obvious way to increase the iconicity of ***i**** diagrams would be to use stick figures for actors (*à la* Use

4 Cultural differences about the meaning of the word *football* would probably require "rugby" and "soccer" dialects of this notation.

Cases). This would increase their discriminability from other constructs (Perceptual Discriminability) but would make it much more difficult to distinguish between the different subtypes of actors (this is discussed further in Graphic Economy). A more radical proposal would be to replace the entire ***i**** symbol set by an iconic vocabulary (*à la* rich pictures).

3.6 Visual Expressiveness

The **visual expressiveness** of a notation is defined by the number of different visual variables used and the range of values (**capacity**) used of each. This measures utilisation of the graphic design space. Using a variety of visual variables results in a perceptually enriched representation which maximises cognitive effectiveness. Currently, ***i**** uses only two visual variables: **shape** and **brightness**. It also uses only two levels of brightness (dotted vs solid lines) and a limited range of shapes (only abstract geometrical shapes): in particular, iconic and 3D shapes are not used which are generally more perceptually and cognitively effective [3, 21, 39]. This means that ***i**** uses only a fraction of the graphic design space.

Use of colour

Colour is one of the most cognitively effective of all visual variables: the human visual system is highly sensitive to variations in colour and can quickly and accurately distinguish between them [25, 40]. However, if not used carefully, it can undermine communication. Currently, colour is used informally and inconsistently in ***i****. Most examples in the ***i**** Guide use green fill for shapes and blue text for labels (e.g. Figure 12) but use of colour is not mentioned in the text, making it unclear whether this is part of the visual syntax.

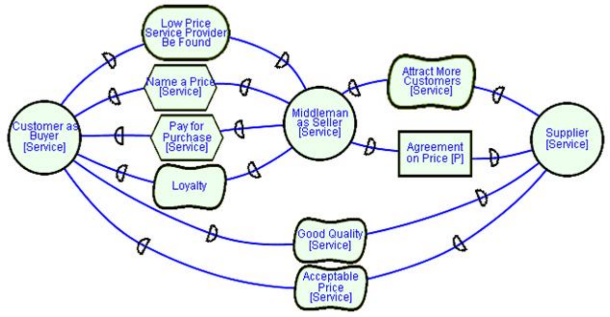
Low Price Service Provider Be Found
Name a Price [Service]
Pay for Purchase [Service]
Loyalty
Customer as Buyer [Service]
Middleman as Seller [Service]
Attract More Customers [Service]
Agreement on Price [P]
Supplier [Service]
Good Quality [Service]
Acceptable Price [Service]

Figure 12. Use of Colour in *i(from [17])**

Whether this is part of the visual syntax or not, it does not represent effective use of colour:

- All symbols are the same colour, meaning that it plays no role in discriminating between symbols. In other words, colour communicates no information.
- Coloured text on a coloured background reduces understanding of text and is the worst possible combination for both legibility and aesthetics [38]. This means that use of colour actively *undermines* communication.

Colour could be used to improve both Visual Expressiveness and Perceptual Discriminability of ***i**** by using different colours (in addition to shape) to distinguish between symbols (e.g. [26]). This is an example of redundant coding: where multiple visual variables are used in combination to discriminate between symbols (see Figure 16 for an example)**.** Like text, colour should not be used as the sole basis for distinguishing between symbols as it is highly sensitive to differences in visual perception and screen/printing technologies.

Textual encoding of information

i* also uses text to define properties of relationships: for example different *dependency strengths* for strategic dependencies (Figure 13). This is *not* an issue of Perceptual Discriminability as these represent *properties* of relationships rather than different relationship types.

