THEORIA ET HISTORIA SCIENTIARUM, VOL. VI, N° 1
Ed. Nicolas Copernicus University 2002

Tony Veale

How to Blend Concepts and Influence People: Computational Models of Conceptual Integration

1. Introduction

Recent work into the cognitive foundations of conceptual integration and blended mental spaces provides a unifying umbrella framework for a range of cognitive ‘siblings’ that have traditionally been studied with relative independence, such as metaphor, analogy, concept combination, grammaticalization, counterfactual thought, and abstract problem solving. The ‘many-space’ or ‘conceptual integration networks’ theory of Fauconnier and Turner (1994; 1998) is an elaboration of the two-space model of metaphor that has been the corner-stone of the metaphor field since Aristotle (see Hutton, 1982), and which has underpinned a string of conceptual theories from Nietzsche (see Culler, 1980) through Richards (1936), Black (1962), Koestler (1964) to Lakoff and Johnson (1980). These theories posit a metaphor (or by extension, its cognitive siblings) to concern the interaction of two conceptual spaces; the space which is described by the metaphor has variously been termed the *target,* the *tenor* or the *topic,* while the space that provides the description has been called the *source,* the *vehicle* or the *base.*

Somewhat problematically, the knowledge required to construct and interpret a metaphor is not always readily shoehomed into one of these two spaces. For instance, one would like the expressiveness to state that certain low-level knowledge is common to each space and acts as a domain mediator between both. Often one needs to state that other knowledge, or perhaps other metaphors, must be recruited from outside both spaces to act as the necessary glue to relate them. Likewise, it is frequently the case that the product of a metaphor is a new conceptual space that owes its structural origins to the interactions of the source and target spaces, but which has a conceptual existence of its own that allows it to grow and accrete new associations independently of those origins. The conceptual integration framework of Fauconnier and Turner provides one with the theoretical apparatus to make each of these distinctions, by augmenting the traditional input spaces with two additional spaces: a *generic space* that captures the common background knowledge that unites the inputs, and an output *blend space* that contains the conceptual product of the integration.

In this paper we explore the computational requirements of the theory of conceptual integration, and propose an algorithmic model that meets these requirements. Broadly speaking, we see three reasons for seeking a computational account of a powerful theory like conceptual integration. Firstly, consider that theoretical utility is inversely proportional to expressive power, and that overly powerful theories have little cognitive status, since scientifically, one should seek the least powerful theory that accounts for the most facts. It is important then that conceptual integration is shown not to be overly powerful. Similarly, a functional view of mind suggests that such a theory should be computationally tractable and not make infeasible processing demands. So just as cognitive theories should be falsifiable via empirical testing, such theories should also be shown to be tractable via computational modelling. This paper demonstrates the tractability of conceptual integration networks by showing how a tractable computational model, called Sapper, can accommodate the processes underlying conceptual integration.

Secondly, as we have noted, conceptual integration expands the descriptive options open to the theorist of metaphor. But increased options regarding the contents and interplay of a greater number of mental spaces also make for extra degrees of freedom. While conceptual integration theory becomes more compelling by allowing sophisticated analyses of a growing body of cases, these analyses should themselves be compelling and unambiguous, and not have the appearance of cognitive ‘just-so’ stories. To this end, Fauconnier and Turner have introduced structural constraints that pin-down the optimality conditions under which integration can occur in a network, sufficiently reducing the theory’s degrees of freedom and thus, the arbitrariness of its analyses. An algorithmic perspective on integration can provide yet another form of constraint, explaining why it computationally necessary to organize and populate the spaces of an integration network in a given way.

Thirdly and finally, a computational model serves as a useful analytic tool in the repertoire of the cognitive linguist, in effect providing a cognitive simulator in which integrations or blends can actually be ‘run’. Simulation of this kind allows a linguist to generate and test a host of different structural hypotheses, such as *what happens if we add this structure to this space?* and *how much structure needs to be added here, or removed there, to force such and such an interpretation.* Computational models thus make explicit the knowledge requirements of any given integration, and allow various empirical claims to be reproduced by others.

With the goal of placing conceptual integration theory on a computational footing, this paper observes the following structure. In section 2 we provide a brief recap on the nature of conceptual integration, or blending, as advocated by Fauconnier and Turner (1994; 1998). In section 3 we introduce the basic computational elements from which our algorithmic account of integration will be constructed, and in section 4 we discuss how these elements are present in various computational models of metaphor and analogy. Section 5 then illustrates how one of these models, called Sapper, can actually be seen as instantiating the many-space perspective on conceptual integration advanced by Fauconnier and Turner. We argue that to view Sapper as a model of conceptual integration is more than convenient rationalisation on our part, and describe how the computational perspective offered by Sapper can actually contribute to our understanding and use of conceptual integration theory in general.

2. Conceptual Integration and Blending: An Overview

In the terms of Fauconnier and Turner, the interacting content spaces that go into producing a conceptual blend are organized according to Fig. 1. Shown in Fig. 1 are the traditional spaces normally associated with metaphoric mapping- the “Source” and “Target” domains. Within the Fauconnier and Turner model, these spaces combine via some structural mapping (often a metaphoric one) to produce another, independent blended space that provides the focal point for the resultant integration. However, perhaps the most significant contribution of the Fauconnier and Turner model, over and above the now standard Lakoff and Johnson (1980) two-space perspective on metaphor, is the use of an additional distinct co-ordinating space, known as generic space. This space contains the low-level conceptual structures that serve to mediate between the contents of the input spaces, thus enabling them to be structurally reconciled. We give this notion of structural reconciliation a computational form in a later section, but for now it is sufficient to say that it involves mapping the conceptual structure of one input space onto another so as to obtain a coherent alignment of elements from each.

For instance, we can reconcile the domains of *Scientist* and *Priest* by seeing laboratories as churches, lab-benches as altars and scientific-method as religious dogma. Often, this reconciliation relies on a knowledge of the conventions governing the use of concepts in particular domains. For instance, in the Fauconnier and Turner (1998) example of *Death as the Grim Reaper,* generic space provides low-level structures that are relevant to the process of personification, and which serve to mediate between the input spaces of a metaphysical concept, *Death,* and a physical concept, *Farmer.*

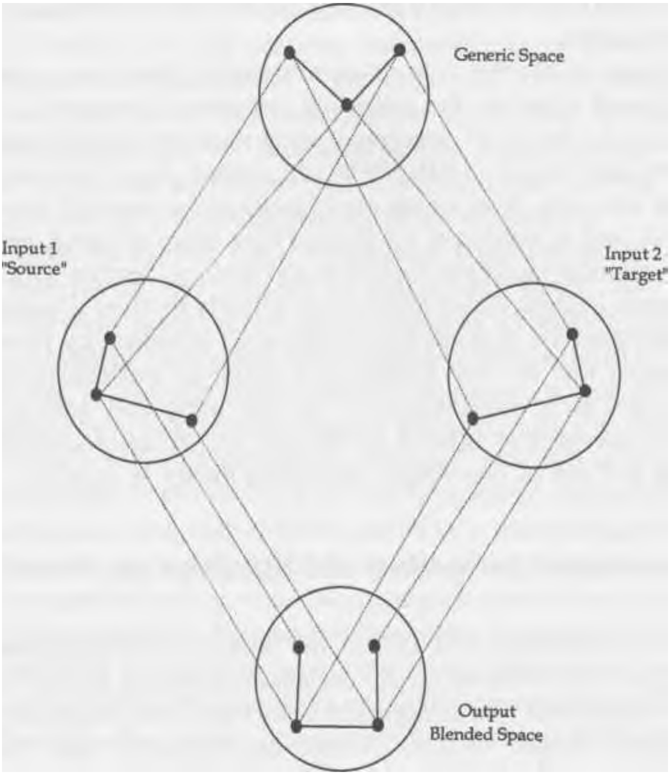


Figure 1: A schematic view of the Blended Space Model of Fauconnier and Turner. Under the structural constraint of Generic Space, a structure in Space 1 is blended with a structure from space 2 to create a more elaborate structure in the Output Space. Solid dots represent entities in each domain, solid lines represent relations between those entities, while dashed lines represent mappings between entities of different spaces.

The result of this mediation is the creation of a new *blend space* into which elements of the inputs are coherently projected. Because the notion of a blend- space provides a convenient means of seperating the product of conceptual integration from the spaces that are actually integrated, integration theory yields a compelling account of why many metaphors/blends often give rise to emergent properties that are, in a sense, pathological from the perspective of the contributing input spaces. For instance, consider the now conventional blend *Black Hole* (a term originally coined by the physicist John Archibald Wheeler), which fuses the abstract notion of a bizarre celestial phenomenon (as predicted by Einstein's Theory of General Relativity) with the notion of a common-place hole or rift. The mediating image-schema in generic space for this blend is most plausibly the notion of SPACE- TIME AS A FABRIC, an oft-used metaphor in modern physics. Incorporated into the blend is an additional source space - that of *Blackness* - which contributes an aura of mystery, invisibility and the unknown to the finished concept. But from its inception, this blend ingredient has idiosyncratically conflicted with the ‘hole’ source space inasmuch as it is believed that anything that enters a black hole cannot exit; this conflicts with our folk understanding of the common-place variety of hole, such as potholes, manholes, and so on.

Indeed, advances in modem physics have seen scientists further distance their models of ‘black holes’ from the idealized cognitive models that underlie both ‘blackness’ and ‘holes’. For instance, black holes are no longer considered truly black, inasmuch as they possess an entropy that radiates detectable quantities of gamma rays; more counter to standard intuitions is the related idea that blackholes are *self filling,* since as radiation is emitted, black-holes lose their energy and shrink, eventually disappearing into themselves (see Hawking, 1975). However, because the blended concept exists in a derived yet independent space of its own, accessible via the lexical item ‘Blackhole’, such alterations do not corrupt our understanding of the original source spaces labeled ‘Black’ and ‘Hole’.

Fauconnier and Turner outline five optimality constraints that delimit what it means for a conceptual integration network to be conceptually well-formed. These constraints are not orthogonal, so one should not expect any given integration to observe them perfectly. Briefly, these constraints are (i): the *integration* constraint, which states that blended elements (such as *Church* and *Laboratory)* should be readily manipulated as single conceptual units; (ii) the *web* constraint, which ensures that the integration constraint does not sever the links between newly blended elements and their original inputs; (iii) the *unpacking* constraint, which states that anyone who comprehends the blended result of an integration should be able to reconstruct the network of spaces that gave rise to it; (iv) the *topology* constraint, which safeguards the semantic validity of an integration by ensuring that those corresponding elements that are blended together (such as Church and *Laboratory)* relate to the other elements of their spaces in a similar fashion (e.g., *Church* relates to *Altar* in the same way *Laboratory* relates to *Lab-Bench)-,* and (v) the *good reason* constraint, which ensures any concepts in the blend can be granted significance or relevance by virtue of its connection to other elements of the blend.

