Formality in Sketches and Visual Representation: Some Informal Reflections

Alan F. Blackwell
University of Cambridge Computer Laboratory
afb21@cam.ac.uk

Luke Church
University of Cambridge Computer Laboratory
lec40@cam.ac.uk

Beryl Plimmer
Auckland University Dept of Computer Science
beryl@cs.auckland.ac.nz

Dave Gray
XPLANE
dgray@xplane.com

Abstract

This paper provides an overview of the ways that sketches function as informal representation tools, especially when used in design contexts. We then consider the tension between this essentially informal practical function of sketches, and two different factors that drive toward formalization. These are 1) the need for a computational interpretation, and 2) the desire to specify visual formalisms as scientific, critical or technical tools.

1. The Thesis

Sketches are informal. Or at least, an informal definition would suggest that they are. Nevertheless, we wish to question the several respects in which formality might be judged, and pick apart the characteristics of some sketching enterprises according to each of these. We do so in an informal way, as a (sketchy) starting point for workshop discussion, rather than a definitive exercise.

1.1. Formal Intention

This, then, is the first of the respects in which formality might be judged: formality of intention. Many enterprises are undertaken without a clear formulation of their objectives, and indeed the writing of this paper was one of them.

1.2. Formal Connotation

A second respect in which formality might be judged is formality of connotation. At the time of writing, it is May Week in Cambridge. This week in mid-June (sic) is renowned for the balls at which students of the ancient Colleges celebrate the end of the year at huge circus-like parties lasting until dawn. They walk into town in evening dress – ballgowns and "black tie" (dinner suits /tuxedos) – that are described as "formal", and certainly carry the connotations of formality as far as onlookers are concerned. Despite the connotations, May Balls in fact are far less formal than the project meeting at which I (Alan) spent the evening, in jeans and T-shirt. Unless something goes seriously wrong, this paper has connotations of formality – it will be distributed on clean paper, printed in a typeface that emulates professional publication.

1.3. Formal Description

A third respect is formality of description. Are the elements of the representation clearly differentiated from each other: both from other elements within the same representation (they are bounded and separate) and from alternative elements that might have been chosen but are not included? "Elements" in this case refers not only to visual symbols, units or entities, but also to the graphical relations between them, including arrangement in the plane, topological relations or grouped graphical attributes. The visual representation of this paper can certainly be described in terms of such formal elements. Each letter is clear and distinct, as are the words, and their organisation into sentences and paragraphs.

1.4. Formal Interpretation

A fourth is formality of interpretation. Are there rules describing the things that the reader should do differently in response to differences in the representation? A particular concern of ours is the situation in which the "reader" is a machine. The formal elements that appear on the screen as I write

VL/HCC Workshop: Sketch Tools for Diagramming
Herrsching am Ammersee, Germany
15 September 2008
Editors: Beryl Plimmer & Tracy Hammond

correspond to particular things that the computer should do. In the case of a programming language, the rules and actions might be relatively complex. In the case of a word processor, they are relatively straightforward. But in both cases, a human reader has many opportunities to interpret a representation according to different rules, possibly rules that the creator had not intended.

1.5. Sketches and Computation

Where visual representations are created or captured by computers, there is a strong tendency toward formality. At the most crude level, computers follow interpretative rules as described in the fourth respect above. However, the prevailing use of computers for business and scientific purposes means that their visual styles have tended to adopt the connotative trappings of business (I see “formal” typefaces, shiny icons and clean window borders on the screen in front of me). Furthermore, user interfaces encourage descriptive formality, in order that the user can anticipate predictable correspondences between the screen contents and computer behaviour. This last might be modified, but we are especially concerned with the question of intention. Are there certain kinds of human endeavour that are not adequately supported by computer processing of visual representations, because they have been inappropriately formalised? We believe that this question is related to that of intention. We also believe that these four aspects of formalisation are regularly conflated, so that freedom of intention is restricted as a result of representational choices, technical implementations, and unnecessary connotations.

2. Related Literature

2.1. Design Sketching

Sketches often arise in technical literature as one of the phases or products of design activity. If considered sufficiently broadly, any novel artifact (including paintings, business processes, drugs, musical compositions) can be considered to arise from a design process, and it is possible to identify particular tools that facilitate “sketching” of some kind within that process.

