Editorial postscript

> … the disease store in my body for a long time and now explose. Just like the virus in the computer.

In the 1970s, when recombinant DNA research hit the media, there was fear about the potential for human cloning and for the release of a new virus that no one could counter. Biologists in this area began to adopt the language of computing to talk about their research, as a way of trying to objectify their work and defuse media fears. Later, computer scientists, perceiving biological systems as "paradigms of adaptive excellence" (Wray 1997), started using terms from DNA evolution research to talk about their own discipline – the "reverse engineering of bioinformatics" (BBSRC 2001) – which has led to the emergence of fields such as genetic algorithms, genetic programming and evolutionary computation (cf. Lundh *et al.* 1997).[1] Biology has now become one of the dominant metaphors for CS, and is providing a framework for understanding and constructing complex computational activity.

Software developers have long been aware of the usefulness of interface metaphors to help users make sense of the medium, as a way of showing the unknown or unfamiliar via the known or familiar (cf. Carroll & Mack 1985; Carroll *et al.* 1988). Thus, the computer has been purveyed *inter alia* as a 'servant' (e.g. by the personalisation of program names), a 'tool', a 'machine', a 'workplace', a 'filing cabinet', a 'toy' (Lawler 1987). The disadvantage of the not infrequent lack of ease of transfer across cultures is seen as outweighed by borrowed familiarity of the metaphors (cf. Mar-cus 1998). Smilowitz (1996), however, shows that there is surprisingly little research to support the popular belief that metaphors in user inter-faces facilitate performance, and cautions that metaphors are not all equally effective, and some no better than non-metaphoric interfaces:

> Metaphors are ubiquitous in the user interfaces of today's computers. Software designers are incorporating metaphors into a variety of software from operating systems to information retrieval applications. Technological advances have made more and more realistic depiction of these metaphors possible. However, technological feasibility does not insure psychological utility. … [N]ot all metaphors are good. Good metaphors facilitate performance, and poor metaphors are no more effective than non-metaphoric interfaces.

[1] I am grateful to Nick Noakes for leading me to this path of development.

Metaphors may not be efficient, or even appropriate for some users. The connotative value of a metaphor may engender emotional reactions, and take on significations far beyond being a simple aid to learning. Keniston (1999), for example, argues that "the language in which computing takes place is a critical variable in determining who benefits, who loses, who gains, who is excluded, who is included – in short, how the Information Age impacts the peoples and the cultures of the world", and stresses the relationship of language to power, wealth, privilege, and access to resources.

One cultural mismatch, highlighted by Grundy (1996), is that computing can be seen to be using metaphors largely drawn from a male environment, male values and male experience. Grundy (2000) even challenges the labelling in the English-speaking world of computing as a 'science', claiming the motivation to be political rather than intellectual, to preserve computing in a predominantly male domain. Condon (2000) gives an example of such male orientation:

> The casual use of the term 'abort' in computing is an example that many people might appreciate. Although 'abort' has the wider meaning of abandoning an action, in popular use it almost always refers to the abortion of a pregnancy. A comparable 'male' term might be to replace the error message 'the command could not be completed due to insufficient memory' with 'your command was impotent due to insufficient memory'. This would be technically correct but many men might feel uncomfortable with it.

What one can express, or what can conceive of wanting to express, depends upon the tools available (Travers 1996). The creation of new metaphors enables the expression of the otherwise ineffable.

A common metaphor in computing is that of the 'agent' (Laurel 1993; Erickson 1997): many programs (abstractions or processes) are anthropomorphised, "the most common kind of metaphor in formal computer discourse" (Johnson 1994: 100). Johnson gives examples of disk 'servers' said to 'police' their 'clients', 'allowing' access to the disk, and 'trusting' their 'clients'. The agent metaphor, therefore, suggests a particular model of what the program is, what it is capable of and how it functions, in animate terms, and how it relates to the human user. This has been shown to lead to anthropomorphic responses to computers (Nass & Steuer 1993; Nass *et al.* 1995), where human emotions are projected onto the electronic world – as in a reaction such as, 'The computer is angry' – reminiscent of Ruskin's (1856) *pathetic fallacy*. Users' understanding of the system is thus deliberately led to be different from how the system functions and what it is achieving: “they read metaphor and see literal truth” (Johnson 1994: 99; cf. Condon 1993). Often expectations cannot be met.