3. Computational Elements

Though not an explicitly computational framework, Fauconnier and Turner’s theory of conceptual integration networks resonates with a number of fundamental computational ideas that find considerable application in the field of Artificial Intelligence. Foremost amongst these is the notion of a *semantic network,* a graphtheoretic structure in which conceptual knowledge can be represented in a structured fashion. A semantic network is a data structure that in turn gives rise to the process of *spreading activation*, an idea that has both computational and psychological origins. Taken together, these ideas provide the algorithmic means to place conceptual integration on an explicitly computational footing.

3.1. Semantic Networks

A semantic network, as defined in Quillian (1968), is a graph structure in which nodes (or vertices) represent concepts, while the arcs between these nodes represent relations among concepts. From this perspective, concepts have no meaning in isolation, and only exhibit meaning when viewed relative to the other concepts to which they are connected by relational arcs. In semantic networks then, structure is everything. Taken alone, the node *Scientist* is merely a syntactic token that happens to possess a convenient English label, yet from a computer's perspective, even this label is an arbitrary alphanumeric symbol. But taken collectively, the nodes *Scientist, Laboratory, Experiment, Method, Research, Funding* and so on exhibit a complex inter-relational structure that can be seen as meaningful, inasmuch as it supports inferences that allow us to conclude additional facts about the *Scientist* domain, as well as supporting semantic regularities that allow us to express these facts in a language such as English (see Cunningham and Veale, 1991; Veale and Keane, 1992).

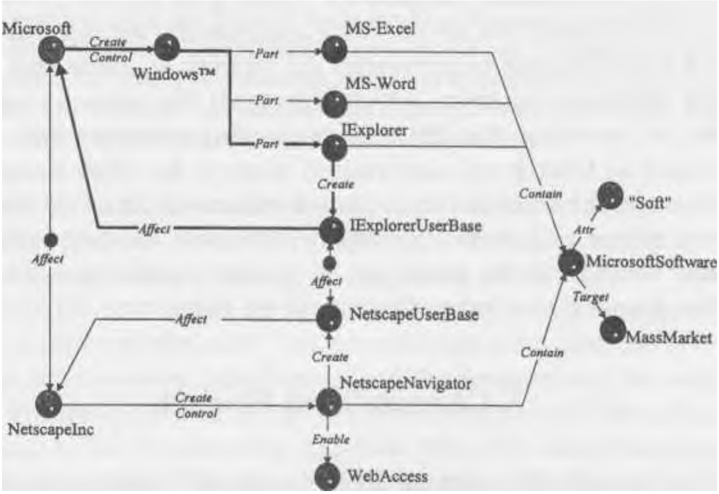
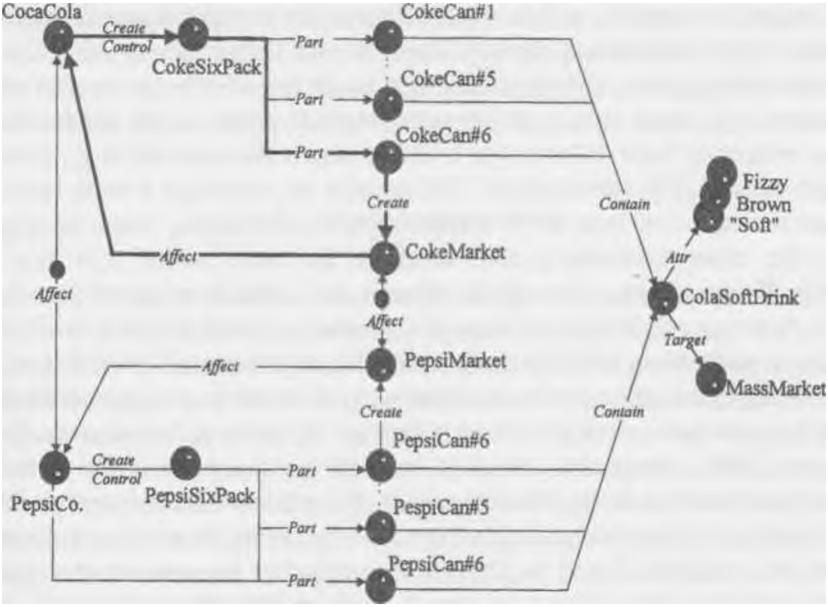


Figure 2: The Market Dynamics of Microsoft and Netscapelnc. Semantic Relations marked with a • indicate pejorative (as opposed to strictly logical) negation; thus, Microsoft - • affect —» Netscapelnc means that Microsoft negatively affects Netscapelnc.

Long-term memory can be seen as a complex graph structure then in which ideas, events and experiences are all represented in this arcs and nodes fashion (we shall refer to the network representation of long-term memory as 'semantic memory'). A defining aspect of semantic networks is that the representation of these ideas will interweave by virtue of sharing common nodes and arcs. For example, the concept node *Oppenheimer* will partake in relations that link it to the domains of *Science, War, Politics* and *Ethics.* A conceptual domain in a semantic network is a structured collection of nodes and arcs that can be reached by recursively traversing all arcs that originate at a given node of origin.



ngure 3: The Mirror Domain to that of Figure I, Illustrating Similar Market Dynamics at Work in the Rivalry between CocaCola and PepsiCo.

For instance, Figure 2 illustrates a sub-section or domain of semantic memory reachable from the concept node *Microsoft,* while Figure 3 illustrates the structurally similar domain of *CocaCola.* Note how the connectivity of the concept M*icrosoft* means that concepts relating to Netscapelnc are also included in this domain, while the connectivity of the *CocaCola* domain causes the concept *PepsiCo* and its associates to likewise be included there.

The domain of *Microsoft* thus comprises all those concept nodes and relations that can be reached by starting at the node *Microsoft* and its immediate neighbours, visiting the neighbours of each new node in turn until no new nodes can be reached.

3.2. Spreading Activation

This recursive node-visiting process is traditionally called *spreading activation* in the cognitive/psychological literature (e.g., see Quillian, 1968; Collins and Loftus, 1975), and marker passing in the computational literature (see Charniak, 1983; Hendler, 1989). From the former perspective, not only are neighbouring nodes visited in a wave-like progression from the starting node, but an activation signal is propagated as well, from node to node. This activation signal has an initial value (or ‘zorch’, as it is often called in the computational literature; see Hendler, 1989) which diminishes and attenuates the further the wave is propagated from its starting point. This attenuation might be specific to the arc carrying the activation (e.g., some arcs in the network might be more or less *conductive* than others, reflecting their salience) or constant across the network (e.g., traversing any arc causes 10% attentuation). The amount of activation a node receives is thus an indication of how far it is from a particular starting point. In cognitive terms, the more activation a node receives, the more central it is to a given domain. If one views a conceptual domain as a *radial category* (see Lakoff, 1987), highly representative concepts (nodes) of that domain will receive significant activation, while less representative members will receive less.

Spreading activation can be simultaneously initiated from a number of starting points in a semantic network (these points are typically called *matriarches'*, see Quillian, 1968), where the activation level of a given node is the sum of the activation it receives from different waves. For instance, the concept nodes *Soft* and *MassMarket* are each reachable from both the nodes *Microsoft* and *CocaCola*, as shown in Figures 2 and 3. These nodes can thus be isolated as a potential common ground for viewing *Microsoft* as *CocaCola.* The process of marker passing is similar to that of spreading activation, and is used in contexts where distance between nodes is not an issue. Rather than use activation signals, marker passers instead propagate distinct symbols from node to node; these symbols, termed *markers* or *colours,* effectively mark each node as being reachable from a given starting point. For example, Charniak (1983) uses marker passing to explore semantic memory for connecting structure that will place certain key concepts of a narrative into a coherent event framework. For instance, given the utterance “Bill felt suicidal. He reached for his belt.”, Charniak’s marker-passing model would determine a conceptual path between the concepts *Suicide* and *Belt,* one that passed through the intermediate nodes *Hanging* and *Chair.* In this way, spreading activation and marker passing can be used to fill in the conceptual blanks and facilitate the drawing of high-level inferences. Looking again to Fig. 2, we see that *Microsoft* relates to *Netscapelnc* not merely by a direct rivalry relation (i.e., both negatively affect each other), but by virtue of negatively relating to each other's market share.

If a network is highly connected, spreading activation may well visit most, if not all, of the nodes in semantic memory. As Chamiak notes, unchecked marker passing can lead to intractable processing problems, by giving rise to a potentially unbounded number of inferential pathways. Practical computational limits thus need to be placed on this idea of a conceptual domain. Typically, these limits take the form of an activation *horizon* or threshold beyond which a wave (of markers or activation) is not allowed to proceed. For instance, a threshold of 0.01 might be placed on the allowable attenuation of each activation wave, effectively causing a wave to terminate if its activation strength falls below this limit. Alternately, a fixed horizon of arcs might be placed on each wave. For instance, a horizon of 6 would mean that only those nodes reachable within 6 arcs or less from the starting point would ever be visited by the traversal process.

3.3. Graph Isomorphism

Given a representation of domain knowledge, we are in a sufficient computational position to describe the process whereby two or more domains are structurally reconciled. This reconciliation may take the form of a structural comparison, an analogical mapping, or a blend. In each case, coherent correspondences must be established between elements of each domain as a guide to the reconciliation process. In the computational literature, this correspondence problem is traditionally seen as one of sub-graph isomorphism, whereby a connected subset of nodes and arcs in one domain structure are placed into a one-to-one alignment with equivalent elements in another domain (see Falkenhainer, Forbus and Gentner, 1989; Indurkhya, 1992; Veale and Keane, 1997). The goal of the graph isomorphism algorithm is to determine the largest subset of nodes and arcs in each domain that are alignable in this way. Because the algorithm has to reason about potential groupings of nodes and arcs, of which there are a combinatorial amount in each domain structure, an optimal solution to this problem whereby the largest isomorphism subsets are determined will most likely require an exponential amount of time to attain (in computational terms, we say the problem belongs to the class NP-Hard; see Garey and Johnson, 1979).

At the heart of analogy and metaphor then is posited a structure-mapping process that is responsible for creating an isomorphic correspondence between semantic sub-structures of two domains (in metaphoric terms, the tenor and vehicle domains). Isomorphism is a mathematical notion that guarantees the systematicity and coherence of any resulting interpretation, by ensuring that each relation and object of the tenor domain receives at most one correspondence in the vehicle domain. One can argue that isomorphism is central to metaphor and analogy because, in logical/computational terms, all meaning is expressed via structure; if a cognitive process does not respect structure, it cannot respect meaning, and thus, cannot itself be a meaningful process. Indeed though it is a graphtheoretic mathematical notion, isomorphism is implicit in the writings of many non- mathematical philosophers of metaphor. Black (1962), for example, describes metaphor as a process in which a blackened sheet of glass inscribed with translucent markings (the vehicle) is placed over a visual scene like the night sky (the tenor). Since only those stars which show through the markings are visible to the observer, a sub-graph isomorphism between glass and scene is created (e.g., the stars of the *Pegasus* constellation might be picked out by a glass inscribed with a picture of a winged horse).

