submitted to:

Bootstrapping knowledge representations: from entailment meshes via semantic nets to learning webs

Francis HEYLIGHEN

Center "Leo Apostel", Free University of Brussels, Pleinlaan 2, B-1050 Brussels, Belgium

E-mail: fheyligh@vub.ac.be

http://pespmc1.vub.ac.be/HEYL.html

ABSTRACT. The symbol-based, correspondence epistemology used in AI is contrasted with the constructivist, coherence epistemology promoted by cybernetics. The latter leads to bootstrapping knowledge representations, in which different parts of the cognitive system mutually support each other. Gordon Pask's entailment meshes and their implementation in the THOUGHTSTICKER program are reviewed as a basic application of this methodology. Entailment meshes are then extended to entailment nets: directed graph representations governed by the "bootstrapping axiom", determining which concepts are to be distinguished or merged. This allows a constant restructuring and elicitation of the conceptual network. Semantic networks and frame-like representations with inheritance can be expressed in this very general scheme by introducing a basic ontology of node and link types. Entailment nets are then generalized to associative nets characterized by weighted links. Learning algorithms are presented which can adapt the link strengths, based on the frequency with which links are selected by hypertext browsers. It is argued that these different bootstrapping methods could be applied to make the World-Wide Web more intelligent, by allowing it to self-organize and support inferences through spreading activation.

1. Introduction

In my contribution to this special issue devoted to the memory of Gordon Pask, I wish to review my research on knowledge representation and knowledge acquisition. Although it was started independently, this research program in a number of ways parallels and extends Pask's work on Conversation Theory, and in particular its representation through entailment meshes and their implementation in the computer program THOUGHTSTICKER (Pask, 1975, 1976, 1980, 1984, 1990; Pask, & Gregory, 1986). My work started around 1983 with the development of a "structural language" for representing the fundamental space-time structures of physics (Heylighen, 1990a). This primitive modelling scheme plays a role similar to Pask's (1984) "protolanguage" or "protologic" Lp (cf. Heylighen, 1992). An investigation into artificial intelligence (AI) and cognitive science made me understand that the main application of this language might lie in knowledge representation rather than in the foundations of physics. However, the limitations of AI made me turn to cybernetics as a more general theoretical framework (Heylighen, 1987, 1990).

I first came into contact with Gordon Pask through the conference "Self-Steering and Cognition in Complex Systems" (Heylighen, Rosseel & Demeyere, 1990) which I co-organized at the Free University of Brussels in 1987. As one of the founding fathers of cybernetics, Pask was invited to give a keynote speech (Pask, 1990). In his inimitable style, he reviewed his work on conversation theory and Lp. His manners of an archetypical British eccentric and his dandyish appearance, with bow-tie, cape and walking-stick, did not fail to impress the audience, myself included. At that moment, I did not yet understand the relevance of his approach for my own work, though.

I had the chance to get better acquainted with the Paskian philosophy during a short stay at the University of Amsterdam in 1988-1989. There I worked for the research program "Support, Survival and Culture", headed by Gerard de Zeeuw and Gordon Pask. The program's focus on computer-supported collaborative work incited me to implement my ideas into a prototype software system, the CONCEPTORGANIZER, which supports one or more users in exteriorizing their implicit knowledge in the form of a hypertext network of concepts (Heylighen, 1989, 1991a). Although I was not consciously influenced, the similarity in both intention and name between this program and Pask's THOUGHTSTICKER seems obvious in retrospect.

My interest in the foundations of cybernetics and in conceptual networks led me to join Valentin Turchin and Cliff Joslyn in the creation of the Principia Cybernetica Project (Joslyn, Heylighen & Turchin, 1993). The project's aim is the computer-supported collaborative development of an evolutionary-cybernetic philosophy. Its results are presently implemented as a large hypertext net on the World-Wide Web (Heylighen, Joslyn & Turchin, 1997). Although some cyberneticists reacted critically to the project's foundational ambitions, it was welcomed enthusiastically by Gordon Pask. In his contribution to the project's first workshop, he noted that "a philosophy of Cybernetics, encapsulated in the [...] title, 'Principia Cybernetica', is not only justifiable, but necessary and in this day and age, utterly essential" (Pask, 1991). In my own contribution, I sketched an extension of my work on the structural language to encompass semantic networks as a possible framework to structure the knowledge produced by the project (Heylighen, 1991b).