Ackerman (1994) offers the examples of the term *memory* which “suggests that capabilities present in social or human reality (such as memory) are also present in the technology” and *digital library*, a metaphor which “constricts our understanding of social reality, and in doing so, may eventually change the social reality itself”.

> Because they deal with highly abstract and arbitrary realities … computer scientists are called on to *name* a very large number of things that may not have obvious designations. Those with simple analogies to natural phenomena are often designated metaphorically. Others may be assigned complex multipart descriptives that are sometimes rather poetic …
>
> (Johnson 1994: 97)

The substitution of the concrete for the abstract, investing analytic or abstract relationships or characteristics, systems, or constructs with objective reality, as though they were concrete or material entities, was identified by A. N. Whitehead (1925) as the *fallacy of misplaced concreteness*. While the hypostatisation of immaterial abstractions may indeed help humans make sense of given phenomena, there is a fallacy in neglecting the degree of abstraction involved when an objectified entity is considered merely insofar as it exemplifies and simplifies categories of thought, or when, as with *digital library*, the metaphor restricts the social or human phenomena to merely what is possible through technology. However, one of the principal problems for computer users is fear of the computer itself. Many people are stressed by computers, and discourse derived from the anthropomorphic metaphor is helpful in making them more comfortable with the medium. Cividalli (1996, cited in Danet 1997) shows that the tensions inherent in the computer environment find expression in the metaphors people use about computers:

> … the computer is an inanimate machine, fully-programmed and, supposedly, completely subject to human control. Yet, we speak of computers as having their a will of their own, and we ascribe to them other human attributes such as friendliness. We name parts of the computer’s hardware or software after human or animal body parts (memory, mouse, bugs); computers can be “ill,” “infected by viruses,” or “die.” We regard computers as independent actors with intentionality: we get angry when they “go crazy,” and so on. This type of analysis helps us understand the tensions between controlling the machine and being controlled by it. (Danet 1997)

Structural interface metaphors such as ‘trash’, ‘file’ and ‘folder’ icons are familiar within the ‘desktop’ setting. More recently, other metaphors for computers have included the ‘book’, the ‘room’, the ‘film’, the ‘studio’, the ‘landscape’, and ‘lifestreams’ – the last based on the idea of adapting the machine to humans rather than humans to the machine. New metaphors attempt to imitate the way the mind conceptualises the world, by time and location, rather than the organisation of information imposed by the paper medium (Madsen 1994; Young 1997). General interface metaphors have led to the instantiation of metaphors of language, e.g. verbs such as *cut*, *kill*, *move*, *paint*, *paste*, *surf*, or nouns such as *bootstrap*, *bookmark*, *button*, *clipboard*, *engine*, *front end*, *garbage collection*, *handshake*, *hook*, *information superhighway*, *mouseover*, *pipeline*, *pointer*, *port*, *platform*, *stack*, *tools*, *wastebasket*, *windows*, *worm*. Such ‘semi-technical’ items are typically drawn from the general vocabulary stock, adapted for use within the context of a specific discipline, as computing in this example from *Netwatcher* (October 1999):

> The Internet was the input conduit for the transactions from the outside. In semi-technical terms, we put a CGI gateway on an internal LAN, linked it to the host via TCP/IP, and then provided that internal LAN with a fire-wall connection to the Internet.