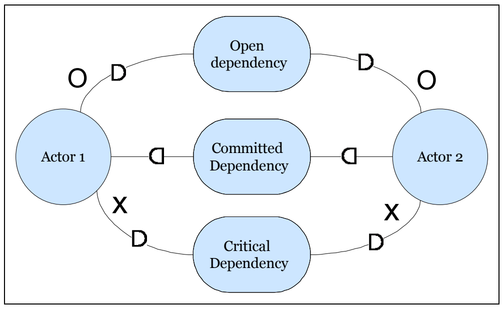

Figure 13. Textual encoding of dependency strengths: O = open, X = critical and no label = committed

Where possible, information should be encoded graphically (i.e. using visual variables) to take maximum advantage of the power of human visual processing. Figure 14 shows an example of how dependency strengths could be encoded graphically.

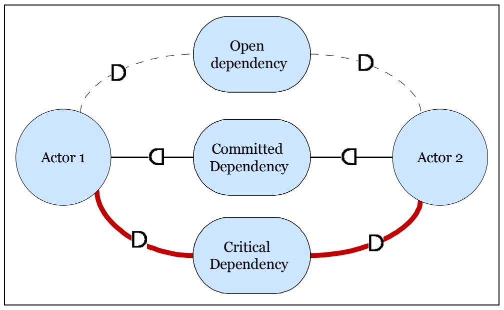

Figure 14. Graphical encoding of dependency strengths: dotted lines for open dependencies (**brightness**), thick red lines for critical dependencies (**size, colour**).

Because dependency strength is an ordinal property, **ordinal visual variables** need to be used to encode this information. Three additional visual variables are used in this representation: *brightness* (dotted lines for open dependencies), *size* (thick lines for critical dependencies) and *colour* (red for critical dependencies).

3.7 Graphic Economy (less is more)

Graphic complexity refers to the number of different symbols used in a notation: the size of its visual vocabulary [32]. This is measured by the number of legend entries required. This is different to **visual complexity** as addressed by Complexity Management, as it relates to complexity at the **type** (language) level rather than at the **token** (diagram) level. Empirical studies show that graphic complexity significantly reduces understanding of RE diagrams, especially by novices [32]. The reason is that the human ability to discriminate between perceptually distinct alternatives (**span of absolute judgement**) is around 6 categories [28]: this defines an effective upper limit for graphic complexity. DFDs (Figure 15), ER diagrams and UML Use Case diagrams are all within this limit.

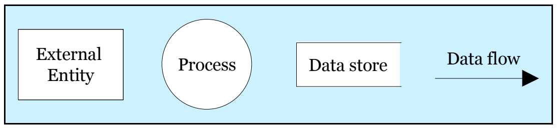

Figure 15. Graphic Economy: DFDs have a very simple and highly discriminable visual vocabulary which supports usability and end user communication

In contrast, ***i**** exceeds this limit by an order of magnitude (almost 3 times). It has a graphic complexity of 17[5], compared to 4 for DFDs and 5 for ER models and UML use cases. Such a level of graphic complexity would be a problem for any visual notation, but particularly for one intended for use in early analysis. Graphic complexity represents a major barrier to both ***i****'s usability and its effectiveness for communication with business stakeholders. There are three strategies for dealing with excessive graphic complexity:

1. Reduce semantic complexity. The number of semantic constructs is a major determinant of graphic complexity as different constructs are usually represented by different graphical symbols. Reducing the number of semantic constructs requires asking some hard questions about the ***i**** metamodel: for example, is it really necessary to distinguish between the various types of Actor (Role, Position, Agent)? For example, DFDs and Use Case diagrams also incorporate all these types of actors but don't distinguish between them. Such questions are beyond the scope of this paper, which focuses on syntactic issues.

2. Introduce symbol deficit. Graphic complexity can be reduced directly (without affecting semantics) by introducing symbol deficit (Semiotic Clarity). This means choosing *not* to show some constructs in graphical form. Judicious use of symbol deficit is one of the most effective ways to reduce graphic complexity. For example, the question could be asked: even if it is necessary to distinguish between Actors, Agents, Roles and Positions at the semantic level, do we need to distinguish between them at the syntactic (diagrammatic) level? Removing this distinction would allow all 4 constructs to be represented by the same symbol, which would reduce graphic complexity by 3 in a single stroke. Further, if there is no need to distinguish between actor subtypes, an icon could be used instead of an abstract shape, thus increasing Perceptual Discriminability and Perceptual Directness.

