From the proceedings of ECAI'98, the 13th European Conference on Artificial Intelligence, Brighton, UK.

'Just-in-Time' Analogical Reasoning: A Progressive-Deepening Model of Structure-Mapping

TONY VEALE, School of Computer Applications, Dublin City University, Dublin, Ireland.
Email: Tony.Veale@UCD.ie

1. Introduction

Structure-Mapping is a graph-theoretic process which lies at the heart of computational models of analogy, metaphor (see Veale and Keane 1997), case-base reasoning (see Kolodner 1993) and example-based machine translation (see Veale and Way 1997). In essence this problem is a variant of the provably NP-Hard problem of determining the largest common isomorphic pair of sub-graphs shared by two semantic structures (see Garey & Johnson 1979; Veale & Keane 1997), called the *Source* and *Target*, such that a systematic and coherent 1-to-1 mapping of elements from the Source to the Target is established. For example, Figure 1 illustrates an instance of structure-mapping between two story examples that share the same narrative backbone. The suspected intractability of this process has led researchers to tame the exponentiality of structure-mapping by introducing heuristics that operate in polynomial time to produce a near-optimal mapping interpretation, making the process of structure-mapping appealing for both cognitive *and* engineering models (see Oblinger and Forbus 1990; Veale *et al.* 1996a,b, 1997).

However, certain observable but as yet untapped cognitive properties of structure-mapping, as exhibited in the manifest everyday uses of analogy and metaphor, have equal merit in more practical engineering domains. Consider that the brain is the ultimate real-time processing system, as no one cognitive faculty, such as metaphor or analogy, can be allowed to absorb the entirety of the system's resources indefinitely. So while theoretical models of metaphor and analogy are typically given free rein to work on a given mapping problem, the same is not true for a situated cognitive system, which relies on the outputs of various processes to be readily available to deal with the steady stream of external stimuli to which it must constantly react. This is also true of many real-world applications.

In a dynamic context such as everyday conversation, serious time constraints are placed on the operation of those cognitive faculties that employ structure-mapping. While one is free to reconsider and elaborate the full meaning of a metaphor or analogy at a later time, a relatively immediate interpretation of reasonable quality must be produced in time-limited contexts in which the agent is pressured to react and move on to new issues

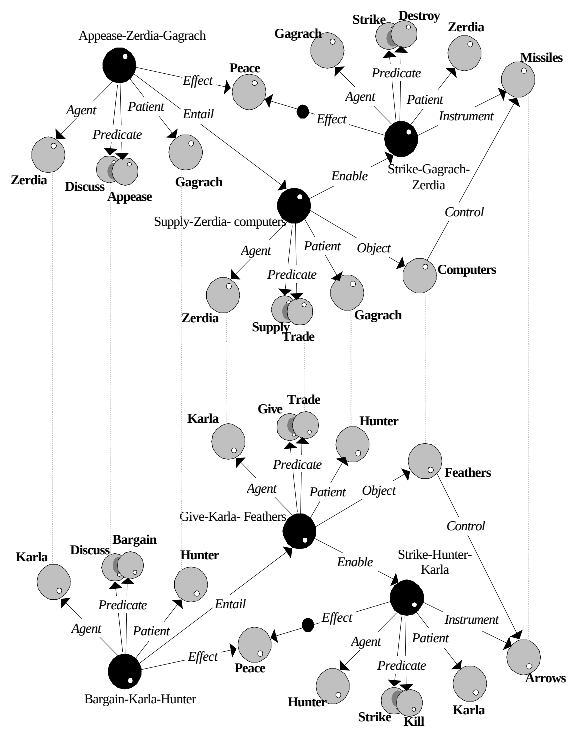

Figure 1: A clear isomorphism is detected between two story domains, those of Karla the Eagle and Zerdia the Country. Grey lines indicate 1-to-1 mappings. This figure is a graphical depiction of an SME benchmark example from Falkenhainer et al. (1989).

See Hoffman and Kemper 1985 for empirical support of this observation. Interestingly, this delicate balance between theoretical competence and time-constrained real-time performance is typified by a classic AI domain – automated games-playing. In chess, for instance, automated players possess the search competence to delve as deep into the game-tree as is needed to defeat their opponents, but are severely constrained by the time requirements of the game to produce a good move in a limited time frame. To maximize their use of time then, automated players employ a strategy known as *iterative or progressive deepening*: a move of reasonable competence is produced safely within the given time-limit, and if enough time remains, the computer attempts to better this move by searching again, from the start, to a deeper ply (see Korf 1985). The system continues to use remaining time in this fashion, until when time eventually runs out it simply uses the last move it was able to properly consider.