Coxhead & Nation (1999) call this ‘academic vocabulary’, and stress that mastering such lexis is important for those learning English for academic purposes because it is common to a wide range of academic subjects; it accounts for a high proportion of words in academic texts; many of the items tend to be not well known by second language learners; and teachers can help learners gain control of it. (They cite *execute* (*a program*), *scroll* (*down*) and *paste*, which can occur outside the area of CS, but have specialised meanings within it.) In this, they echo Skehan’s (1981: 117) premise that vocabulary teaching in English for specific purposes (ESP) should emphasise such, what he terms ‘sub-technical’, words.

‘Sub-technical’ or ‘semi- technical’ (the term preferred by Jacqueline Lam in this volume) are themselves semi-technical terms in linguistics, used in a spectrum of, sometimes contradictory, senses,[2] e.g.:

St John (1996: 5): “a relatively well-agreed core of … lexis and grammar that is widely used in academic scientific and technological subjects”

Luzón Marco (1997): “items which are not specific to a certain field of knowledge but are used in a distinctive way and for specific functions in specialized texts”

Dudley-Evans & St John (1998: 83): “general vocabulary that has a higher frequency in a specific field” and “general English words that have a specific meaning in certain disciplines”

Fang (1999): “words common to all scientific disciplines”

EFL Tech – France (n.d.): “a kind of general-purpose technical English”

Batchelor & Offord (2000): “the sort of vocabulary that every educated person ought to be familiar with”

ALADIN (n.d.): “not the technical vocabulary of a particular academic discipline. [It] is used across all academic disciplines to teach about the content of the discipline.”

Gibbon (2000): “words and expressions which are part of the general vocabulary and may be ambiguous”

Graedler (2000): “words that are associated with one specific subject area”.

Graedler’s definition would even appear to include what others would call ‘technical’ lexis (cf. the extract from *Netwatcher* cited above). Indeed

[2] Subject specialists use the term ‘semi-technical’ differently from applied linguists – cf. these items, identified as “semi-technical terms” by the author of an article in *Palm OS®* (2000): *appInfo*, *card*, *chunk*, *compaction*, *creatorID*, *feature*, *feature pointer*, *fixed chunk*, *fragmented*, *free list*, *handle*, *localID*, *locking*, *lock count*, *master pointer*, *movable chunk*, *offset*, *ownerID*, *pointer*, *record database*, *record index*, *relocated*, *resized*, *resource database*, *semaphore*, *sorted database*, *store*, *uniqueID*, *unlocked*.

Corbin's (1998) use of 'semi-technical' gives a fairly explicit example of such an overlap:

> There is a great deal of emphasis placed on the buzzword "carbide" at this time. Carbide is a rather generic term that covers a lot of ground, rather like the word "chlorophyll" back in the 1960's, or any other semi-technical word that is turned into an advertising catchword.

The identification of technical and semi-technical vocabulary in linguistic terms provides a useful and pedagogically sound basis for designing materials and activities to facilitate teaching and learning (cf. Strevens 1973; Johnson & Hwang 1983). *EFL Tech – France* (n.d.) offers some general indications:

> All these semi-technical words are also used in general English but they do come up extremely frequently in technical documents, papers or conversations. Some of them are fairly international but most of them, although they look harmless, can cause a lot of trouble to students …
> vocabulary necessary for description: nouns such as top, bottom, amount, edge, level, length, width, stiffness;
> adjectives such as light / lightweight, loose, tight etc;
> vocabulary referring to quality: reliable, low-cost, cost-efficient, affordable, to feature, general purpose, failure, to improve, versatility, etc.;
> verbs used in operation manuals: check, secure, fasten, perform, fit, avoid, supply, fit, insert etc.;
> the basic vocabulary of computing;
> phrases expressing position or motion: upside down, at the back, on the left, across, through, off, back-wards etc.