3.4. Path structures

The importance of structural isomorphism in counterfactual thought and argument by analogy (two phenomena for which conceptual integration theory provides an appealing model), is readily illustrated by a topical example illustrated in Figures 2 and 3. Chafing under the U.S. government's decision in a recent antitrust case against Microsoft (on behalf of the competion rights of a rival company, Netscape inc.), its CEO and chairman Bill Gates argued that to expect Microsoft to distribute Netscape Navigator as part of the Windows'98 operating system was as irrational as expecting CocaCola to bundle three cans of Pepsi with every sixpack of Coke. The analogy is a good one, for it grounds the corporate rivalry between Microsoft and Netscape in the well-appreciated, indeed almost visceral, fear and loathing that has traditionally existed between CocaCola and PepsiCo. Both of the latter sell near-identical products in an intensely competitive market, where the most apparent sources of marketability are brand recognition and customer loyalty. Like Netscape Navigator and Microsoft's rival browser, Internet Explorer (or *IExplorer),* both products have little to distinguish them at a content- level, so for a company to use its distribution mechanisms to deliver a rival's product to the market-place can be seen as financial suicide.

Highlighted in Fig. 2 and 3 are the relational chains that might conveniently be termed the *backbones* of each domain structure. In Fig. 2 we see that *Microsoft* creates (and controls) Windows'98, which in turn contains the browser IExplorer, which creates a market for itself denoted *IExplorerBase,* which in turn reinforces *Microsoft* as a company. Similarly, in Fig. 3 we note that CocaCola creates (and controls the makeup of) *CokeSixPacks,* which contain cans of Coke-branded soda, which generate a market for themselves denoted *CokeMarket,* which in turn reinforces *CocaCola's* corporate status. In representational terms suited to later algorithmic exploitation, we denote these relational chains using the notation of a semantic pathway, yielding *Microsoft — create —> Windows—part —> IExplorer — create —» IExplorerUserBase — affect —> Microsoft* and *CocaCola — create —> CokeSixPack — part —» CokeCan#6 — create —» CokeMarket — affect —> CocaCola* respectively. Both of these pathways are structurally isomorphic, and are ultimately grounded in a sub-metaphor that reconciles *MicrosoftSoftware* with *ColaSoftDrink* (both are, in a sense, "soft" products that are aimed at the mass market). This isomorphism should allow an algorithm to generate a partial interpretation of the analogy that maps *Microsoft* to *CocaCola, Windows'98* to a *CokeSixPack*, *IExplorer* to a *CokeCan* (labelled *CokeCan#6* in the semantic memory of the system) and *IExplorerUserBase* to *CokeMarket.*

Microsoft and *CocaCola* are viewed in network terms as the root concepts of each domain (and hence, of the analogy), causing all isomorphic pathways within a certain horizon, or size limit, originating at these nodes to be considered as the basis of a new partial interpretation. Typically, an algorithm should only need to consider pathways that comprise six relations or less, a modest computational bound that should nonetheless allow it to model analogical reasoning that involves six levels of recursion, a significant cognitive feat from a human perspective. So when all partial interpretations within this limit have been constructed, the algorithm will have mapped *PepsiCo* to *Netscapelnc, NetscapeNavigator* to a can of Pepsi (labelled *PepsiCan#6* in semantic memory), and *NetscapeUserBase* to *PepsiMcrrket.* It simply remains for the algorithm to choose a maximal set of partial interpretations that can be merged together to form an overall interpretation of the analogy that is rich yet internally consistent.

When the number of partial mappings is small, all possible combinations can be examined in an attempt to find a non-conflicting set that produces the richest overall mapping. When the number is too large to permit exhaustive search of this type, a heuristic approach must instead be pursued, whereby the richest partial interpretation is chosen as a the backbone of the analogy, and other interpretations are aggregated around this backbone if it does not violate structural isomorphism to do so (this latter approach is termed a greedy algorithm, since it concentrates the bulk of its processing resources into developing a single hypothesis; greedy algorithms are the basis for Falkenhainer, Forbus and Gentner's (1989) SME model, and Veale and Keane's (1994; 1997) Sapper model).

3.5. Tree structures

Our discursion so far has focussed on the algorithmic felicity of the path structure, since pathways are a natural outgrowth of semantic network organization. For instance, spreading activation is best viewed as a path-finding process, in which an activation wave originating at a particular node in memory ekes out all those pathways that can be traversed from that point to a given horizon. The activation process thus combines the problems of determining relevance (i.e., what other concepts are relevant?) with that of structural analysis (i.e.,how do these concepts relate?) and with that of metaphor recruitment (i.e., what other metaphors, or blends, must be recruited into this analysis to coherently bind a overall interpretation together?). In the Microsoft analogy of Figures 2 and 3, we have seen how spreading activation is responsible for highlighting relevant domain concepts such as Netscape and its browser, while exploring the different ways these concepts can interact, as well as drafting in a suitable sub-metaphor to tie this domain to that of *CocaCola* and *Pepsi* (i.e., viewing soft-drinks as software).

But the pathway is not the traditional structure favoured by linguists, and for that matter, many computer scientists and logicians. Many models instead rely on the notion of a tree. For instance, syntacticians use tree structures to represent both the surface and deep form of a sentence, while logicians use nested predication structures, effectively trees, to represent the higher-order meaning of such sentences. The same preference is reflected in a number of well-known models of analogical structure-mapping, such as the *Structure-Mapping Engine* (or SME) of Falkenhainer, Forbus and Gentner (1989), and the *Analogical Constraint Mapping Engine* (or ACME) of Holyoak and Thagard (1989). In these models, domain structure is represented as a collection of tree-like nested predications, in effect producing a forest of trees graph representation. For instance, in either of these models, we might represent one aspect of the relationship between *Microsoft and Netscape* as follows:

cause(cause(sell(Netscape, NetscapeBrowser),
increase(NetscapeUserBase)), decrease(IexplorerUserbase)).

In other words, Netscape's attempts to increase its user-share will have concomitantly diminishing effects on the size of Microsoft's share of the browser market.

Computational models of structural alignment can be classified into those that use tree representations and those that use path representations. In either case, the model algorithm begins by recognising structures of the appropriate type (trees or paths) in the representation of each space or domain, and then attempts to isomorphically align these sub-structures in a pair-wise fashion to create local graph isomorphisms, or partial mappings between the domains. These partial mappings (often called *pmaps)* are then selectively coalesced into larger scale, global graph mappings, either through a bruteforce search through the space of combinations, or via some greedy heuristic that bets on certain partial mappings as being more central to the overall alignment than others.

4. Computational Models of Metaphor and Analogy

In this section, we outline the basis of five current models of computational metaphor and analogy. These models, while reflecting a general agreement that consistent cross-domain mapping is central to cognitive functions such as metaphor and analogy, approach the problems inherent to establishing this mapping in very different ways. In the section that follows, we demonstrate that the design principles of one of these models, Sapper, resonates more with the notion of conceptual integration and blending than the others. In anticipation of section 5 then, we take the liberty of describing Sapper in greater algorithmic detail in the current section.

4.1. SME: The Structure-Matching Engine

As described in Falkenhainer, Forbus and Gentner (1989), SME - the *Structure Mapping Engine* - occupies one extreme of a functional continuum, and may be described as an exhaustively optimal and maximal approach to structure mapping. SME tirelessly produces all possible interpretations of a given analogical pairing, each alternate interpretation deemed maximal in the sense that no additional correspondence can be added to it without destroying its internal systematicity and coherence. Additionally, SME is optimal in the sense that it scores each alternate interpretation, and indicates the best mapping according to a predefined systematicity metric.

Actually, SME is a *configurable analogy toolkit*, capable of applying different match rules to different mapping tasks. Heuristic modifications to SME are additionally reported in Forbus and Oblinger (1990), which replace the factorial merge stage of the original algorithm with a new sub-optimal greedy merge. However, as argued in Veale et al. (1996), even with such modifications SME is largely unsuited to the mapping of structures in which richly detailed character/ object descriptions - as opposed to high level causal actions - play an important role. Such descriptions in turn frequently underlie a pervasive form of conceptual blending in language - the XYZ construct (see Fauconnier and Turner, 1998), in which two noun concepts are related against the ground provided by a third (e.g., '*Elephants* [X] were the *tanks* [Y] of Hannibal's *army* [Z]'; see Veale and Keane, 1998). In such examples, the semantic representation of each of the trio of concepts are interrelated via an integration network that binds together key object correspondences in each (e.g., *Elephant —> Tank, Hannibal —> Rommel, Scipio —> Montgomery, Zama —> ElAlameiri).*

In SME parlance, a systematic collection of inter-structure correspondences is termed a *gmap* (global mapping). Initially, a set of *kernel* or *root* gmaps is constructed by systematically comparing the corresponding arguments of identical predicates in each structure. This set is grist for the core of SME, a process which then produces successively larger combinations of these partial maps (called *pmaps)* until maximal global mappings are generated. Clearly, the size of the initial root set is a key factor in the tractability of the combination process; SME employs the notion of *structural support* to limit the size of this set, exploiting systematicity across the nested organization of predications in each structure as an evidential basis for generating new roots. However, we demonstrate in Veale et al. (1996) that this support is not at all visible to SME in object/character- based metaphors (such as *Elephant* as *Tank),* which tend to involve many shallow, tree-structured representations linked via common leaves (e.g., *ElephantHide* and *ArmourPluting* are both *Protective, Strong, External* and *Grey),* rather than a few, deeply nested trees linked via a common governing predication.

4.2. ACME: The Analogical Constraint Matching Engine

ACME, the Analogical Constraint Mapping Engine, also places great emphasis on the property of mapping systematicity, or isomorphism, but eschews the exhaustively optimal and maximal strategy pursued by SME. Instead, ACME constructs a constraint network for each new analogical problem to model the various pressures of similarity, context and isomorphism which shape the final interpretation. This network is the subject of a parallelized constraint relaxation process, from which a sole interpretation emerges, one that is neither guaranteed to be optimal, or maximal, or, for that matter, even wholly systematic. Unlike SME, ACME guarantees nothing, embodying a heuristic rather than complete approach to the problem. Indeed, ACME pursues what may be called a natural or evolutionary model of computation, in which enviromnental forces pressurize a system into converging toward a good, rather than optimal, solution (much like the *CopyCat* model of Hofstadter and Mitchell 1988; Hofstadter et al. 1995).