A project that investigated design experiences across a very wide range of design domains was able to draw systematic comparisons between these different functions of sketching in different kinds of design [8]. This project found that sketches are used as depictions of potential objects in idea generation, but also as thinking aids for reasoning about abstract concepts. They are used in those domains, such as software design, where there is no pictorial description of the product, but also in more visual design domains such as engineering design, to sketch out abstract properties. Finally, sketches are a vital means of communicating design ideas to users, or among teams, in many different domains. The following sections draw out some of these common themes in terms of prior research.

2.2. The Pencil in Practice

A common concern, as sketching on paper has been replaced by computer drawing tools, has been the deficiencies of the mouse and keyboard as “craft tools” by comparison to the pencil. This problem became evident most early with the advent of mass-market “desktop publishing”, which threatened to completely replace the traditional craft skills of the typographic designer. Hewson’s study of the practices of typographic designers, embedded in the materiality of paper and the versatility of the pencil, recorded many professional sketching practices that are still not supported by computer tools [14], Black similarly drew attention to the practices of graphic designers [1].

This concern with the physical craft of sketching, and the possibilities of the pencil as an informal drawing tool, continue to be a topic of research for those investigating the relationship between computer tools and design [17]. If the full range of interpretations supported within the craft tradition are to be maintained, computer tools must either emulate or substitute for these, at both the level of construction and interpretation.

2.3. Perceived Finishedness

Black’s study of graphic design drew attention to the fact that design clients may be reluctant to criticise designs that look too finished. This is a well-known advantage of low-fidelity prototyping, which encourages users to offer feedback on designs that are clearly work-in-progress because they appear on the computer screen [27]. Other work on fidelity includes [7]. This attribute of sketches has been described more generally by Bresciani et. al. [4] as “Perceived Finishedness”, a generic usability dimension that can be chosen or adjusted in any visualisation for collaborative work.

An extreme solution to this problem is to take finished computational models, and render them to look as though they have been created by hand, with uneven lines and approximate junctions. The results are “sketch renderers” [25], which pretend to be unfinished, even though the underlying model could alternatively have been rendered using photo-realistic ray tracing. Sketch renderers adjust the connotations of formality, without necessarily modifying the formality of the representation in any other respects.

2.4. Imagery and Ideation

Many people think that imagination is fundamentally pictorial (as the word indicates: imag[e]ination). Philosophers (Aristotle, Kant) and poets (Keats, Coleridge) describe mental images as the source of creativity. Even engineers, if asked, are able to report the pictorial form of their technical imagination [21]. However an essential attribute of creative problem solving is the ability, not just to form an image of the solution, but also to change perspective and imagine alternative solutions. There is some psychological evidence suggesting that mental images have sufficient interpretive content integrated into them that they cannot be re-interpreted [5]. In this case, externalising the image by sketching it provides a mechanism for seeing it differently, in a way that could not be done with the internal image alone.

In order to achieve this effect, it is essential that the sketch should not have semantics overlaid on it, but has alternative interpretations – that it is ambiguous. Studies of design professionals such as architects show that they sometimes make sketches that are intentionally messy and fuzzy, so that looking back at what you have just drawn supports re-interpretation [12]. The process of repeated sketching and re-interpretation has been described by Schon as a conversation between the designer and the sketch. These are all examples where sketching exhibits different formality of intention, supporting activity in which indeterminacy of outcome is an essential feature.

2.5. Drawing Boundaries

Determinacy is also a function of the representational convention itself. As noted by Goodman [13], some languages of art are “autographic” – they have their own identity, and there is a clear distinction between the original and a copy, as in an oil painting – while others are “allographic”, consisting of discrete symbols that can be copied (without loss) but are then interpreted through a performance, as in a musical score. Although most computer representations are allographic in Goodman’s terms, sketches seem to share the properties of the autographic. Rich interpretations of artworks also arise through the possibility of multiple readings, not solely by the creator, but as multiple possibilities for interpretation by readers [9].