As a more radical proposal, ***i**** currently uses a large number of different contribution and correlation types. As well as adding to graphic complexity, they also greatly increase the visual complexity of ***i**** diagrams (Complexity Management): inclusion of such relationships can result in a web of lines which can obscure the dependency structure. We could ask the question: do these need to be shown on the diagram at all? This may seem to conflict with the recommendations of Visual Expressiveness, but not all information needs to be shown on the diagram: diagrams are famously good for representing some types of information but not others [18]. Part of the secret to using visual notation effectively is knowing when *not* to use it [10, 35]. In particular, interactions can often be shown most effectively using matrices (e.g. CRUD matrices, quality matrices). The advantage of matrices for such purposes is that they support more systematic analysis: a missing link on a diagram is not as obvious as a missing cell in a matrix. Removing contributions and correlations from the visual notation would reduce both graphic complexity and visual complexity (Complexity Management). Together with the removal of Actor subtypes (described above) and symbol overload (Semiotic Clarity), this would reduce graphic complexity to manageable levels.

3. Increase visual expressiveness. The human ability to discriminate between visual stimuli can be expanded by increasing the number of visual variables on which the stimuli differ [28] (Visual Expressiveness). Using multiple visual variables to differentiate between symbols can increase the human span of absolute judgement in an (almost) additive manner.

4. Conclusion

This paper has conducted a systematic analysis of ***i**** visual syntax, using a set of principles for cognitively effective visual notations. The results can be used to improve its usability and effectiveness (especially for communicating with end users) and remove some po-

[5] The graphic complexity of ***i**** is artificially deflated by the high level of symbol overload. As discussed under Semiotic Clarity, symbol overload is a common, but cognitively ineffective, way of dealing with excessive graphic complexity.

tential barriers to its adoption in practice. We have identified some serious flaws in the ***i**** visual syntax, and in most cases, possible ways of resolving these problems. However, we don't claim to have all the answers and welcome ideas and suggestions (from researchers or ***i**** users) about alternative ways of addressing these issues.

It is beyond the scope of this paper to propose a new visual notation for ***i****. However, as an example of what is possible and as a starting point for further discussion, we propose a simplified visual vocabulary for ***i**** based on some of the recommendations in this paper (Figure 16) (this is not a complete symbol set as it does not include relationship types). Note the mnemonic colour scheme used:

- Tasks are yellow (like "sticky notes")
- Resources are green (like natural resources)
- Softgoals are pink (suggesting softness)

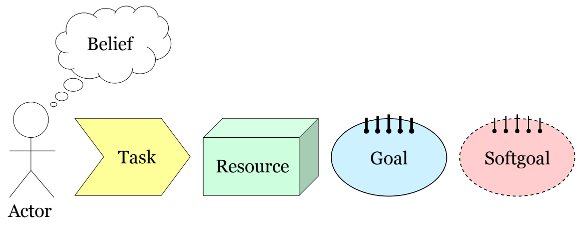

Figure 16. A simplified visual vocabulary for *i**

Compared to the existing symbol set (Figure 5), this is more:

- Semiotically clear: it contains no synographs.
- Perceptually discriminable: it uses a greater variety of shapes, redundant coding (which increases visual distance) and exploits family resemblances among shapes (i.e. Goal, Softgoal).
- Perceptually direct: it uses shapes and colours that suggest the meaning of their referent concepts.
- Visually expressive: it uses three visual variables (colour, shape, brightness) instead of only one (shape); also, a greater range of shapes (including iconic shapes and 3D shapes) are used.
- Graphically economical: the size of the visual vocabulary is reduced by eliminating symbol redundancy and introducing symbol deficit.