In this paper we present a progressive-deepening model of structure-mapping that meets the real-time demands of both cognitive theories *and* practical systems. The model we describe is an elaboration of Sapper, a spreading-activation model of metaphor and analogy that has been demonstrated to possess the mapping competence of Falkenhainer *et al.*'s SME – *the Structure-Mapping Engine*, and Holyoak and Thagard's ACME – the *Analogical Constraint Mapping Engine*, while exhibiting superior tractability and run-time performance (empirical results are presented in Veale *et al.* 1996a,b; 1997). SME is a symbolic matcher for expressions in FOPC, whose interlocking nested predications form a *forest-of-trees* representation, while ACME is a connectionist model that views analogy as a process of constraint satisfaction. However, both models view complex nested predications as trees, wherein predicates act as internal nodes and objects act as leaves. As we shall show, this structural preference makes these models resistant to the advantages of progression-based search.

2. Constraints and Reactivity in Structure-Mapping

The situated and dynamical nature of analogy and metaphor imposes a variety of real-time demands on the structure-mapping process. In particular, we distinguish three broad forms of contextual sensitivity to which the process may be required to react:

Temporal constraints: The cognitive agent may impose a time limit on the structure-mapping process. In this case, the process must return the optimal, or near-optimal, mapping available in the given time. The agent should thus act in a *bounded-optimal* fashion (see Etzioni 1989). However, given the proven NP-Hardness of structure-mapping, and the need to use sub-optimal heuristics, the best one can expect is *bounded-near-optimal* behaviour from such agents.

Quality constraints: The cognitive agent may require an interpretation that satisfies a particular level of mapping richness. A deep interpretation will thus map more elements from the Source to the Target than a shallow one, and most likely require a greater amount of time to do so.

Pragmatic constraints: The cognitive agent may wish to produce an interpretation in which a certain element of the Target or Source domains is mapped in a particular way. For instance, when mapping the domain of *Japanese banking* onto that of the *Tamagotchi virtual pet* (as done on the cover of *'The Economist'* Nov. 29, 1997), the agent may impose a pragmatic directive that Yamaichi, the most notorious of Japan's failed brokerages, receive a mapping in the metaphor.

These considerations allow us to describe three forms of a reactive structure-matcher:

Time-Driven: The system assumes infinite time resources, and calculates a mapping solution accordingly. However, it may be interrupted at any moment, at which point it must return a reasonable (i.e., bounded near-optimal) solution that reflects the temporal resources it has consumed.

Quality-Driven: The system again assumes infinite time resources, and progresses toward a solution that meets given quality requirements, typically measured in terms of the number of elements mapped by the interpretation. Such a system will terminate when the quality level is met, or when interrupted by the cognitive agent.

Goal-Driven: In the much the same fashion as a quality-driven system, the matcher progresses until certain goal-related elements of the Target are bound (perhaps to pre-specified elements of the Source) or until an interruption occurs.

In many cases a structure-matcher may be expected to observe all three constraints on its progression, i.e., to generate/interpret a goal-driven metaphor of a given quality in a given time–frame. As we shall show, the iterative-deepening structure-matcher described in the following sections is open to all three forms of constrained progression.

3. Sapper: A Memory-Situated Account of Metaphor

The Sapper model of Veale *et al.* (1996a,b) views semantic memory as a localist graph in which nodes represent distinct concepts, and arcs between those nodes represent semantic/conceptual relations between concepts. A conceptual domain C is thus a connected sub-graph of semantic memory rooted at node C, i.e., that collection of nodes and arcs reachable from node C. Memory management under Sapper is pro-active toward structure mapping, that is, it employs rules of structural similarity to determine whether any two given nodes may at some future time be placed in systematic correspondence in a metaphoric context. If so, Sapper notes this fact by laying down a *bridge relation* between these nodes, which can be exploited in some future structure-mapping session. The two rules which Sapper employs to lay down these bridges are termed the *Triangulation* and *Squaring* rules:

Triangulation: *If memory already contains two linkages* L_{ij} *and* L_{kj} *of semantic type L forming two sides of a triangle between the concept nodes* C_k, C_i *and* C_j, *then complete the triangle and* *augment memory with a new conceptual bridge linkage B_{ik}.*