However, we are also concerned with the consequences of defining representational boundaries within computer systems [2]. As noted by Bowker and Star, deterministic classifications can be socially inappropriate in many situations [3]. Even within the framework of Western epistemology, there is ground to question the universality of the way we divide the world up [24]. If there are further reasons to question the appropriateness of Western epistemology, then sketch-like visual representations may be the only way to avoid the boundaries inherent in most computational representations [26]. In these cases, it is formality of description itself that may need to be adjusted.

3. Case Studies

We now present two case studies to which these analytic perspectives can be applied. These were selected, not because they are any more or less relevant than others we might have chosen, but simply because they were the projects that happened to be on our (Alan and Luke’s) desks on the day that Beryl proposed we contribute to this workshop. We are reasonably confident that anybody conducting research into the use of visual representations, in any domain, would find it equally easy to identify similar problems of formalisation in their own problems.

3.1. Dave Gray’s Q-Tools

Dave is founder and principal of XPLANE, a consultancy that offers services related to the clear expression of ideas in various media. He develops those services in collaboration with a broad network of correspondents, and through a variety of online discussion and dissemination forums. In a recent initiative, he set out to formulate a set of strategies for developing arguments and descriptions that he calls “Q-tools.” Each of the Q-tools is a specific kind of logical operation, for example: *Prism* - a question that divides information into smaller groups; *Razor* - a question that divides information into two categories, based on relevance; *Flanker* – a question that seeks patterns or ideas that are similar.

Dave initially consulted us to invite comment. However, rather than focus on the Q-tools themselves, we were interested in the process by which he had formulated them, and the intuitions that underlay that process. In particular, Alan noticed that his descriptions of ideas and their transformations seemed to describe an implicit "space" or "landscape" within which these transformations occurred. In much the same spirit as that of Petre and Blackwell [21], we invited him to draw this conceptual design space. His drawings, as shown in Figure 1, can be regarded as sketches that externalise his previously internal and intuitive imagery. (Of course, we don't question that the process of externalisation, in response to our request, might well have modified the intuition that we wished to investigate).

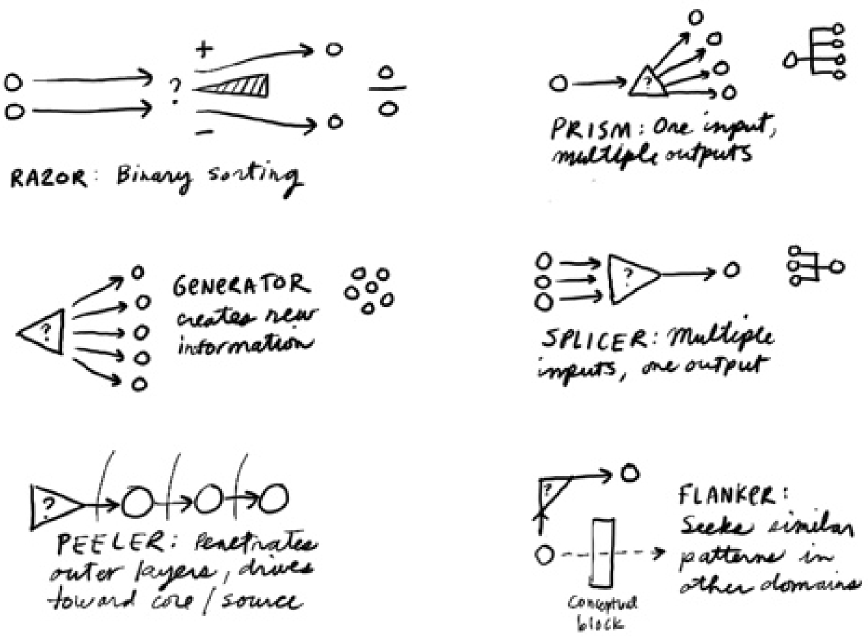

Figure 1 – Dave Gray's visualizations of his Q-tools

In considering these sketches according to the aspects of formalisation described above, Alan made the following observations:

Dave's intention is relatively informal. He is exploring the concept of Q-tools, and has invited comment (it seems) partly in the spirit that he is willing to take them to some other place if the collaborative process suggests that. In particular, he seems to be playing around with possibilities of how the Q-tools might provide him with new business opportunities, and suggests a number of speculative scenarios for how the Q-tools might turn out to be of value to people.