Like SME, ACME is a tree-based structure matcher that compares two domain descriptions in a predicate-calculus-style representation. Hierarchical structure in such descriptions - which is originally expressed via the tree-like nesting of predications - is translated into a series of inhibitory and excitatory linkages in the ACME network. Nodes in this network correspond to possible object correspondences between the source and target domains (e.g., there would exist a single node for *Microsoft as CocaCola).* Nodes representing consistent mappings reinforce each other, while nodes representing competing mappings attempt to diminish each other's activation level. Once the network is activated, the activation levels of these nodes gradually converge toward asymptotic values as the network proceeds through a succession of epochs before eventually settling. An ACME network is deemed to have settled when a certain large proportion of its nodes have reached their asymptote. Nodes which are still positively activate after the network has settled in this way then indicate the cross-space mappings of an overall interpretation.Yet while neither maximal or optimal, ACME is slower than SME, and is certainly less systematic; this result is borne out in the empirical analysis of Veale et al. (1996).

4.3. LISA: A Model of Learning, Induction and Schema Abstraction

The LISA model of Hummel and Holy oak (1996) can be seen as a conceptual descendant of Holyoak and Thagard's ACME, inasmuch as it too is a connectionist model of structure-mapping, albeit one that casts a thriftier eye on the scale of its network representations. Whereas AC IVIE constructs a large, specialized network whose number of neurons/nodes is the square of the number of concept symbols in each domain, and whose number of inter-node connections is a fourth power of this domain size, LISA employs the notion of synchrony of firing to reduce the scale of its network considerably. Rather than dedicate a network node to representing a role binding between a specific pairing of a predicate symbol and an argument symbol, LISA instead assumes that the nodes representing these symbols are related if they are both firing (emitting an output signal) *in phase.*

Another feature which characterizes the evolution of LISA from ACME is its postulation of a semantic layer of micro-feature units (representing generic concepts such as Male, Action, etc.) in which higher-order structural representations are grounded. LISA thus uses feature-based semantic criteria to judge if two conceptual structures are similar, in addition to the isomorphism- based criterion of structural similarity employed in SME and ACME. In effect, this layer of semantic features serves as a simple form of generic space against which to perform a metaphoric mapping, by providing the common semantic vocabulary with which to synthesise the input domains.

4.4. Tabletop: A Micro-domain for the Exploration of Similarity

Mining the same micro-domain vein as Hofstadter and Mitchell's (1988) Copycat architecture, the Tabletop model of Hofstadter and French (1995) and French (1995) is a non-deterministic study of the role of similarity and analogy in high-level perception. Tabletop is an environment for studying the pragmatically shifting nature of similarity judgements that one experiences during the analysis of *'do as I do analogies',* in which a *student* attempts to replicate the actions of a *master* given certain, contextually-enforced restrictions. For instance, imagine a colleague with a broken left arm immovably encased in plaster of paris, who attempts to point to the exact bone in his elbow that is damaged; unable to point directly to his left elbow (covered as it is in plaster), he uses his left hand to point to the equivalent spot on his right elbow, exclaiming 'here's the bone I broke'. People generally have no trouble in comprehending this kind of analogical action; in fact, the mapping may be so transparent as to be unremarkable to the hearer.

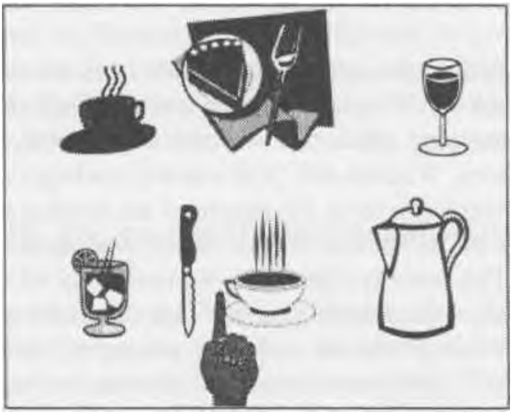


Figure 4: 'Henry' points to a coffee-cup in the Tabletop micro-domain. 'Eliza' must now choose the corresponding piece of tableware to touch in response.

Tabletop employs this kind of 'perceptual analogy under contextual constraints' to the micro-domain of a coffee-shop table-top, on which various restaurant paraphernalia have been arranged. On one side of the table sits the master (denoted'Henry' by Hofstadter and French), who touches a single item on the table; on the opposite side sits the student (denoted 'Eliza', to complete the Pygmalion metaphor), who attempts to mimic this action from her own perspective. Such a situation is illustrated in Figure 4.

Eliza chooses what she considers the best analog to Henry's choice from a host of perceptual gestalts, each informed and shaped by different pragmatic'grouping pressures'. For instance, if Henry touches his coffee cup, as illustrated in Figure 4. Eliza may respond by touching her own coffee-cup. However, this rather literal reaction ignores both the geometric position of Henry's choice (her right versus his middle) and the high-level grouping of objects in which Henry's coffee cup participates (knife to the right, glass to the far right, coffee-pot to the left). Eliza may thus find that her plate of chocolate cake better reflects the geometry and chunking of Henry's cup, and touch that accordingly.

Hofstadter and French model these various slippage pressures in a non- deterministic, stochastic manner, by embodying each pressure via a computational device termed a *'codelef.* Individual codelets represent potential choices and actions, and are given as much attention as their numeric ranking of urgency dictates. Tabletop thus moves its attention amongst a variety of different codelets, each competing at differing levels of urgency for the system's attention. Because Tabletop's attention mechanism is probabilistic over these urgency levels, different runs of the system may produce different results, yet on average, the most plausible results are produced most often.

Though restricted to a specific micro-domain, Hofstadter and French's system is nonetheless a model of considerable clarity that serves to pin-point those forces that shape an analogical/metaphoric interpretation. In an important sense, the model we describe in the next section, Sapper, is very much informed by the Tabletop model.

4.5. Sapper: A Bridge-Building Model of Structure-Mapping

Like SME and ACME, Sapper is a computational model of metaphor and analogy founded upon the notion of structure-mapping between domains (see Winston 1980; Gentner, 1983). However, unlike SME and ACME, but like Tabletop and LISA, Sapper requires that two cross-domain concepts have more in common than an isomorphic structural setting if they are to be paired in an interpretation of a given metaphor. In addition to structural isomorphism, Sapper requires that two analogical concepts either share a common set of features (abstract or concrete) or be structurally related to another pair of analogical concepts that do. Concepts that share a number of semantic features or attributes are said to be linked by a 'bridge relation', and it is upon such 'bridges' that Sapper grounds the interpretation of a given metaphor. For instance, the concepts *Scalpel* and *Cleaver* share the associations *Sharp, Blade* and *Blood,* and thus a bridge relation is established between both. Higher-level analogical correspondences can be founded upon this bridge if the corresponding concepts relate to the bridge in an identical semantic fashion; thus, because *Surgeons* use *Scalpels,* and *Butchers* use *Cleavers,* a mapping between *Surgeon* and *Butcher* can be grounded in the bridge relation between *Scalpels* and *Cleavers.* Bridges based upon low-level literal and perceptual similarities, such as *Sharpness,* correspond to basic attributive metaphors, and are considered by Sapper as instantiations of the basic mapping schema X - *metaphor* —> Y. Sapper views metaphor interpretation as a process of bridge-building in which new bridges are constructed using existing bridges as foundations; thus Sapper might construct the bridge *Surgeon - metaphor —> Butcher -* by building upon the lower-level bridges *Scalpel - metaphor —> Cleaver* or *Surgery - metaphor —> Slaughter.*

At the algorithmic core of Sapper lies a graph-matching process (see Veale and Keane, 1997 for a full complexity analysis), one which exploits the bridge schema *X - metaphor —> Y* to ensure that certain, pivotal elements of a crossdomain mapping are grounded in perceptual similarity. Sapper is also then, in an important sense, a partial theory of memory organization, inasmuch as it suggests that long-term memory is not simply a passive reservoir of information, but a reactive system that assimilates new conceptual structure by extending bridge relations to neighbouring structures that are, at a local level, semantically similar. Sapper thus employs a pro-active view of long-term memory in which shared associations between concepts are automatically recognised and noted, making low-level bridge construction a memory-centred rather than mapping-centred task. Built upon this reactive memory is a structure-mapping process that exploits these low-level bridges as construction cues for the elaboration of a global mapping between the tenor and vehicle spaces; this mapping, essentially a graph- isomorphism, serves as the semantic interpretation of a metaphor or analogy.

4.5.1. Cross-Domain Bridging in Sapper

In Sapper terminology, bridging schemas lay down bridge relations in memory between two concepts that are recognised to share some local regularity of structure. These bridges are initially dormant, since each represents a potential, rather than an actual, analogical correspondence between concepts; it remains for a structure-mapper to later confirm that a given bridge does indeed contribute to a systematic cross-domain interpretation. At this point a bridge becomes active. This *dormant I active* distinction proves very useful in lexical priming and spreading-activation applications of semantic memory, since an *active bridge* can spread activation (or pass markers) into metaphorically related domains. Analogy and metaphor in Sapper thus have a representational effect on memory, actually warping the contours of long-term conceptual representation, rather than merely relying on memory as a passive warehouse of static domain descriptions.

The Sapper model typically employs two distinct constructors to augment its long-term memory representation with new dormant bridge relations - the *Triangulation Rule* and the *Squaring Rule.* The T riangulation rule is invoked whenever two concepts share a common association or superclass; for instance, in a metaphor that relates surgeons to butchers, triangulation may occur among *HumanFleslr. Meat* and *Flesh, Scalpel: Cleaver* and *Sharp, WhiteSmock : Apron* and *Clothing,* and *OperatingTheatre : Abattoir* and *Location,* laying down dormant linkages between the concepts *HumanFlesh* and *Meat, Scalpel* and *Cleaver, WhiteSmock* and *Apron,* and *OperatingTheatre* and *Abattoir.* In essence the triangulation rule is a formalization of a similar principle which underlies the plan recognition model of Hendler (1989), in which two high-level concepts can be seen as plan analogues if they share one or more task-specific micro-features. For instance, an antique letter-opener can be recognized as a workable substitute for a knife in a *killing plan,* being an object that is sharp enough to accomplish the task at hand (murder), yet one which - unlike a knife - will not arouse suspicions at airport customs. The same intuition is used in the LISA model of Hummel and Holyoak (1996), whereby two high-level concepts can be viewed as analogous if they relate to the same set of low-level semantic features.