Squaring: *If B_{jk} is a bridge, and if there already exists the linkages L_{ij} and L_{lk} of the semantic type L, forming three sides of a square between the concept nodes C_i, C_j, C_k and C_l, then complete the square and augment memory with a new bridge linkage B_{il}.*

At some future time, if Sapper wishes to determine a structural mapping between a target domain rooted in the concept node T (for Target) and one rooted in the node S (for Source), it applies the algorithm of Fig. 2.

```
Spread Activation from nodes (T)arget and (S)ource in memory to a horizon H
    When a wave of activation from T meets a wave from S at a bridge T':S'
            linking the tenor domain concept T' to the vehicle domain concept S' Then:
        Find a path of semantic relations R that links both T' to T and S' to S
        If R is found, then the bridge T':S' is balanced relative to T:S, so Do:
           Generate a partial interpretation (pmap) π of the metaphor T:S as follows:
                 For every tenor concept t between T' and T as linked by R Do:
                   Put t in alignment with the equivalent concept s between S' and S
                   π ←  π ∪ {<t : s>}
            Φ ← Φ ∪ {π}
    Once the set Φ of all pmaps within the horizon H have been found, Do
        Evaluate the richness of each pmap π ∈ Φ
        Sort the collection Φ of pmaps in descending order of richness.
        Pick the first (richest) interpretation Γ ∈ Φ as a seed for overall interpretation.
        Visit every other pmap π ∈ (Φ - Γ) in descending order of richness
            If it is coherent to merge π with Γ (i.e., without violating 1-to-1ness) then
                 Γ ← Γ ∪ π
            Otherwise discard π
        When Φ is exhausted, Γ will contain the overall Sapper interpretation of T:S
```

Figure 2: The Sapper Algorithm, as based on the exploitation of cross-domain bridge-points in semantic memory.

The second phase then coalesces this collection of pmaps into a coherent global whole; it does this using a *best-first seeding* algorithm which starts with the structurally richest pmap as a seed, and attempts to fold every other pmap into this seed, if it is coherent to do so, in descending order of richness of these pmaps (this approach is directly equivalent to the greedy merge used by Oblinger & Forbus 1990). Such a best-first approach is sub-optimal, as even a highly elaborate seed can preclude a union of smaller pmaps from enriching the overall solution. Nevertheless, empirical results reported in Oblinger & Forbus (1990) and Veale & Keane (1997) indicate the approach to be near-optimal (e.g., Veale & Keane report that Sapper quality levels are, on average, within 5% of optimal).

4. Requirements of a Progressive Deepening Model

Two algorithmic requirements characterize the progressive-deepening search strategy: (i) the search begins anew each time from the same, fixed starting point; and (ii) the search is easily constrained by a movable horizon. The first constraint ensures that successive progressions produce solutions which are at least as good as those previously found, while the second allows the search to asymptotically approach a bounded-near-optimal solution given the stated time limitations. But not all semantic representations used in structure-matching, such as those used by ACME and SME, are amenable to these constraints.

Firstly, when determining the largest sub-graph isomorphism between two structural descriptions, it is important that the Source and Target are represented as rooted graphs, and that the roots of these graphs comprise a mapping in the final solution. For instance, when analogically mapping the domain of *Surgeon* to that of *General*, the nodes *Surgeon* and *General* will comprise a mapping of the overall interpretation (which may also include mappings such as *Scalpel* *Snub-Fighter*, *Surgery* *Slaughter* and *Cancer* *Enemy-Army*). The mapping *Surgeon* *General* thus acts as a rooted started point from which the structure-matcher can commence, and return to as the algorithm iteratively seeks a richer (i.e., deeper) interpretation. It will always be possible to generate such a rooted initial mapping for an analogy, even in cases where the Source and Target are each represented as collections of individual narrative events, as is the case with the *Karla* and *Zerdia* analogy of Figure 1. In such cases the system need simply create a new pair of graph nodes, *Zerdia-*

Story and *Karla-Story*, to act as story roots that connect to the main events of their respective narratives.

In Sapper, a given knowledge-domain is characterized as a semantic network that extends from a given conceptual node; thus the domain of *Surgeon* is that connected sub-graph of memory that is reachable from the node *Surgeon.* In this respect all domains presented to the structure-matcher are rooted graphs, establishing a fixed starting point from which the algorithm can progressively operate. In contrast however, algorithms such as SME and ACME each view a knowledge-domain as comprising a collection of related predications, in effect comprising an unrooted *forest of trees* representation. Without a root from which to start, a progressive-deepening matcher has no fixed point from which to conduct (and restart) its search of the mapping space.