His initial description of the Q-tools was an informal one, drawing on metaphors that are not necessarily associated with rigorous philosophical analysis. It was these that alerted Alan to the fact that this was a case of mental images underlying a creative design process, and led him to ask for some drawings.

The drawings themselves are sketchy in their style. As a professional graphic practitioner, Dave is presumably able to externalise rapidly, and probably produced these images quickly. This was both efficient (he didn't waste a lot of time catering to academic curiosity), and also carried a clear connotation, as discussed above, that we should be free to propose modifications. Furthermore, that connotation allows him to distance himself from the contrast between the initially speculative motivation, and the possibly far-reaching ambition of producing a system of thought.

It is this final aspect that offers the most interesting tension in reading the formality of Dave's enterprise. Although based on imagistic metaphors, and rendered in a sketchy visual style, the Q-tool drawings have a strong resemblance to electronic schematics, the electrical schematization of the London Underground [11] or to the box and wire diagrams that are so common in software engineering [18]. In this respect, the ultimate objective is a formal one, to support formality of description.

Indeed, the Q-tools offer encouragement to their users, to formalise (or at least to make more rigorous) their own conceptualisations. One consequence may be clarity of thinking, the principle objective in much of Dave's work. Another unintended consequence might be that these more formal descriptions are open to computational interpretation, in the manner provided by Beryl's computational sketching tools. Her initial response to the (verbal form of) the Q-tools was that these might not be suited to her systems, which tend to assume box-and-wire representations as the basic visual formalism. As I (Alan) write this paragraph, Beryl has not seen the sketches, but my expectation is that she will find the Q-tools more amenable to computer implementation than she initially thought. We leave it to the discussion/response section of this paper for her to reply, but first we proceed to a second case study in the formalisation of visual representations.

3.2. Computational Biology

Computer Scientists and Biologists are increasingly collaborating in modelling complex biological systems. Initially this focussed around data-mining the large volumes of text from genome sequencing projects, but recently much emphasis has been placed on 'Systems Biology', attempts to characterise biology not as lone entities, but as interacting systems.

An example of Systems Biology is the study of regulatory networks. These are networks of interacting biological entities (RNAs, proteins etc.) that increase or decrease each others' activity. Formality in the descriptions of such systems varies widely. Fig 2 shows a diagram of such a regulatory network, with the implicit semantics overlaid in blue.

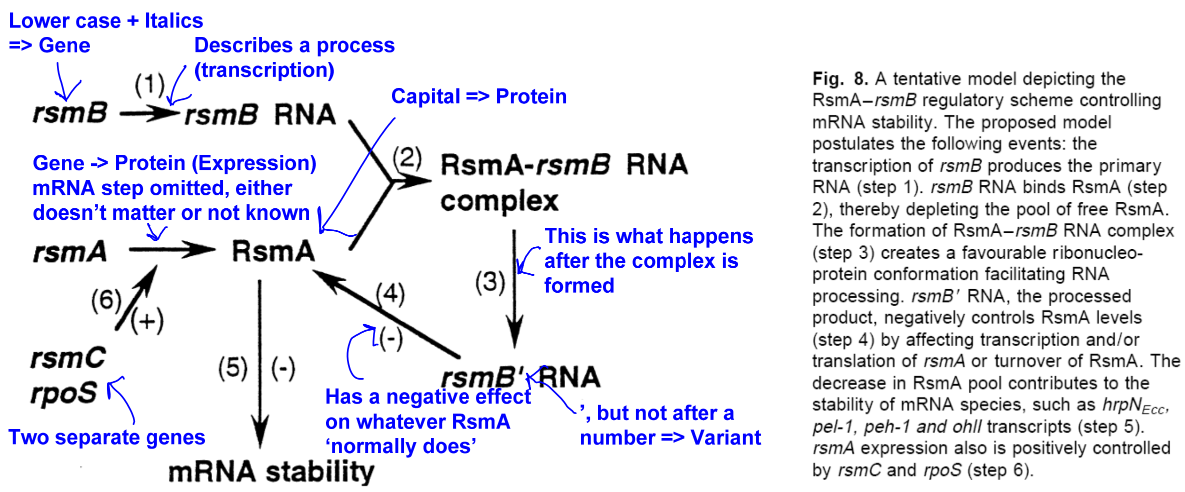

Figure 2 - Diagram and Caption from Liu et al. (1998), additional, implicit semantics overlayed in blue (not in original). The reader is not expected to understand the details of the content, but rather to note the difference between the implicit and explicit semantics.