The *Squaring* rule is a *second-order* constructor that acts upon the linkages laid down between low-level feature concepts by the triangulation rule to build bridges between higher-level concepts associated with those similar but domain- incongruent features. For instance, it may be used to build (or *reinforce')* the bridges *Surgery: Slaughter, Scalpel: Cleaver,* and *Patient : Cleaver* in the *"surgeons are butchers"* metaphor. Sapper thus employs the squaring rule to ensure that any low-level similarities that are discovered by the triangulation rule are percolated up to higher-level concepts in a structurally coherent fashion. The use of these constructors is illustrated in Figure 5.

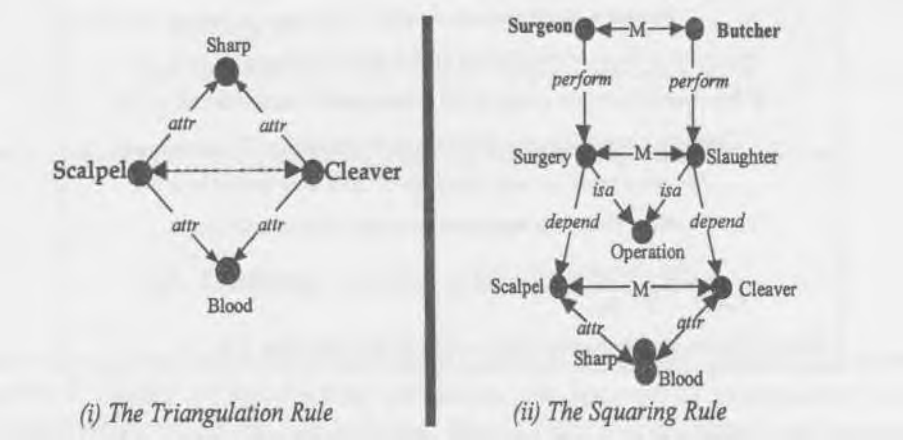


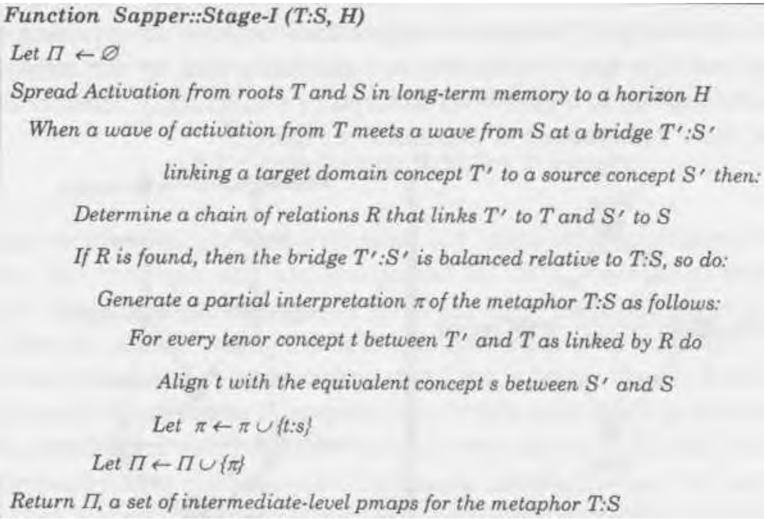
Figure 5. The Triangulation Rule (i) and the Squaring Rule (ii) augment memory with additional bridges (depicted as <— M , a shorthand for the bridging schema), indicating potential future mappings

We now turn to a consideration of how these local regularities of structure, expressed in memory as dormant conceptual bridges, are exploited by a structuremapping algorithm.

4.5.2. Structure-Mapping in Sapper

The Sapper structure-mapping algorithm comprises two consecutive structurebuilding stages. The first of these performs a bi-directional breadth-first search from the root nodes of the source space or *vehicle* (denoted S) and target space or *tenor* (denoted T) in memory, to seek out all relevant bridges that may potentially connect both domains.

This search is constrained to occur within a maximum search horizon *H* (typically *H = 6),* to avoid the combinatorial explosion that occurs due to the considerable interconnectivity of nodes representing complex concepts. This stage, a pseudo-code description of which is presented in Figure 6, produces an initial set of intermediate-level artial mappings (or pmaps) by aligning any isomorphic semantic pathways that meet at a cross-domain bridge.

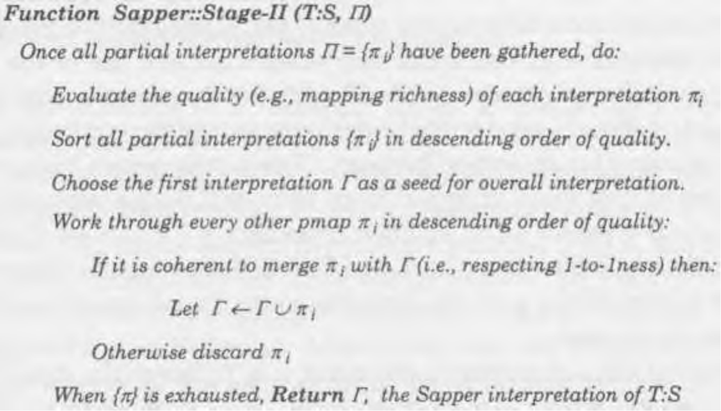


```
Function Sapper::Stage-I (T:S, H)
 Let $\Pi \leftarrow \emptyset$
 Spread Activation from roots T and S in long-term memory to a horizon H
   When a wave of activation from T meets a wave from S at a bridge $T':S'$
                linking a target domain concept $T'$ to a source concept $S'$ then:
       Determine a chain of relations R that links $T'$ 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 $\pi$ of the metaphor T:S as follows:
           For every tenor concept t between $T'$ and T as linked by R do
             Align t with the equivalent concept s between $S'$ and S
             Let $\pi \leftarrow \pi \cup \{t:s\}$
         Let $\Pi \leftarrow \Pi \cup \{\pi\}$
 Return $\Pi$, a set of intermediate-level pmaps for the metaphor T:S
```

Figure 6: The construction of intermediate-level pmaps is performed using a bidirectional search from the root nodes of both conceptual domains.

Sapper uses the same criterion of predicational matching as SME, namely: inter-concept relations (predicates) must match identically, while their arguments may not. Two semantic pathways are thus isomorphic if (i) they are of the same length; (ii) they comprise the same semantic relations, in the same order; and (iii) there exists a bijective mapping between the concept arguments of both. For instance, the source-domain pathway *Surgeon — perform —> Surgery — depend -> Scalpel — attr -> Sharp* is isomorphic with the target path *Butcher — perform —> Slaughter — depend —» Cleauer — attr —> Sharp,* and both combine to produce the pmap *(<Surgeon : Butcher>, <Surgery : Slaughters <Scalpel: Cleauer>j.*

The second stage proceeds in much the same fashion as the greedy extensions made to SME in Forbus and Oblinger (1990): the most elaborate intermediate- level pmap is chosen as a seed mapping to anchor the overall interpretation, while other pmaps are folded into this seed, if it is consistent to do so, in descending order of the richness of those pmaps. Pseudo-code for this stage is presented in Figure 7.



```
Function Sapper::Stage-II (T:S, $\Pi$)
 Once all partial interpretations $\Pi = \{\pi_i\}$ have been gathered, do:
    Evaluate the quality (e.g., mapping richness) of each interpretation $\pi_i$
    Sort all partial interpretations $\{\pi_i\}$ in descending order of quality.
    Choose the first interpretation $\Gamma$ as a seed for overall interpretation.
    Work through every other pmap $\pi_i$ in descending order of quality:
        If it is coherent to merge $\pi_i$ with $\Gamma$ (i.e., respecting 1-to-1ness) then:
                Let $\Gamma \leftarrow \Gamma \cup \pi_i$
        Otherwise discard $\pi_i$
    When $\{\pi\}$ is exhausted, Return $\Gamma$, the Sapper interpretation of T:S
```

Figure 7: The construction of global-level pmaps is performed using a seeding algorithm.

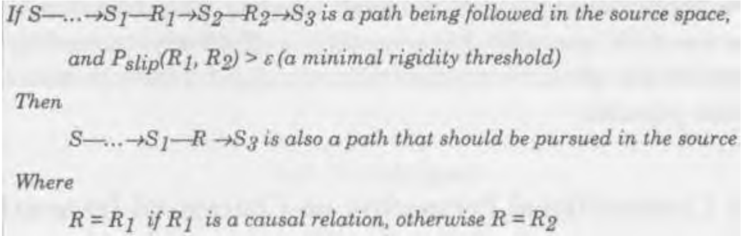
4.5. Modelling Structural Slippage in Sapper

As is evident even from a micro-domain as simple as Tabletop’s, context typically imposes a variety of interlocking pressures that necessarily complicate our decision-making processes. For instance, should Eliza touch her own coffee-cup in response to Henry touching his, even though her cup occupies a distinctly different geometric position and gestalt, or should she *loosen* her conception of coffee-cup to encompass her plate of chocolate cake, which bears some superficial similarities to a coffee-cup but which, more importantly, occupies a similar relative position to Henry’s designated item? Moving outside the closed world of the coffee-shop, and its limited conceptual repertoire (which Hofstadter and French term its *'Platobef),* we see that any cognitive process that employs structure-mapping is equally likely to demonstrate this form of slippage. The ability to fluidly reorganize one's conceptual structures in response to contextual obstacles and pressures is thus a necessary element of metaphoric, analogical, and blend- centered mapping.

Interestingly, Sapper's squaring rule does, in an important sense, already serve as a recursive mechanism for conceptual slippage, one that allows two apparently dissimilar concepts to be reconciled if two other concepts, each related in the same way to the first pair, can themselves be reconciled. For example, Surgery can be seen as a form of *Slaughter* by virtue of being able to see a *Scalpel* as a type of *Cleaver*. But as defined, Sapper is inflexible in the face of another manifestation of conceptual slippage, a form which effects the structural interrelation of these concepts. For instance, two conceptual spaces or domains may be organized according to two similar, but superficially different, sets of conceptual relations, e.g., one space may make extensive use of the substance relation (e.g., *Table - substance —> Wood)* while another uses the contain relation *{Body - part —> Blood),* yet it might be desirable to treat these relations as having the same meaning for analogical purposes. Transitivity across such relations is also an issue of conceptual slippage: if we know that in one conceptual domain *A - cause B - enable —> C,* and that in another *X - cause —>* Z, it might be pragmatically sensible to map A to X and C to Z. Following the Tabletop model, we outline in this section a simple extension to the Sapper model that allows for these forms of slippage.