Such a symbol set would make ***i**** diagrams more visually appealing and accessible for business stakeholders and would clearly differentiate them from most other technical-looking diagrams used in IT practice (of which UML is a prime example).

Acknowledgment. This work is partially funded by the Interuniversity Attraction Poles Programme, Belgian State, Belgian Science Policy, under the MoVES project.

References

[1] Avison, D.E. and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools (3rd edition)*. 2003, Oxford, United Kingdom: Blackwell Scientific.

[2] Ayala, C.P., C. Cares, J.P. Carvallo, G. Grau, Mariela Haya, G. Salazar, X. Franch, E. Mayol, and C. Quer. *A Comparative Analysis of i*-Based Agent-Oriented Modeling Languages*. in *Proceedings of the International Workshop on Agent-Oriented Software Development Methodology (AOSDM'2005)*. 2005.

[3] Bar, M. and M. Neta, *Humans prefer curved visual objects*. Psychological Science, 2006. **17**(8): p. 645-648.

[4] Bertin, J., *Semiology of Graphics: Diagrams, Networks, Maps*. 1983, Madison, Wisconsin, USA: University of Wisconsin Press.

[5] Bertolini, D., A. Perini, A. Susi, and H. Mouratidis. *The TROPOS Visual Modeling Language: A MOF 1.4 Compliant Metamodel*. in *Agent-Oriented Software Engineering Technical Forum*. 2005. Ljubljana, Slovenia.

[6] Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopolous, *TROPOS: An Agent-Oriented Software Development Methodology.* Autonomous Agents and Multi-Agent Systems, 2004. **8**: p. 203-236.

[7] Britton, C. and S. Jones, *The Untrained Eye: How Languages for Software Specification Support Understanding by Untrained Users.* Human Computer Interaction, 1999. **14**: p. 191-244.

[8] Checkland, P.B. and J. Scholes, *Soft Systems Methodology In Action*. 1990, Chichester, England: Wiley.

[9] Cheng, B.H. and J.M. Atlee. *Research Directions in Requirements Engineering*. in *International Conference on Software Engineering (ICSE07)*. 2007. Washington, DC, USA: IEEE Computer Society.

[10] Citrin, W., *Strategic Directions in Visual Languages Research.* ACM Computing Surveys, 1996. **24**(4).

[11] Dardenne, A., A. van Lamsweerde, and S. Fickas, *Goal-Directed Requirements Acquisition.* Science of Computer Programming, 1993. **20**: p. 3-50.

[12] Dijkstra, E.W., *On the Cruelty of Really Teaching Computer Science.* Communications of the ACM, 1989. **32**(12): p. 1398-1404.

[13] Estrada, H., A.M. Rebollar, O. Pastor, and J. Mylopoulos. *An Empirical Evaluation of the i* Framework in a Model-based Software Generation Environment.* in *CAiSE 2006 (LNCS 4001)*. 2006: Springer-Verlag.

[14] Frutiger, A. and A. Bluhm, *Signs and Symbols: Their Design and Meaning*. 1998, New York, USA: Watson-Guptill Publications.

[15] Ganek, A.G. and T.A. Corbi, *The Dawning of the Autonomic Computing Era.* IBM Systems Journal, 2003. **42**(1): p. 5-19.

[16] Goodman, N., *Languages of Art: An Approach to a Theory of Symbols*. 1968, Indianapolis: Bobbs-Merrill Co.

[17] Grau, G., J. Horkoff, E. Yu, and S. Abdulhadi. *i* Guide 3.0*. (last updated August, 2007; accessed February 10, 2009); Available from: http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=67.

[18] Green, T.R.G. and M. Petre, *Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions'*

framework. Journal of Visual Languages and Computing, 1996. **7**: p. 131-174.

[19] Gurr, C.A., *Effective Diagrammatic Communication: Syntactic, Semantic and Pragmatic Issues.* Journal of Visual Languages and Computing, 1999. **10**: p. 317-342.