The original diagram could then be considered to be formal in Connotation and Description, but only partially in Intention and Interpretation. This is well supported by the social processes; such diagrams are commonly used as summaries with the text of the paper providing the details of each of the arcs and nodes.

However, computer scientists are often critical of such informal semantics. Indeed the 'Formal Methods in Systems Biology' conference [10] that Luke recently attended was deliberately named to counter-pose informalities [of interpretation] in descriptions of biological systems. However in doing so, the title of the conference confused 'formality of interpretation' in the sense of 'visual formalisms, Small Matter of Programming' [19] with the more conventional meaning of 'formal methods' in computer science, which attempts to separate and analyse the syntax independently of the semantics[1].

As might be anticipated, the desire to formalise the interpretation is now affecting the formalism of intention. The complex, socially constructed, goals of biology in the pre-genomic era are being complemented (if not partially replaced) by a need to

[1] J. Fisher & D. Harel. Discussion at First Int Workshop on Formal Methods in Systems Biology, Cambridge UK, 4 Jun 2008

provide specific information to improve the computational accuracy of the models. Accurate temporal and quantitative data, once considered too inconvenient to collect, are now needed more urgently to feed the computational models.

Formal knowledge representations (in intention and interpretation) are taking hold, with Gene Ontologies, and Standardised markup languages[15]. Here there is open conflation of formality of intention, description and interpretation.

The challenge that we face is no less than how do we design ‘abstractions for biology’ in this increasingly formal, but largely unknown domain? What part does sketching have to play in mediating between the (often informal) social process of working biologists, and the formal demands of a computer system?

4. Discussion / Responses

This paper is the result of long-distance discussions, and closes with two responses – first from Dave, responding to our analysis of his intentions, and then from Beryl, drawing these issues back to the theme of the workshop.

4.1. From Dave

Any investigation begins with guesses, often accompanied by sketches that are, essentially, groping in the dark. Through iteration and conversation the ideas take form. The act of writing or drawing entails commitment. Writers begin by putting their thoughts onto paper, which they then refine over time. At some point the thoughts are final enough to be formalized in some way, by publishing for example. But in actual fact this iteration never really ends, as ideas will always continue to evolve, and so we end up with editions, commentaries, citations, etc.

This may sound obvious but the entire body of scientific thought is a work in progress.

At the same time notation systems must be formalized to be useful. The formalization of the notation system seems to be a useful constraint when refining or clarifying initially "fuzzy" ideas. For example, a Venn diagram has a useful rigidity because it forces decisions about what something "is" or "is not."

So what I am searching for is that ideal set of constraints which will force rigor into my thinking about information flows. Ideally this will result in some kind of notation system that can serve as a framework for teaching and thinking about information visualization – the visual language that can be used to describe visual language?

4.2. From Beryl

We have, perhaps, by this point convinced you, if indeed you needed convincing, that there is a problem. Our computers are poorly designed to support informal representations, yet informal representations are a fundamental problem solving and communication tool that humans have used for the last few thousand years.

My concern is that computers are rapidly taking over from old fashioned pens and paper. Schools encourage students to prepare most of their work on these cumbersome machines. My students struggle to handwrite continuously for two hours in exams. This is not a criticism of the computer revolution. Computers, I think, are absolutely the best thing that people have ever invented. But they are not a good fit for some of the tasks for which people are using them.

When I look at Dave’s sketches I see standard diagram components. Nodes representing ideas, connectors showing some flow, triangles to join the flows together and a few notes. And indeed Alan is correct I could reasonably quickly build a recognizer to recognize the components for each of these diagrams in InkKit [] (figure 3). Notice ‘each’, while we humans can look at the set and using heuristics quickly work out Dave’s encoding scheme, our recognition engines are not so smart.