Underlying this *soft* extension of Sapper is a Tabletop-like *slipnet* in which different semantic relations (i.e., graph labels) are probabilistically connected (e.g., *Popart —> contain) = 0.9).* Operating in conjunction with this slipnet is a semantically motivated structure-warping rule, which essentially softens the standard Sapper triangulation and squaring rules to allow non-isomorphic structures to be mapped. Sapper can thus comprehend analogies between domains that have been defined at different levels of detail and redundancy: for instance, in the *SportsCar* domain one might state that the *Engine* contains *Pistons* which *control* the *Wheels,* or alternately, that the *Pistons control* the *Crankshaft* which in turn *controls* the *Wheels.* When mapping this source structure then to that of either the *Jaguar* or *Puma* panther say (an analogy used by the Ford automobile manufacturer for two of its sports cars), it may be necessary to either contract or stretch the target structure to accommodate the possible occurrence of the node *Crankshaft* (which might or might not map to *LegMuscle,* say).

Given two pmaps of equal depth (i.e., each composed of paths of a given length), a probabilistic *rigidity* measure of how much slippage each involves can be ascertained, as a product of the necessary slippage probabilities entailed by each. Thus, a pmap that maps *X - part —> Y - contains Z to A - part —> B - contains C* has a rigidity measure of *1.0,* while one that maps the same path to A - *contains —» B - part —» C* has a rigidity measure of *0.9 x 0.6 - 0.54.* These measures can in turn be incorporated into a quality metric that prefers rigid pmaps over their looser variants that have *slipped.*



```
If $S—...→S_1—R_1→S_2—R_2→S_3$ is a path being followed in the source space,
        and $P_{slip}(R_1, R_2) > \varepsilon$ (a minimal rigidity threshold)
Then
        $S—...→S_1—R →S_3$ is also a path that should be pursued in the source
Where
        $R = R_1$ if $R_1$ is a causal relation, otherwise $R = R_2$
```

Figure 8: The Core Slippage Principle employed in Sapper.

Given the existence of a relational slipnet to handle label slippage, the complementary problem of structural warping can be handled with the single, compositional rule of Figure 8. The action of this rule is simple yet effective: two successive semantic relations Rj and R_2, linking two concepts S! and S_3 via an intermediary S_2, can be *snipped* to produce a path that links S, and S_3 directly; if R, is a causal relation (such as *cause, enable, support,* etc.) then it is favoured as the relation that directly connects S, and S_3; otherwise R_2 is chosen. If applied at every stage of a given pathway's development, this rule is capable of removing a significant number of linkages, as many are as needed to make the pathway structurally isomorphic with a mirror pathway in the target domain. For instance, *part —> cause* reduces to *cause,* as does *cause —> part,* while *part —> substance* and *part —> contains* both reduce to part. As illustrated in Figure 9, the concepts Engine and *Crankshaft* are temporarily removed from the source picture to accommodate a mapping between *Muscle* and *Piston.*

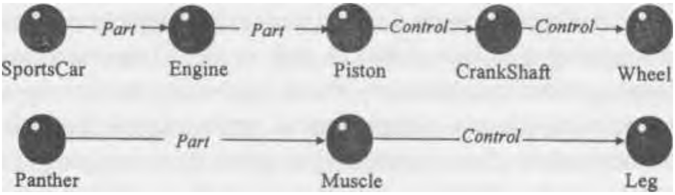


Figure 8: Path simplification in the SportsCar domain yields a path isomorphism with the Panther domain

Note that this convenient deletion of Engine is indeed both temporary and nondestructive, inasmuch as it effects just this single pathway. Other pathways that ultimately find a mirror partner in the target domain may instead provide a mapping for *Engine* (for instance, a pathway between *FuelCap* and *ExhaustPipe* will necessarily pass through *Engine,* mapping it to either *Brain* or *Heart).* Ultimately, Sapper will choose those pathways, simplified or otherwise, that collectively contribute to the richest overall mapping. The issue then of whether slippage is warranted at any level in a mapping is effectively resolved by Sapper's in-built optimality pressure to construct the most coherent and elaborate interpretation possible.

5. A Computational Perspective on Conceptual Integration

In this section we reconsider the above computational models relative to the conceptual demands of Fauconnier and Turner's blending theory. The evidence, as we see it, points most clearly to Sapper as the architecture most directly informed by the theory of conceptual blending, specifically in respect of such issues as the mediating role of generic space, the interaction of this space with the input spaces, the emergent character of blend space, and the conceptual integration that occurs between each.

5.1. Generic Space

Consider first the notion of generic space, and the algorithmic guise this notion might assume in a computational model. Recall that the generic space of a blend contains those conceptual schemas that underlie, and serve to unite, the individual constructs of the input (e.g., tenor and vehicle) spaces, thus providing the common conceptual vocabulary of the blend. A model like SME makes little or no appeal to the idea of a generic space, primarily relying as it does on structural alignment to reconcile its inputs Though SME's mapping may be influenced by the surface properties of the concepts in each domain, it does not depend on these properties to generate a mapping, and thus does not seek to recruit intermediate blends that make the mapping more compellingly vivid. Likewise, neither does the ACME architecture support an obvious counterpart to generic space, as again, this model does not look beyond its input constructs to generate a mapping.

But in contrast, the Tabletop architecture makes recourse to a *slipnet,* or probabilistic network, of platonic concepts (collectively dubbed its *'platobet'* to reason about those elements of a problem that are contextually similar to one another. Those *platobetic* concepts that become activated in the course of a mapping problem, such as that of Figure 4, can thus be construed as forming the generic space of the mapping. In a similar fashion, LISA makes recourse to a substrate layer of semantic nodes to ground the mapping of two higher-order structures, such that any node that becomes active corresponds to a semantic feature that is generic to both input spaces. Likewise, Sapper also grounds the mapping of two conceptual structures in literal similarity, via its use of the triangulation and squaring rules, so that those concepts which serve as the basis of triangulation (such as *Blood* and *Sharp* in Fig. 5(i)) can, like their counterparts in LISA, be seen as forming the generic conceptual vocabulary of the mapping.

5.2. Blend Space

Though each of the five models supports an obvious counterpart to the input spaces of a blend, few actually give computational support to the idea of an independent blend space. Again, SME and ACM E support no direct equivalent of a blend space, for since neither model supports an explicit model of semantic memory, both are powerless to describe the effects on semantic memory of a mapping, the most notable effect being the creation of a new conceptual space in which the blend is to reside. Similarly, the modification of semantic memory and the creation of new conceptual spaces (which might correspond to a structured partition of its slipnet) are beyond the remit of the Tabletop architecture. In contrast however, the LISA model makes specific, theoretical claims about the process of schema induction in metaphor and analogy, demonstrating how one can acquire, via abstraction, generic conceptual schemas from specific mapping problems. However, as the label ‘generic’ here implies, these abstractions will most likely correspond to the rarified contents of generic space rather than to the elaborate structures of blend space, since a primary function of blend space is to facilitate the accretion of emergent features around a newly blended structure.

We instead argue that Sapper best captures the notion of blend space via its use of *active conceptual bridges.* While dormant bridges serve to capture in memory the potential for combination between two input spaces, the newly awakened active bridges of a metaphor serve to explicitly represent the corresponding elements of these spaces that actually fuse to create the blend. For example, given the core bridge of a blend, such as that connecting *Composer* and *General*, it is a simple matter to employ the squaring rule in reverse to visit all those bridges with which that bridge is structurally consonant, such as *Army : Orchestra* and *Artillery : Percussion.*

Collectively then, the bridges constructed by Sapper for a given mapping (both dormant and active) correspond to the conceptual integration network that ties each of the contributing spaces together in a blend. Dormant bridges serve to relate elements of the input spaces to each other while simultaneously highlighting those elements of generic space that make the correspondence possible, whereas active bridges relate the input spaces directly to the resulting blend space. Since these bridges are represented in long-term memory as explicit traces of a specific blended space, access to a given bridge will, in turn, provide access to the concepts that connect to the end-points of this bridge. For instance, given the bridge *Cannon : Drum*, a cognitive agent can, by a process of *spreading activation* (see Quillian, 1968; Collins and Loftus, 1975; Chamiak, 1983), retrace the efferent associations of its conceptual end-points *Drum* and *Cannon* back into the domains of *Composer* and *General.* This process, which Fauconnier and Turner dub *'unpacking a blend',* is of particular cognitive and computational importance, since it underlies an agent's powers of blend introspection, decomposition and re-organization.

5.3. The Computational Necessity and Role of a Constructor Space

The conceptual foundations of blend theory are conventionally defined in terms of four spaces, namely a generic space, two input spaces and a blend space, though this basic architecture is easily extended to accommodate blends that fuse more than two inputs. However, to offer a detailed computational picture of the algorithmic processes employed in blending, we argue that one must posit a fifth, ontologically different type of space, one which we dub *'constructor space'.* While the four conventional spaces contain conceptual structures of varying degrees of abstraction and experiential grounding, this fifth space contains the computational rules of structure composition that allow these conceptual spaces to be structurally aligned and coherently fused.

These rules, or *constructors,* are inherent to all computational models of analogical and metaphoric mapping, though some models exploit constructors more transparently than others. In SME, for instance, one can employ different sets of explicit alignment rules for solving different types of mapping problem, while in contrast, the workings of ACME'S constructors are altogether more implicit, manifest only in the way different mapping nodes of the constraint network are hard-wired to each other. In Sapper, these constructors correspond to the triangulation and squaring rules, and as such, they can be seen as responsible for the dynamic construction of the generic space of a blend.

The computational interaction between spaces under this algorithmic view is illustrated in Figure 9, where those inter-space dependencies that are computational in nature are depicted as grey arrows, while those that are wholly conceptual are depicted in black. Following this diagram, one can see how the constructor space uses specific rules of structural organisation to determine the conceptual middle-ground of a blend, and thus populate the generic space with the concepts necessary to achieve an integrated network.

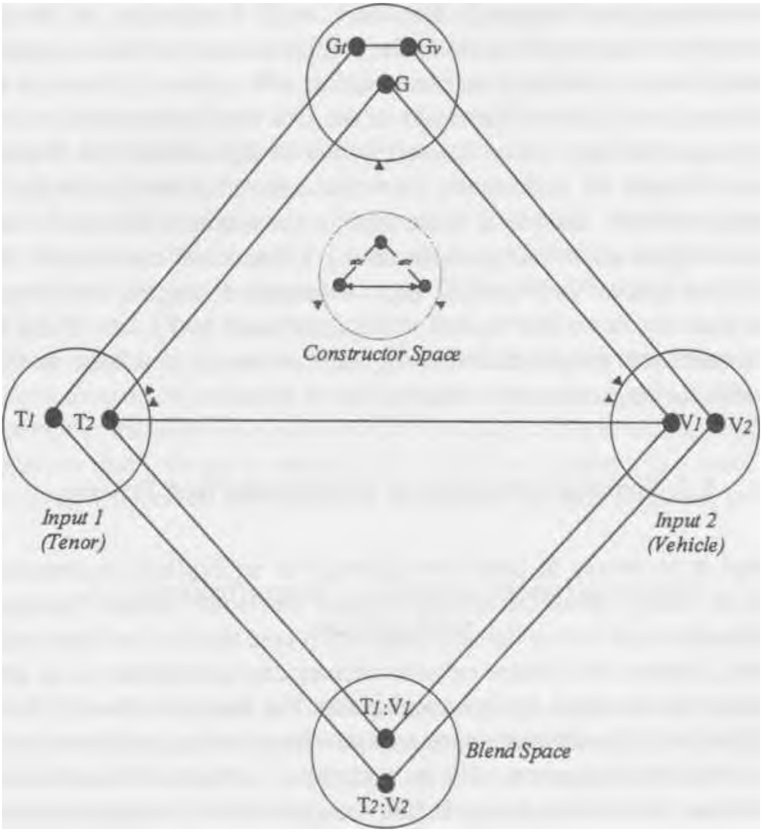


Figure 9: The Conventional 4-Space model of Blending, Augmented with a Fifth ‘Constructor’ Space when Considered from an Algorithmic Perspective.