[20] Hitchman, S., *The Details of Conceptual Modelling Notations are Important - A Comparison of Relationship Normative Language.* Communications of the AIS, 2002. **9**(10).

[21] Irani, P. and C. Ware, *Diagramming Information Structures Using 3D Perceptual Primitives.* ACM Transactions on Computer-Human Interaction, 2003. **10**(1): p. 1-19.

[22] Kaindl, H., S. Brinkkemper, J.A. Bubenko, B. Farbey, S.J. Greenspan, C.L. Heitmeyer, J.C.S.P. Leite, N.R. Mead, J. Myopolous, and J. Siddiqui, *Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda.* Requirements Engineering, 2002. **7**: p. 113-123.

[23] Larkin, J.H. and H.A. Simon, *Why a Diagram is (Sometimes) Worth Ten Thousand Words.* Cognitive Science, 1987. **11**(1).

[24] Lynch, M., *The Externalized Retina: Selection and Mathematization in the Visual Documentation of Objects in the Life Sciences.* Human Studies, 1988. **11**: p. 201-234.

[25] Mackinlay, J., *Automating the Design of Graphical Presentations of Relational Information.* ACM Transactions on Graphics, 1986. **5**(2): p. 110 - 141.

[26] Maiden, N.A.M., N. Kamdar, and D. Bush. *Analysing I* System Models for Dependability Properties: The Uberlingen Accident.* in *REFSQ'06*. 2006. Luxembourg.

[27] Masri, K., D. Parker, and A. Gemino, *Using Iconic Graphics in Entity Relationship Diagrams: The Impact on Understanding.* Journal of Database Management, 2008. **19**(3): p. 22-41.

[28] Miller, G.A., *The Magical Number Seven, Plus Or Minus Two: Some Limits On Our Capacity For Processing Information.* The Psychological Review, 1956. **63**: p. 81-97.

[29] Moody, D.L. *Complexity Effects On End User Understanding Of Data Models: An Experimental Comparison Of Large Data Model Representation Methods*. in *Proceedings of the Tenth European Conference on Information Systems (ECIS'2002)*. 2002. Gdansk, Poland.

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

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

[32] Nordbotten, J.C. and M.E. Crosby, *The Effect of Graphic Style on Data Model Interpretation.* Information Systems Journal, 1999. **9**(2): p. 139-156.

[33] OMG, *MOF Core specification*. 2006: Object Management Group.

[34] Peirce, C.S., *Charles S. Peirce: The Essential Writings (Great Books in Philosophy)*, ed. E.C. Moore. 1998, Amherst, USA: Prometheus Books.

[35] Petre, M., *Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming.* Communications of the ACM, 1995. **38**(6): p. 33-44.

[36] Siau, K., *Informational and Computational Equivalence in Comparing Information Modelling Methods.* Journal of Database Management, 2004. **15**(1): p. 73-86.

[37] Tufte, E.R., *Beautiful Evidence*. 2006, Cheshire, Connecticut, USA: Graphics Press.

[38] Wheildon, C., *Type and Layout: Are You Communicating or Just Making Pretty Shapes?* 2005, Hastings, Victoria, Australia: Worsley Press.

[39] Winn, W.D., *Encoding and Retrieval of Information in Maps and Diagrams.* IEEE Transactions on Professional Communication, 1990. **33**(3): p. 103-107.

[40] Winn, W.D., *An Account of How Readers Search for Information in Diagrams.* Contemporary Educational Psychology, 1993. **18**: p. 162-185.

[41] Yu, E., *Modelling Strategic Relationships for Process Reengineering (PhD thesis)*. 1995: Department of Computer Science, University of Toronto.

[42] Yu, E. *Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*. in *Proceedings of the 3rd IEEE International Conference on Requirements Engineering (RE'97)*. 1997. Washington D.C., USA.