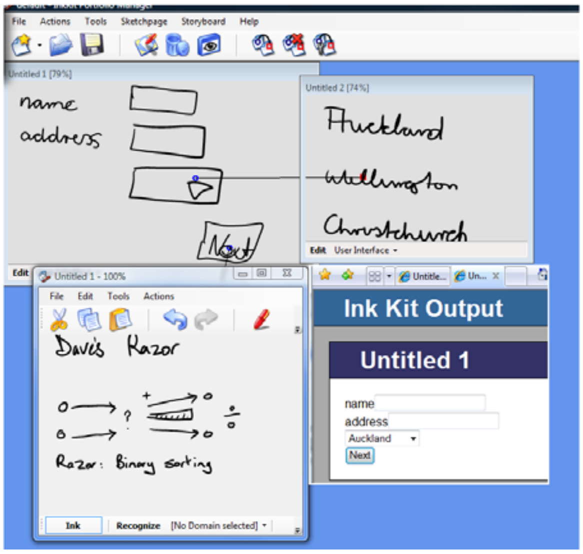

Figure 3: InkKit [22] an extensible sketch toolkit showing a UI design sketch and the generated output from the sketch and a reproduction of one of Dave’s diagrams

A little aside, the observant may have noted that I hadn't seen Dave's sketches when he was writing his section. We are scattered across the globe and our discussion has been hosted by a very well known online document collaboration system. Dave referred to his sketches that either he hadn't added to the discussion or if he had added, I couldn't find/see.

There is a fundamental divide in all our current computer systems between text and non-text. Text editors are, perhaps, the most frequently used software applications (word processing, on line forms, programming IDEs, etc, etc) and they do a fantastic job at word wrap, spell check and fonts. But they do a dreadful job at managing the non-text elements. Likewise drawing tools are clumsy when dealing with text elements. Most of us are probably Microsoft Word or Latex disciples – and have a passionate dislike for the other creed. Much of the debate around which is best is about their picture handling.

Ink annotation, the old fashioned red pen we used to decorate our students work with, is even more problematic. It is supported in some document tools (MS Word for example) – but it is extremely easy to break the ink reflow. Furthermore, adding ink annotation capability is extremely difficult. We have succeeded in standalone applications, but have tried and failed three times in IDEs [6]. Wouldn't it be cool to be able to write notes with a pen on code files?

Back to our example diagrams – if the computer did recognize Dave's sketches, what then? A good start would be to treat them at least as well as we do words – i.e. keep the parts together and properly arranged. Dave asserts that he wants a formal notation to increase the rigor of his thinking, but the cost of formalization is often constraint – this may not be helpful. One last thing to draw from Dave's sketches – I suspect he drew them on paper and scanned the page. We do have stylus active surfaces, but they are a very poor imitation of paper and pen/pencil. There is little pleasure in doodling on a tablet screen.

To the Biological System example – again we have nodes and connectors. This time the nodes are text and there is an enormous amount of secondary notation on the diagram that is highlighted for us. (If you printed this document out on a B&W printer you probably wondered about the reference to 'blue' parts of the diagram. Did you make a guess? The blue parts are annotations that have been added to point out the implicit semantics. They consist of informal arrows, you probably got those, and text explanations at the tail end of the arrows – if you looked carefully at the diagram you would have got those. But if the annotations had been hand written a glance would have given you the information.) If the diagram was hand-drawn how good a job could our diagram recognizers do? Sadly, not very well, we are working on the problem of separating text from non-text (that basic divide again) and can now achieve about 80% accuracy [20]. I would posit that the average 3 year old could do better.

So where are we today? We have an increasing use of computers for document creation and distribution. However the toolkit is half empty. Could Leonardo da Vinci have drawn his man on a 2008 desktop computer? Could the monks of Ireland have created the Books of Kells? They had better physical tools than our *clever* computers, and their minds were far more advanced than our most exciting AI tools of today.

Acknowledgements

Luke Church's research is supported by the Eastman Kodak company.