Note that here, the "basic vocabulary of computing" – with no indication of whether this is perceived to incorporate metaphorical usage – is now assumed to constitute an essential element of the lexical baggage for a student of ESP.[3]

In a series of small experiments conducted with university students in the US, Elissa Smilowitz (1996) endeavoured to determine the effectiveness of metaphorical, as opposed to non-metaphorical, language in web browsers. Her experimental paradigm was based on the *Mosaic* application used to search for information on the web: the terminology for her

[3] The language of CS has received considerable attention over the past decade or so (e.g. Beedham & Bloor 1990; Gotti 1990; Johnson 1994; Bloor 1996; Anthony 1999, 2000; Mey 1997; Gibbon 2000; Tateson 2000; Moore, forthcoming), and recently other corpora of CS language have been reported (Gil Salom *et al.* 2000; Luzón Marco 2000; Peters 2000).

no-metaphor conditions was based on *Mosaic*'s function names, and that of the metaphor conditions based on a library metaphor (cf. Table 1).

Table 1: Function labels used in no-metaphor and metaphor conditions in Smilowitz' (1996) experiments.

***Mosaic* terminology (no metaphor)**	**Library terminology (metaphor)**
World Wide Web	*library*
open URL	*search bookshelves*
window history	*stack of viewed books*
hotlist	*bookmark*
annotate	*post-it notes*
document	*book*
home	*first*
forward	*next*
back	*previous*
clone window	*copier*
save as	*translator*
find in current	*find text*
Internet resources metalist	*reference section*

Smilowitz' subjects received four conditions of interface terminology: text-based – no metaphor, or library metaphor; and icon-based – with library icons or without. All the subjects were told that they would be performing a series of computer-based tasks, and that it would be to their advantage to try to learn as they performed them. The subjects in the metaphor conditions were told that the tasks were based on a metaphor of a library, and that thinking about a library would help them perform the tasks. Subjects in the no-metaphor conditions were simply told that the basic task was to search for information.

On average, subjects made fewer errors with the metaphor interface than with the no-metaphor. Smilowitz found no significant main effect of icon, suggesting that the presence of icons did not appear to improve performance in the metaphor conditions or worsen that in the no-metaphor conditions. (But she notes: "Perhaps, the icons were not compelling enough or meaningful enough to aid in task performance.") However, the evidence is striking for the major effect of the metaphor being provided linguistically, through the terminological function labels.

In a post-test survey, Smilowitz found that her subjects perceived the metaphoric interface as significantly easier to use than the non-metaphoric, which reflected their test performance; they did not perceive a difference due to the use of icons. The metaphor advantage appears, therefore, to be carried in the language, even in an icon-based system.

This is an important finding for those concerned with the linguistics of CS texts. All of the terminology, for example, in Table 1 above, could be considered 'semi-technical' in the terms in which Jacqueline Lam has defined this in her report: lexis and expressions which students may have met in their general reading, but whose specialised meaning in a specific domain they may not know. But it is the vocabulary in the second column of the Table – the metaphorical uses – which native-speakers find to be a more effective and efficient vehicle to aid understanding in a CS environment. Further re-search is called for to investigate whether the same would hold true for non-native speakers. Lam's findings reported here would tentatively sug-gest that metaphorical usage may pose a greater problem than non-meta-phorical usage to non-native speakers, but further experimentation would need to capture differences in the degree of abstraction of metaphor, and the nature of any cultural allusions or other variables inherent in texts.

As a corollary, we may also consider the phenomenon of specialised uses being the primary forms learned, and then met, and misunderstood or not understood, in general texts (cf. Li & Pemberton 1994). William Safire (2001)[4] argues, further, that when a technical word, with a clear meaning in a specialised domain, crosses over into the general language, its meaning can get twisted: "This infuriates the specialists, who see it not merely as a form of linguistic corruption but also as highway robbery from their vocabulary." Safire gives some apt examples:

> Physicists cannot string together a theory to explain why *quantum jump*, which in their world means "a sudden alteration in an atom's energy," and is therefore exceedingly small, has leapt into general public usage with the meaning of "huge change."
>
> Psychiatrists can be seen to approach hysteria when *schizophrenia*, a psychosis often characterized by withdrawal and hallucinations, is bandied about by a public that thinks it means "split personality" and uses *schizoid* to describe any duality.
>
> Neuroscientists wince at the way *congenital*, which to them means "inborn; existing at birth," is stretched by vituperative columnists to a more general "habitual, chronic."
>
> Mathematicians cannot calculate why their *parameter*, "a variable constant used to determine other variables," is confused by laypeople with the quite different *perimeter* and has now adopted the second word's meaning of "limits" or "characteristics."