In this particular example, the triangulation rule is used to reconcile the concepts T, and V, via a shared association *G,* causing *G* to become newly active in the generic space of the blend. Similarly, the squaring rule is used to reconcile the concepts T and V_2 via an established, underlying metaphor $G_($: G_v, causing this metaphoric schema to likewise become active in generic space. This latter mechanism can be seen at work in blends like *„Norman Mailer is Hemmingmay crossed with Patton”,* where established metaphors such as Pen as Sword may be recruited to reconcile and blend the underlying schemas of Author and General. Likewise in the metaphor *“Viagra is the new sexual rocket fuel”,* the hackneyed (but visually grounded) Freudian metaphor of *Phallus as Rocketship* can be recruited to allow a higher-level mapping between *Viagra* and *RocketFuel.*

The rules of constructor space simultaneously determine the generic content of the blend and apply this content to structurally reconcile the input spaces via a coherent isomorphic mapping. Therefore, only a selection of the possible mappings between the input spaces is chosen, since many of these mappings are unsystematic when considered in combination with others. For instance, when mapping *Surgeon* to *General* (the basis of the CNN/military metaphor *"surgical airstrike"),* one can map either *Enemy-Soldier* to *Cancer-Cell or Enemy-Army* to *Bacteria,* but not do both, since the latter does not cohere with the former. What is projected into the blend space then is a maximal collection of mutually systematic bridges, each bridge representing a fusion of counterpart elements from the input spaces. In Figure 9, input elements T_t and $V_;$ are fused in the blend space as $T_f : V_p$ while T_2 and V_2 become fused as T_2: V_2. These bridges, newly activated and established in long-term memory, may later serve as the generic basis for an even more complex future blend.

5.4, Structure-Mapping as a Paramaterized Process

We feel it necessary to posit the existence of an explicit constructor space, at the risk of complicating the already elegant four-space model, because of the apparent freedom one has in determining which constructors are most suited to a given blend. Indeed, the choice of constructors can sometimes be as important as the choice of the input spaces themselves. For instance, though the default tenants of Sapper's constructor space are the triangulation and squaring rules, an agent may feel it necessary to add an additional slippage rule, such as that of Figure 8, when the context demands that the principle of mapping isomorphism be relaxed somewhat. Indeed, some contexts actually call for the structurepreserving principle of isomorphism to be abandoned altogether, forcing an astute agent to populate its constructor space with rules that perversely destroy structure. Consider the mapping problem of Figure 10, in which an agent is asked to assign the letters A through J to the nodes of a graph such that no two adjacent nodes contain letters that are themselves alphabetically adjacent.

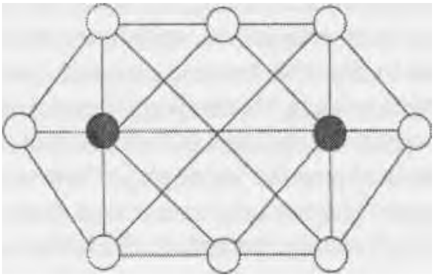


Figure 10: Map the Letters A through Z onto the ten nodes of this graph such that no two nodes contain alphabetic neighbours.

Though the problem of Figure 10 is one of reasonable complexity (there are *10!* potential mappings), it is easily solved if one views it as a structure-destroying blend or formal *dis-analogy*. The alphabetic sequence A to J can itself be organised as a graph, albeit a very linear one, where A connects to B, B to C, and so on to J. The problem can thus be rephrased as follows: map this alphabetic graph onto the graph of Figure 10, such that the adjacency structure of the former is completely destroyed. With this as the goal, a key insight is to map A and J onto the grey nodes of Figure 10, since A and J are the least connected nodes of the alphabetic sequence, while these grey nodes are the most highly connected nodes of the problem graph. With A and J in place, the problem becomes so constrained as to be near trivial. This insight, if formalised as a structure-destroying constructor (or more accurately, a *destructor)* rule, can be applied in similar, future problems that demand such a perverse mapping (e.g., the task of dividing a four-seasons pizza between two people, so that both get an equal share of all four toppings, is a more mundane example of how apparent structure must sometimes be destroyed, not preserved).

6. Applications of a Computational Account of Blending

Given a computational account of conceptual integration and blending, algorithmic models can be used to explore the role of blending in both linguistic and non-linguistic domains. A case in point in the analysis of cinematic narrative, and the prevalence of borrowing and pastiche in modem film, as described in Veale (1997).

Blended spaces, or networks of integrated concepts, serve as a powerful framework in which one can describe both the *means* and the *ends* of conceptual invention. A blend of two or more input spaces can allow a cognitive agent to create novel conceptual structures that increase either its understanding of the world, or its ability to describe that world. So from a cinematic perspective, blending theory can yield both a constructive theory of how new films are created from old, and a descriptive theory of how viewers perceive resonances between different films. We currently employ a computational model of blending - in effect Sapper - as a critical exploratory tool for examining the potential borrowings or pastiches that make up a given cinematic narrative. For instance, Sapper can be used to blend the semantic representations of *KingArthur* and *FlashGordon* to produce a structure that is highly isomorphic with that of the 1977 film *'Star Wars'*. Likewise, the domain of *BaseBall* can be blended with the same representation of KingArthur to produce a narrative structure that mirrors that of the 1984 film *'The Natural'*. Fig. 11 presents a Sapper analysis of the isomorphism that exists between the semantic network representations of *StarWars* and the *KingArthurSaga*.

If KingArthurSaga is like Star-Wars *Then* Dark-Ages is like A-Long-Time-Ago *and* Camelot is like A-Galaxy-Far-Far-Away *and* Magic is like The-Force *and* King-Arthur is like Luke-Skywalker *and* Guinnevere is like Princess-Leia *and* Lancelot is like Han-Solo *and* Merlin is like Obi-Wan-Kenobi *and* Excalibur is like Fathers-Light-Saber *and* Uther-Pendragon is like Lukes-Father *and* Mordred is like Darth-Vader *and* Mordreds-Castle is like Death-Star *and* Knights-Of-Round-Table is like Rebel-Alliance *and* Knight-Of-Round-Table is like Rebel *and* Obtain-Arthur-Grail is like Buy-Luke-C3PO-R2D2 *and* Obtain-Arthur-Excalibur is like Obtain-Luke-Fathers-Light-Saber *and* Advise-Merlin-Arthur is like Advise-Obi-Wan-Luke *and* War-Mordred-on-Camelot is like War-Empire-on-Rebeis *and* Battle-Arthur-Mordred is like Attack-Luke-Darth-Vader *and* Kill-Arthur-Mordred is like Defeat-Luke-Darth-Vader *and* Marry-Arthur-Guinnevere is like Love-Luke-Princess-Leia

Fisure 11. Structural Basis for the metanhor “Star Wars “ is the “Arthur Sasa”.

Computational tools such as Sapper allow an investigator to also explore the role of multiple borrowings in the same narrative. For instance, as argued in Veale (1997), *‘Star Wars’* is a pastiche that combines many different original sources, from the Arthurian Sagas and *‘Flash Gordon'* to *‘The Dambusters’* (1954), *‘Shane’* (1953) and even Akira Kurasawa’s *‘The Hidden Fortress’* (1958). Fig. 12 presents an analysis of an additional isomorphism between the narratives of *‘Star Wars’* and *‘The Dambusters’*.

Interestingly, the notion of a generic space in blending theory resonates well with what is commonly called the genre of a film (indeed, it is more than coincidence that both words derive from a common root). That is, we see generic space as containing those narrative conventions and expectations that frequently guide our appreciation of a film, and which allow us to determine conceptual similarities, both superficial and deep, between different narratives. Once a pair of narratives have been analysed by Sapper (or more than two, if analysed pair-wise), the generic space that has been dynamically constructed is open to analysis, allowing one to explore the common foundations on which superficially different narratives have been constructed.

If Star-Wars is like The-Dambusters Then A-Galaxy-Far-Far-Away is like Europe *and* The-Empire is like The-Third-Reich *and* Darth-Vader is like Himmler *and* Rebel-Base is like Blightey *and* A-Long-Time-Ago is like The-Forties *and* X-Wing-Fighter is like Lancaster-Bomber *and* Space-Rebel is like Allied-Soldier *and* Rebel-Alliance is like The-Allies *and* Rebel-Command-Centre is like Allied-Command-Centre *and* Death-Star is like Ruhr-Dam *and* Death-Star-Waste-Vent is like Ruhr-Dam-Sweet-Spot *and* Build-Empire-Death-Star is like Build-Germany-Ruhr-Dam *and* Declare-War-Empire-Rebels is like Declare-War-Germany-Allies *and* Advise-Comm-Centre-Rebels is like Advise-Comm-Centre-Allies *and* Destroy-Rebels-Death-Star is like Destroy-Allies-Ruhr-Dam *and* Celebrate-Rebel-Alliance is like Celebrate-Allies

Figure 12: Structural Basis for the Metaphor *“Star Wars"* is *“The Dam Busters”*

Computational models of conceptual integration not only enable such conceptual analyses, but allow the theorist to hypothesise about perceived shortfallings in the conception of the theory itself For instance, Veale (1997) notes that an adequate study of cinematic pastiche must account for blending not only at the level of narrative, but also at the level of character composition. In *‘Star Wars’,* for example, the character of *HanSolo* is clearly a pastiche of numerous sources, from the knightly Lancelot in the Arthurian blend to the gunfighter in *‘Shane’,* with additional elements of *‘Buck Rogers’* and the pirate smuggler of *‘Captain Blood’* thrown in for good measure. As noted by Eco (1995) in his rather churlish analysis of the film *‘Casablanca’* (1943), this type of *intertextual collage* is very prevalent in modem film where characters can exhibit different personae in different scenes to reflect plot and setting. The exact composition of a character blend can thus vary from scene to scene, as dictated by the flow of the narrative. Han Solo first appears as a selfish pirate and cowboy, but later sheds some of the negative associations of these sources in favour of a more heroic, knightly demeanor.