Secondly, the use of a moving horizon means that the representations being matched must not be subject to *horizon effects*. That is, the collection of semantic propositions cordoned within the current horizon setting must comprise a sensible meaning structure with a sensible interpretation, if the analogical mapping derived from this structure is itself to be meaningful. As illustrated in Figure 1, Sapper employs a semantic network perspective on domain knowledge, where both events and entities are modeled as graph nodes, while relations between them are modeled as relational arcs. Drawing a horizon a given distance from the roots of the graph (*Zerdia* and *Karla* in this instance) will thus always cordon off a well-formed semantic sub-network. However, because ACME and SME match FOPC predicational structures (equivalent in expressive power to the semantic-network approach, but not always as computationally felicitous), such a cordon will frequently separate predicates from their arguments. While this is also true of Sapper, it is problematic for SME and ACME as these systems grant differing ontological status to objects versus predicates. Indeed, because SME always maps predicates identically, an interpretation of any SME analogy will always be expressed in terms of the object/entity correspondences that it entails. It thus makes no sense for SME (or ACME, which is essentially a connectionist equivalent of SME in terms of how it views structure) to map nested predicational structures (essentially trees) that have been separated by an artificial horizon from their leaf-level arguments, as such mappings would contribute nothing to the richness of the final interpretation. SME and ACME are thus *all-or-nothing* approaches that are not amenable to the idea of a moving search horizon.

In contrast, as shown in the algorithm of Figure 2, Sapper is an inherently horizon-based algorithm since it is built upon the mechanism of spreading activation. This horizon is depicted as a parametric constant in Figure 2, but can easily be made to move deeper in memory from the starting point of the search to implement a progressive-deepening structure-matcher.

Function Progressive-Sapper (T:S, Goals, Quality)
Let *H 1;* ***Let*** *BestSoFar*
Try *{*
    Start: ***Let*** *BestSoFar Sapper(T:S, H)*
    If *BestSoFar satisfies Goals and Quality Then*
    Return *BestSoFar*
    Else
    H H + 1
    Goto *Start*
} ***CatchException*** *(OutOfTime, OutOfSpace) {*
    Return *BestSoFar*
}

Figure 3: A Progressive-Deepening Framework for Sapper

A progressive version of Sapper can thus be described as in Figure 3. Each increment to the horizon H in the algorithm of Figure 3 causes Sapper to search one relational link further from its starting point T:S. Sapper thus progresses through the space of structure-matches in a breadth-first fashion, which ensures an *asymptotically-bounded* near-optimal behaviour (see Russell and Subramanian 1993). Progressive-Sapper's performance is asymptotically-bounded because of the inherent overheads of iterative-deepening search: these overheads, incurred when Sapper returns to its rooted starting point to start each new progression anew, are small, since the search-space is exponential, but it does mean that a non-iterative structure-matcher could achieve the same results with slightly less temporal resources. Nonetheless, such a non-iterative solution would not exhibit Progressive-Sapper's adaptability to analogue complexity.

5. Empirical Results

Progressive deepening confers a number of important advantages to the structure-matching process. Firstly, it performs well in time-limited situations when dealing with large analogue structures. The analogues one typically employs to validate a theory of metaphor or analogy have an artificial, toy-like quality that rarely stretches the mechanism under test, but one imagines that the nature of real cognitive structures to be significantly more complex. If one considers that a given analogue structure, for *Surgeon* say, will comprise all those predications of relevance to *Surgeon* (from *'Scalpel cuts Flesh'* to *'Medical-School is Expensive'*), the memory and time requirements of a non-progressive matcher will be potentially huge. However, a progressive-matcher will gradually expand its coverage to include only those elements of the domain that it is capable of handling, given its own time and space restrictions. Besides its inherent engineering practicability, this would also seem to reflect a more accurate picture of human cognition.

Secondly, progressive-deepening also responds well to those situations where the structure-matcher *over-estimates* the complexity of the domains it must map. Consider again the vanilla Sapper algorithm of Figure 2; by setting the horizon of spreading activation to a default value of 5, all analogies requiring a recursive depth of 5 or less can be handled. However, those requiring a greater depth of 6, say, will be under-processed while those requiring a more shallow depth of 2, say, will be *over-processed*, returning the correct result but at greater computational cost than is necessary. Figure 4 presents a complexity breakdown of the metaphors on which Sapper is typically employed. While the most interesting of these comparisons require a deep level of search to generate a complete mapping (e.g., H = 5), many others require a significantly lower-level of inquiry on average (e.g., H = 1.76).