[4] I should like to thank Patrick Coyne for bringing this article to my attention.

> And musicians note the way *crescendo*, which to them means a gradual increase in volume," has been seized by nonmusicians to mean "climax"— not the reaching but the reached.

And biologists have to live with the colourful semantic diversification of their non-cellular infectious organisms multiplying inside host cells: without *immune software*, the system may be *prone to infection* by *parasitic programs*; in the case of *contamination* by a *viral attack*, the *virus – benign* or *malignant* – will have to be *killed*, and the system *disinfected* or *inoculated*. *Digital hygiene* is recommended.

Gregory James

References

Ackerman, M. S. 1994. Metaphors along the information highway. *Proceedings of the Symposium on Directions and Impacts of Advanced Computing* (DIAC '94), Cambridge MA, April. [Online.] Available at: www.ics.uci.edu/~ackerman/docs/diac94/diac.final. html .

ALADIN. n.d. *The ALADIN* [Academic Language: Assessment and Development of Individual Needs] *curriculum*. California State University, Fresno. [Online.] Available at: caracas.soehd.csufresno.edu/academiclanguage/Academic_Language.htm.

Anthony, L. 1999. The English of computer science – A preliminary report. Paper presented at the JACET Chugoku/Shikoku Annual Regional Meeting, Tottori, June.

________ 2000. Applying genre analysis to technical writing in a foreign language classroom. Paper presented at the International Conference on Research and Practice in Professional Discourse, City University of Hong Kong, November.

Batchelor, R. E. & Offord, M. H. 2000. *Using French: A guide to contemporary usage*. Cambridge: Cambridge University Press.

BBSRC. 2001. Biotechnology and Biological Sciences Research Council *Newsletter*, January.

Beedham, C. & Bloor, M. 1990. English for Computer Science and the formal realization of communicative functions. *Fachsprache 11*, 13–24.

Bloor, M. 1996. The English of computer science: Linguistic issues and learners' problems. In M. Hewings & T. Dudley-Evans (Eds), *Evaluation and course design in EAP* (pp. 180–91). Hemel Hempstead: Prentice Hall Macmillan, in association with the British Council (*Review of ELT 6*, 1).

Bradshaw, J. M. (Ed.). 1997. *Software agents*. Menlo Park CA: AAAI Press & Cambridge MA: MIT Press.

Carroll, J. M. & Mack, R. L. 1985. Metaphor, computing systems and active learning. *International Journal of Man-Machine Studies 22*, 1, 39–57.

Carroll, J. M., Mack, R. L. & Kellogg, W. A. 1988. Interface metaphors and user interface design. In M. Helander (Ed.), *Handbook of human–computer interaction* (pp. 67–85). Amsterdam & Oxford: Elsevier Science.

Cividalli, A. 1996. *Metaphor in computer discourse.* Working paper for the 'Language, Play and Performance in Computer-mediated Communication' project. Jerusalem: He-brew University of Jerusalem.

Condon, C. 1993. The computer won't let me: Cooperation, conflict and the ownership of information. In: S. Easterbrook (Ed.), *CSCW: Cooperation or conflict?* (pp. 171–85). London: Springer-Verlag.

_______ 2000. A semiotic approach to the use of metaphor in human–computer interfaces. PhD, Brurel University.

Corbin, D. R. 1998. *The Corbin handbook of bullet swaging, No. 8*. White City OR: Corbin Publications. [Online.] Available at: www.corbins.com/hb-8.htm.