10. References

[1]. Black, A. (1990). Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers. *Behaviour and information technology*, 9(4), 283-296

[2]. Blackwell, A.F., Church, L. and Green, T.R.G. (2008). The abstract is 'an enemy': Alternative perspectives to computational thinking. Submitted to PPIG 2008.

[3]. Bowker, G.C. and Star, S.L. (1999). Sorting Things Out: Classification and Its Consequences. MIT Press.

[4]. Bresciani, S., Blackwell, A.F. and Eppler, M. (2008). A Collaborative Dimensions Framework: Understanding the mediating role of conceptual visualizations in collaborative knowledge work. Proc. 41st Hawaii International Conference on System Sciences (HICCS 08), pp. 180-189.

[5]. Chambers, D and Reisberg, D. (1985). Can mental images be ambiguous? *Journal of Experimental Psychology: Human Perception and Performance* **11**(3): 317-328

[6]. Chang, S.H.-H., Chen X., Priest R. and Plimmer B. Issues of Extending the User Interface of Integrated Development Environments. in Chinz. 2008. Wellington: ACM.

[7]. Coyette, A., Kieffer, S. and Vanderdonckt, J. Mult-fidelity Prototyping of User Interfaces. (2007). In *Proceedings of Interact'2007, Springer-Verlag 149-162*

[8]. Eckert, C.M., Blackwell, A.F., Stacey, M.K. and Earl, C.F. (2004). Sketching across design domains. In *Proceedings of the Third International Conference on Visual and Spatial Reasoning in Design (VR'04).*

[9]. Empson, W (1930). Seven types of ambiguity. London: Chatto and Windus.

[10]. Fisher, J. (2008). Formal Methods in Systems Biology. Springer Lecture Notes in Computer Science. Vol. 5054

[11]. Garland, K. (1994). Mr. Beck's Underground Map. Capital Transport Publishing.

[12]. Goldschmidt, G. (1991). The dialectics of sketching, *Creativity Research Journal* **4**: 123-143.

[13]. Goodman, N. (1968). Languages of art: an approach to a theory of symbols. Indianapolis: Bobbs-Merrill.
[14]. Hewson, R. (1994). Making and marking. PhD. Thesis, Milton Keynes, Open University.
[15]. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A et al.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 2003, 19:524-531
[16]. Liu, Y., Cui, Y., Mukherjuee, A. and Chatterjee, A.K. (1998) Characterisation of a novel RNA regulator of *Erwinia cartovora* ssp. *cartovora* that controls production of extracellular enzymes and secondary metabolites, Molecular Microbiology 29(1), 219-234
[17]. McCullough, M. (1998). Abstracting craft: The practiced digital hand. MIT Press.
[18]. Morris, S.J. and Gotel, O.C.Z. (2006). Flow Diagrams: Rise and fall of the first software engineering notation. In Proc. Diagrams 2006, pp. 130-144.
[19]. Nardi, B.A. (1993). A Small Matter of Programming: Perspectives on end-user computing. MIT Press.
[20]. Patel, R., B. Plimmer, et al., Ink Features for Diagram Recognition, in SBIM. 2007, IEEE: California
[21]. Petre, M. & Blackwell, A.F. (1999). Mental imagery in program design and visual programming. International Journal of Human-Computer Studies, 51(1), 7-30.
[22]. Plimmer, B. and I. Freeman (2007). A Toolkit Approach to Sketched Diagram Recognition. HCI, Lancaster, UK, 205-213.
[23]. Schön, D.A. (1983) The Reflective Practitioner: How Professionals Think in Action, Basic Books, New York, NY.
[24]. Smith, B. C. (1996). On the Origin of Objects. MIT Press, Cambridge, MA.
[25]. Strothotte, C. & Strothotte, T. (1991). Seeing between the pixels: Pictures in interactive systems. Berlin: Springer.
[26]. Verran, H. and Christie, M. (2007), Using/Designing Digital technologies of Representation in Aboriginal Australian Knowledge practices, Human Technology Vol 3 (2), pp. 214-227.
[27]. Yeung, L., Plimmer, B., Lobb, B., Elliffe, D. (2007). Levels of Formality in Diagram Presentation, in OzCHI 2007. 2007, ACM: Adelaide. p. 311-318.