Metaphor/Analogy	**Avg. Horizon**	**Deepest H.**
Star Wars: Arthur	*1.83*	*4*
Kennedy: Arthur	*1.74*	*4*
The Natural: Arthur	*1.89*	*5*
Star Wars: Dambusters	*1.54*	*4*
Eco’s O/S metaphor	*1.96*	*5*
Sports Car: Jaguar	*1.89*	*5*
Surgeon: General	*1.33*	*3*
General: Composer	*1.47*	*3*
Profession Metaphors (avg. 100 examples)	*1.15*	*1.76*

Figure 4: Average and maximum horizon settings required to process some Sapper mappings. ‘Mean Mapping Depth’ indicates the average horizon setting (or level of recursion) needed to discover the average mapping in a given metaphor.

Progressive-Sapper does not over-process the simple cases, yet is able to dynamically extend its coverage to meet the demands of the more complex (and rarer) examples. In fact, for the our corpus of 100 profession metaphors (which includes *Surgeon : General* and *General : Composer*), Progressive-Sapper searches to an average horizon depth of 1.74 when asked to generate an interpretation of the same quality as that produced by vanilla Sapper for the same metaphors.

Progressive Sapper has also been tested on the classic analogy benchmarks upon which SME and ACME’s competence has been traditionally established. For these experiments an automatic transformation is applied to the FOPC representations of these problems, which include the *Karla : Zerdia* analogy of Figure 1 (shown there as a Sapper semantic network) to produce equivalent Sapper representations from which the same interpretations are derived. The relative search complexity of these examples is tabulated in Figure 5.

Metaphor/Analogy	**Avg. Horizon**	**Deepest H.**
Karla: Zerdia	*1.5*	*4*
Fortress: Tumor	*1.38*	*3*
Cannibals: Farmer	*1.28*	*2*
Solar System: Atom	*1.22*	*2*
Rebels: Contras	*1.29*	*2*
Heat-flow: Water	*1.27*	*2*
Socrates: Midwife	*1.31*	*2*

Figure 5: Average and maximum horizons required to process SME/ACME benchmarks.

5.1. Improving Near-Optimal Results

Clearly, Progressive-Sapper possesses the same mapping competence as vanilla Sapper, inasmuch as its roving horizon is capable of find all those interpretations that the basic Sapper algorithm can find, usually within a greatly reduced search cordon. But ironically, Progressive-Sapper can sometimes provide better results than vanilla Sapper within this reduced search-space, as it is possible, albeit rarely, for a seed pmap rich in mappings to be less effective than a more mapping-impoverished pmap. For though mapping-rich pmaps provide a good starting point for a greedy-seeding algorithm such as that used by Sapper and by Greedy-SME (see Oblinger & Forbus 1990), large seeds can also preclude the inclusion of numerous smaller, but conflicting, pmaps into the final solution. But unlike Greedy-SME, Progressive-Sapper from a minimal seed (of size H = 1) and progresses from there to more complex solutions, thus identifying the most effective search-depth at which to build its interpretation. Progressive-Sapper might thus find a richer interpretation with a horizon setting of H = 2 than vanilla Sapper might find with a setting of H = 5 (this is indeed the case with the *Author : Architect* metaphor in our profession corpus). Indeed, for this reason, the average search horizon required by Progressive-Sapper to match Sapper’s competence on the profession corpus is 1.74, slightly less than the average search-depth of 1.76 required by Sapper itself (as reported in Figure 4).

5.2. Elaborative Progression

While progressive structure-matching makes near-optimal use of limited resources, it can also be applied in situations which are not strictly resource-constrained. For instance, even if time is not an issue, cognitive parsimony suggests that a structure-matcher should not spend more time than is necessary in producing a near-optimal solution.