Coxhead, A. & Nation, P. 1999. The specialised vocabulary of English for academic purposes. *Wellington Occasional Papers in Applied Linguistics 1*, 31–55.

Danet, B. 1997. *Language, play and performance in computer-mediated communication*. Final report submitted to the Israel Science Foundation. Jerusalem: Hebrew University of Jerusalem. [Online.] Available at: atar.mscc.huji.ac.il/~msdanet/report95.htm.

Dudley-Evans, T. & St John, M. J. 1998. *Developments in ESP: A multi-disciplinary approach*. Cambridge: Cambridge University Press.

EFL Tech – France. n.d. *Teaching EST without fear. Tips for "novice" teachers of EFL for engineering, 3*. [Online.] Available at: www.multimania.com/jcviel/tips/tips3.htm.

Erickson, T. 1997. Designing agents as if people mattered. In Bradshaw, pp. 79–96. [Online.] Available at www.pliant.org/ personal/TomErickson/agents.html.

Fang, M. 1999. Sci-tech translation and its research in China. *Meta XLIV*, 1. [Online.] Available at: www.erudit.org/erudit/meta/v44n01/fang/fang.htm.

Gibbon, D. 2000. Introduction: Computerspeak: Jargon or terminology? *Computers and the English language*. *Zwischenprüfung in Englisch*. Fakultät für Linguistik und Literaturwissenschaft, Universität Bielefeld. [Online.] Available at: hcoral.lili.uni-bielefeld.de/Classes/Summer2000/GScomputers/gscomputers/gscomputers.html.

Gil Salom, L., Soler Monreal, C. & Stuart, K. 2000. Lexico-grammatical resources of evaluation in research articles. Paper presented at the 18th Congreso Nacional de AESLA (Asociación Española de Lingüística Aplicada), University of Barcelona, May.

Gotti, M. 1990. Main features of the lexis of computer science. *Studi Italiani di Linguistica Teorica e Applicata XIX*, 1, 49–60.

Graedler, A.-L. 2000. *English-style language: Some words on style and register*. Oslo: Department of British and American Studies, University of Oslo. [Online.] Available at: www.hf.uio.no/iba/nettkurs/translation/grammar/top5style.html.

Grundy, F. 1996. *Women and computers*. Exeter: Intellect Books.

________ 2000. Where is the science in computer science? Paper presented at the 7th International Federation of Information Processing – (IFIP) Working Group 9.1 (Computers and Work), Women, Work and Computerization Conference, Simon Fraser University, June. [Online.] Available at: www.keele.ac.uk/depts/cs/Staff/Homes/Frances/science.htm.

Johnson, C. W. & Hwang, R. 1983. Learnability of technical vocabulary depends upon familiarity, comprehensibility and imagery. *Psychological Reports 53*, 3, 1, 767–70.

Johnson, G. J. 1994. Of metaphor and the difficulty of computer discourse. *Communications of the ACM 37*, 12, 97–102.

Kennedy, G. 1998. *An introduction to corpus linguistics*. London: Addison Wesley Longman.

Keniston, K. 1999. *Language, power and software*. [Online.] Available at: web.mit.edu/kken/Public/papers3.htm.

Kuhn, T. 1993. Metaphor in science. In A. Ortony (Ed.), *Metaphor and thought*. 2nd edi-tion (pp. 533–42). Cambridge: Cambridge University Press.

Laurel, B. 1993. *Computers as theatre*. New York NY: Addison-Wesley.

Lawler, J. M. 1987. Metaphors we compute by. Lecture delivered at the Informational Technology Division, University of Michigan. [Online.] Available at: www-personal.umich.edu/~jlawler/meta4compute.html (27.1.1999).

Li, E. S. L. & Pemberton, R. 1994. An investigation of students' knowledge of academic and subtechnical vocabulary. In L. Flowerdew & A. K. K. Tong (Eds), *Entering text* (pp. 183–96). Hong Kong: Language Centre, Hong Kong University of Science and Technology.