In response to these requirements, Veale (1997) proposes a dynamic extension to the theory of conceptual integration. The general structure of the proposed extension is illustrated in Fig. 13.

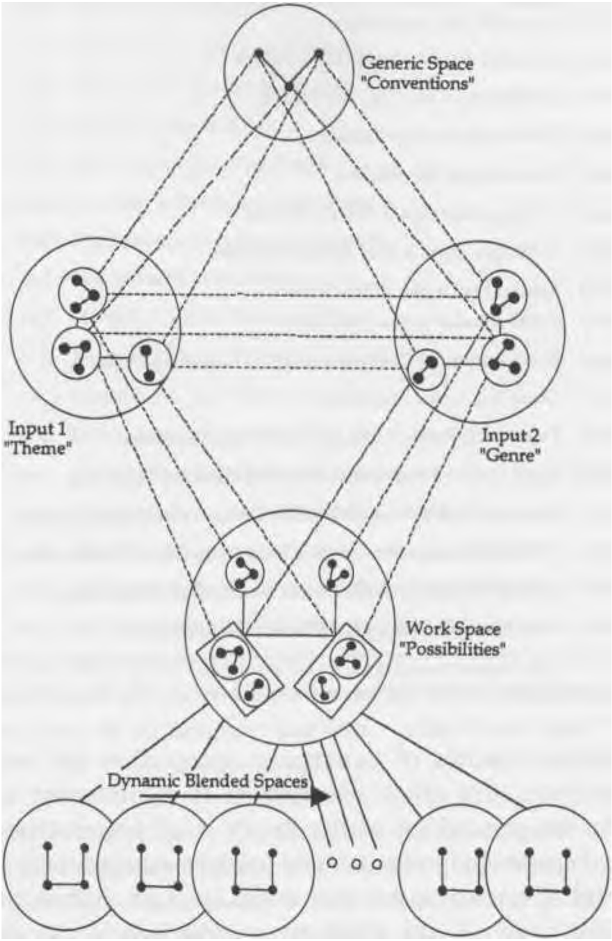


Figure 13. A revised blended space model which accounts for the dynamic nature of complex narrative blends.

In the extended scheme of Fig. 13, an additional conceptual space, dubbed the *workspace,* is inserted into the standard model to mediate between the input spaces and the blend space (much like the constructor space was inserted to mediate between the inputs and generic space). This workspace provides a synchronic picture of the possibilities inherent in the blend space, bringing together all aspects that contribute to the blend at each level of the narrative (from overall plot structure to individual character traits). For instance, in the workspace *HanSolo* is represented as an equal measure of *Pirate*, *Cowboy*, *Smuggler* and *Knight.* From this workspace a diachronic succession of individual blend spaces can be created, to reflect different levels of blend composition at different points in the narrative (e.g., *HanSolo* as *Pirate* precedes *HanSolo* as *Knight).*

We have yet to model this aspect of blend theory computationally, since Sapper currently lacks a model of temporal progression. Interestingly, however, it is this very lack that brought to our attention the role of dynamic blend composition in the first place. Even in the absence of concrete implementations, a computational perspective on cognitive processes can prove to be a useful aid to theory development.

7. Summary and Conclusions

This paper has argued that a computational perspective on cognitive theories such as conceptual integration provide not only a proof of concept for these theories, but also serve to elucidate important aspects of the theory that might otherwise go unspoken. Supplementary to the primary spaces of the basic theory, two new spaces have been proposed: the *constructor space*, which we posit to contain those parametric structural schemas necessary for building the generic space from the given inputs, and the workspace, which we claim is a necessary intermediary between the inputs and blend space when integrating situations in which temporal progression plays a key role.

In addition, we have argued that a particular computational model of structural integration, called Sapper, resonates more deeply with the principles of blending theory than any other model in the computational literature. As well as exhibiting computational tractability in the form of polynomial run-time requirements, Sapper also provides explicit algorithmic support to the optimality constraints proposed by Fauconnier and Turner (1998) that serve to curb the expressive power of an integration network.

Sapper's *construction of* bridges in long-term memory means that blended concepts have a first-class representation that can be manipulated as a single entity. For instance, a bridge between *Lancelot* and *FlashGordon* can be labelled with the symbol *HanSolo,* to reflect their contribution to this blended concept, and manipulated either as a whole, or compositionally as a pairing of two concepts (the integration constraint). Since bridges monotonically add to the structure of semantic memory, rather than taking from it, the concepts linked by a bridge still maintain their connectivity to the rest of semantic memory (the *web* constraint). Given a particular bridge, the squaring rule can be used to find neighbouring bridges in the same conceptual space with which it is structurally coherent. This allows a blended concept to be deconstructed to reveal the additional conceptual structures that contribute to its organization (the *unpacking* constraint). Sapper's focus on graph isomorphism as the basis of establishing cross-space correspondences also trivially satisfies the *topology* constraint. But since isomorphism is but one of several constraints acting on the formation of a blend, it must sometimes be relaxed in favour of other considerations (e.g., that the blend be transparent, or appear to be pragmatically relevant). For this reason, Sapper provides for a principled means of structure slippage, making isomorphism a strong preference rather than an inviolable demand. Finally, Sapper's use of spreading activation across a semantic network representation of memory means that once certain elements of a space are placed into correspondence by a blend, the squaring rule will recursively attempt to find correspondences for concepts directly related to these elements. Every concept pairing that becomes incorporated into a Sapper analysis is thus explainable either as an obvious mapping of counterparts (e.g., *HanSolo* as *Shane),* or as a structural consequence of this mapping (e.g., *DarthVader* as Jack Palance's villain in *Shane),* thereby respecting the good reason constraint.

We conclude by noting that the implementation of Sapper described in this paper, in addition to the semantic network representations of the spaces used (such as *StarWars* and *KingArthurSaga),* are available on-line from the author's web-site:

http://www.compapp.dcu.ie/~tonyv/metaphor.html

References

Black, M. 1962. *Models and Metaphor: studies in language and philosophy.* Ithaca, NY: Cornell University Press.

Eco, U. 1995. *Faith in fakes: travels in hyperreality.* Translated from the Italian by William Weaver. London: Minerva.

Chamiak, E. 1983. *Passing Markers: A Theory of Contextual Influence in Language Comprehension.* Cognitive Soience 7, pp 171-190.

Collins, A. & E. F. Loftus. 1975. A Spreading-Activation Theory of Semantic Processing. Psychological Review 82, pp 407-428.

Culler, J. (1981). The Problem of Metaphor, in *Language, Meaning and Style: Essays in Memory of Stephen Ullman.* Leeds, UK: Leeds University Press.

Cunningham, P. & T. Veale. 1991. Organizational issues arising from the integration of the Concept Network and Lexicon in a Text Understanding System, in the *Proceedings of the 12th Imernational Joint Conference on Artificial Intelligence.* San Mateo: Morgan Kaufman.

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

1994. Conceptual projection and middle spaces. UCSD: Department of Cognitive Science Technical Report 9401.

Fauconnier, G. & M. Turner. 1998. Conceptual Integration Networks. Cognitive Science, 22:2. pp 133-187.

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

French, R. 1995. *The Subtlety of Sameness.* Cambridge: MIT Press.

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

Gentner, Dedre. 1983. Structure Mapping: A Theoretical Framework for Analogy. Cognitive Science 7:2. pp 155-170.

Hawking, S. 1975. *Particle Creation by Black Holes.* Communications in Maths Physics 87. pp 199-220.

Hendler, J. A. 1989. Marker Passing over Micro-Features: Toward a Hybrid Symbolic/ Connectionist Model, Cognitive Science 13:1.

Hofstadter, D. & M. Mitchell. 1988. Conceptual Slippage and Analogy-Making: A report on the CopyCat Project, in Proc, of the IOth Annual Conference of the Cognitive Science Society, Montreal, Quebec.

Hofstadter, D. & M. Mitchell & the Fluid Analogy Research Group. 1995. *Fluid Concepts and Creative Analogies: computer models of the fundamental mechanisms of thought.* NY: Basic Books.

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

Hummel, J. E. & K. J. Holyoak. 1996. LISA: A Computational Model of Analogical Inference and Schema Induction, in the *Proceedings of the Eighteenth Annual Meeting of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

Hutton, J. 1982. Aristotle’s Pootics. NY: Norton.

Indurkhya, B. 1992. *Metaphor and Cognition: Studies in Cognitive Systems.* Kluwer Academic Publishers, Dordrecht: The Netherlands.

Koestler, A. 1964. The Act of Creation. London: Picador.

Lakoff, G. 1987. *Women, Fire and Dangerous Things.* Chigaco, Illinois: University of Chicago Press.

Lakoff, G. & Mark Johnson. 1980. Metaphors We Live By. Chigaco, Illinois: University of Chicago Press.

Quillian, M. R. 1968. Semantic Memory, in Semantic Information Processing, ed. Marvin Minsky. Cambridge, MA: MIT Press.

Richards, I. A. 1936. *The Philosophy of Rhetoric.* NY: Oxford University Press.

Veale, T. 1997. Creativity as Pastiche: A computational treatment of metaphoric blends, with special reference to cinematic “borrowing”, in the *Proceedings of Mind* II: *Computational Models of Creative Cognition,* Dublin, Ireland, September 1997. (available on-line at http://www.compapp.dcu.ie/~tonyv/Pastiche/Pastiche.html).

Veale, T. & Mark T. Keane. 1992. Conceptual Scaffolding: A spatially founded meaning representation for metaphor comprehension, Computational Intelligence 8:3.

Veale, T. & D. O'Donoghue & M. T. Keane. 1996. 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). John Benjamins.

Veale, T. & M. T. Keane. 1994. Belief Modelling, Intentionality and Perlocution in Metaphor Comprehension, in the *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society,* Atlanta, Georgia. Hillsdale, NJ: Lawrence Erlbaum.

Veale, T. & M. T. Keane. 1997. The Competence of Sub-Optimal Structure Mapping on 'Hard' Analogies, to be pre.vented at IJCAI'97, the International Joint Conference on Artificial Intelligence, Nagoya, Japan, August 1997.

Veale, T. & M. T. Keane. 1998. Principle Differences in Structure-Mapping, in the *Proceedings of Advances in Analogical Research,* eds. Keith Holyoak, Dedre Gentner and Boicho Kokinov. NBU.S'eries in Cognitive Science, Sofia 1998.

Winston, P. H. 1980. Learning and Reasoning by Analogy, Communications of the Association for Computing Machinery, 23:12. 1980.