In such unconstrained scenarios the structure-matcher needs a criterion other than time to halt the progression of the search horizon, or to indicate the futility of additional search. Conveniently, the squaring rule of section 3 provides such a terminating heuristic. After searching to a horizon depth H, another iterative variant of Sapper called *Elaborative-Sapper* examines each individual mapping in the interpretation BestSoFar to see, if by simple application of the squaring rule, that mapping can be connected to another bridge in conceptual memory not already considered by the search. If so, Elaborative-Sapper considers it worthwhile to extend the horizon and continue the search; if not, it terminates and returns the current interpretation. By further analogy with chess-playing, a BestSoFar solution that cannot be extended in this way by the squaring rule is said to be a *quiescent* solution, i.e., one that will not change if one more iteration of progressive deepening is applied. Applying Elaborative-Sapper to the corpus of 100 profession metaphors requires an average search horizon of 2.01 to reach a quiescent solution, slightly more than that required by Progressive-Sapper (whose experiment was guided be quality thresholds).

However, a quiescent solution might nevertheless be improved if two or more additional iterations of deepening are applied, since the squaring rule can only see a single relation beyond the current horizon. Since use of the squaring rule as a terminating criterion is a heuristic that may sometimes fail, Elaborative-Sapper does not possess the asymptotically-bounded near-optimality of Progressive-Sapper. Nonetheless, measuring interpretation quality in terms of the number of individual cross-domain mappings produced for each metaphor, Elaborative-Sapper generates interpretations that possess, on average, 99% of the quality of the equivalent Progressive-Sapper interpretations.

6. Summary and Conclusions

This paper has presented three variants of a structure-matching algorithm: *Sapper*, *Progressive-Sapper* and *Elaborative-Sapper*. Each algorithm operates upon a semantic-network representation of analogue structure, but is also applicable to the benchmark analogies of the SME and ACME models with the aid of a simple and automatic representational transformation (given that FOPC and semantic-networks are representationally equivalent). The use of progressive-deepening makes Sapper capable of processing large analogies in an asymptotically-bounded near-optimal fashion within the time and space constraints of the system in which it is used. This not only makes Sapper a more cognitively attractive theory of metaphor and analogy, but also a more practicable approach to the general problem of structure-matching as it is used in real applications.

Source code for Sapper system (implemented in Sicstus Prolog), in addition to the corpus of test metaphors on which it has been tested, is available from the author's web-site at: *http://www.compapp.dcu.ie/~tonyv/*.

7. References

Etzioni, O. (1989). >Tractable decision-analytic control. In *the Proceedings of the 1st International Conference on Representation and Control,* pp 114-125, Toronto.

Falkenhainer, B, K. D. Forbus, & D. Gentner. (1989). The >Structure-Mapping Engine.>*Artificial Intelligence*, 41, pp 1-63.

Garey, M. R. & D. S. Johnson. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York.

Hoffman, R. & S. Kemper. (1987). What could reaction-time studies be telling us about metaphor comprehension?, *Metaphor and Symbolic Activity* **2**(3), 149-186.

Holyoak, K. J. and P. Thagard. (1989). Analogical Mapping by Constraint Satisfaction, *Cognitive Science* **13**, pp 295-355.

Kolodner, J. (1993). *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufman.

Oblinger, D. and K. D. Forbus. (1990). Making SME Pragmatic and Greedy, in *the Proceedings of the Twelfth Annual Meeting of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

Korf, R. E. (1985) Iterative deepening: An optimal admissible tree search. *Artificial Intelligence* 27 pp 97-109.

Russell, S. J. & D. Subramanian. (1993). Provably bounded optimal agents, *in the Proceedings of IJCAI'93, the 13th International Joint Conference on Artificial Intelligence*, Chambery, France. San Mateo, CA: Morgan Kaufmann.

Veale, T., B. Smyth, D. O'Donoghue & M. T. Keane. (1996a). Representational Myopia in

Cognitive Mapping, *in the proceedings of the 1996 AAAI workshop on Source of the Power in Cognitive Theories*, Portland, Oregon, August 1996.

Veale, T. D. O'Donoghue & M. T. Keane. (1996b). Computability as a Limiting Cognitive Constraint: Complexity Concerns in Metaphor Comprehension about which Cognitive Linguists Should be Aware, *Cognitive Linguistics: Cultural, Psychological and Typological Issues (forthcoming).*

Veale, T. & M. T. Keane. (1997). The Competence of Sub-Optimal Structure Mapping on 'Hard' Analogies, in *the Proceedings of IJCAI'97, the International Joint Conference on Artificial Intelligence,* Nagoya, Japan, August 1997.

Veale, T. & A. Way. (1997). Gaijin: A Template-Driven Bootstrapping Approach to Example-Based Machine Translation, in *the Proceedings of NeMNLP'97, New Methods in Natural Language Processing,* Sofia, Bulgaria, September 1997.