Lundh, D., Olsson, B. & Narayanan, A. (Eds). 1997. Paper presented at the Biocomputing and Emergent Computation Conference, Skövde, September. *Biocomputing and emergent computation*. *Proceedings of BCEC 97*. Singapore & London: World Scientific.
Luzón Marco, M. J. 1997. Phraseological patterns in medical discourse. *The ESPecialist 19*, 1, 41–56.
________ 2000. The construction of novelty in computer science papers. *Revista Alicantina de Estudios Ingleses 13*, 123–40
Madsen, K. H. 1994. A guide to metaphorical design. *Communications of the ACM 37*, 12, 57–62.
Marcus, A. 1998. Metaphor design in user interfaces. *Journal of Computer Documentation 22*, 2, 43–57.
Mey, J. 1997. Metaphors in computing. Signs and realities. *Rask 5/6*, 3–20.
Moore, N. Forthcoming. What is computer English? Paper to be presented at the 'Technology in Language Education' conference, Language Centre, Hong Kong University of Science and Technology, June 2001.
Nass, C. & Steuer, J. 1993. Anthropomorphism, agency, and ethopœia: Computers as social actors. *Human Communication Research 19*, 4, 504–27.
Nass, C., Lombard, M., Henriksen, L. & Steuer, J. 1995. Anthropocentrism and computers. *Behaviour and Information Technology 14*, 4, 229–38.
Netwatcher 17, 10. 1999. October. [Online.] Available at: www.cimicorp.com/Net1710.html.
Palm OS®. 2000. *Take Two: 3.0 and beyond*. [Online.] Available at: oasis.palm.com/dev/kb/papers/1145.cfm.
Peters, C. 2000. *Guaranteeing multilinguality in the information society*. European Research Consortium for Informatics and Mathematics (ERCIM). [Online.] Available at: www.ercim.org/medconf/papers/peters.html.
Ruskin, J. 1856. *Modern painters*. Vol. III: *Of many things*. Boston MA: Estes & Lauriats.
Safire, W. 2001. On language: No 'way'. *New York Times Magazine*, 1.4.2001. [Online.] Available at: www.nytimes.com/001/04/01/magazine/01ONLANGUAGE.html.
St John, M. J. 1996. Business is booming: Business English in the 1990s. *English for Specific Purposes 15*, 3–18.
Skehan, P. 1981. ESP teachers, computers and research. *ELT Documents 112*, 106–25.
Smilowitz, E. D. 1996. *Do metaphors make web browsers easier to use?* [Online.] Available at: www.baddesigns.com/mswebcnf.htm.
Strevens, P. 1973. Technical, technological, and Scientific English. *ELT Journal 27*, 3, 223–34.
Tateson, R. 2000. Biological metaphors in computing. Lecture notes: Bioinformatics. BBSRC Summer School, School of Biochemistry and Molecular Biology, University of Leeds, September. [Online.] Available at: www.bioinf2.leeds.ac.uk/~bbsrc/presentations/BiologicalMetaphorsInComputing/index.htm.
Travers, M. D. 1996. Programming with agents: New metaphors for thinking about computation. PhD, Massachusetts Institute of Technology.
Whitehead, A. N. 1925. *Science and the modern world*. London: Macmillan.
Wray, S. 1997. Biological tools and metaphors for computer science. Lecture given in the Department of Computer Science, Victoria University of Wellington, May. Abstract. [Online.] Available at: www.mcs.vuw.ac.nz/events/seminars/seminars-1997.html#biological-tools-and-metaphors-for-computer-01-may-1997.
Young, J. R. 1997. New ways of organizing data could change nature of computers. Scientists' designs are based on how people tend to remember and classify information. *Chronicle of Higher Education*, 4.4.1997. [Online.] Available at: www.cs.umd.edu/hcil/pad++/press/chronicle.html.