A focus on the philosophical content as well as on the organizational and technical implementation of the project during the following years kept me from further developing my scheme for knowledge representation. However, in 1994, when I was joined by Johan Bollen, we started to work on a new application, an associative network of concepts that would self-organize or "learn" from the way it is used (Bollen & Heylighen, 1996). Although I knew his health was deteriorating, I had hoped to present this work to Gordon Pask during the 13th European Meeting of Cybernetics and Systems in Vienna (April 1996), where we were both chairing a symposium. Unfortunately, Gordon died a few weeks before the congress, and the meeting became the first occasion for those who had known him to reminisce and pay tribute to his work.

The present paper will review my on-going research on "bootstrapping" methods for knowledge representation, emphasizing the similarities (and differences) with Pask's approach. Although this review will mostly summarize ideas scattered over different papers (Heylighen, 1989, 1990a, 1990d, 1991a, 1991b; Bollen & Heylighen, 1996, 1997; Heylighen & Bollen, 1996), some of the results included here have not been published before.

2. Epistemology: correspondence versus coherence

Both Pask's work on knowledge representation and my own are distinguished from the more traditional AI approach by their underlying epistemology. Most AI work implicitly assumes a correspondence epistemology, which sees knowledge as a simple mapping or "reflection" of the external world. Every conceptual object (symbol) in the knowing subject's model is supposed to *correspond* to a physical object in the environment. The structure of the model can be seen as a homomorphic map, or an *encoding*, of the structure of outside reality. This epistemology, which has been called the "reflection-correspondence" theory by Turchin (1993), leads to a host of conceptual and practical problems (Bickhard & Terveen, 1995).

The most pressing ones center around the origin and nature of the mapping from reality to its symbolic representation. Since a cognitive system has no access to reality (Kant's "Ding an Sich") except through perceptions—which are already internal models—, how can it ever determine whether it uses a correct mapping? Another formulation of this difficulty is the *symbol grounding problem* (Harnad, 1990): how are the symbols, the elements of the model, "grounded" in the external reality which they are supposed to represent? This problem cannot be solved within the model itself. This follows from the "linguistic complementarity" principle (Löfgren, 1991), which generalizes classic epistemological restrictions such as the theorem of Gödel or the Heisenberg indeterminacy principle. It states that *no language can fully describe its own description or interpretation processes*. In other words, models cannot include a representation of the mapping that connects their symbols to their interpretations.

Because there is no inherent procedure to determine a correct mapping, models in AI tend to be arbitrarily imposed by the system's programmer or designer. The model's foundations or building blocks, the symbols, are primitives which have to be accepted at face value, without formal justification. The model can be more or less adequate for the problem domain, but never complete, in the sense of covering all potentially relevant situations. As argued by van Brakel (1992), this 'problem of complete description' generalizes the famous 'frame problem' in AI (Ford & Hayes, 1991). Incompleteness would not be a real obstacle if the models could adapt or learn, that is, extend their capabilities each time a problem is encountered. But the correspondence philosophy does not allow any simple way for a model to be changed. New symbols cannot be derived from those that are already there, since they are supposed to reflect outside phenomena to which the model does not have access. Introducing a new symbol must be done by the programmer, and requires a redefinition of the syntax and semantics of the model. Thus, models in AI tend to be static, absolutist and largely arbitrary in structure and contents.

These problems have helped focus attention on an alternative epistemological position, *constructivism*, which is espoused by most cyberneticists, and emphasized in "second-order" cybernetics (von Foerster, 1996) and the theory of autopoiesis (Maturana

& Varela, 1992). According to this philosophy, knowledge is not a passive mapping of outside objects, but an active construction by the subject. That construction is not supposed to reflect an objective reality, but to help the subject adapt or "fit in" to the world which it subjectively experiences.

This means that the subject will try to build models which are *coherent* with the models which it already possesses, or which it receives through the senses or through communication with others. Since models are only compared with other models, the lack of access to exterior reality no longer constitutes an obstacle to further development. In such an epistemology, knowledge is not justified or "true" because of its correspondence with an outside reality, but because of its coherence with other pieces of knowledge (Rescher, 1973; Thagard, 1989). The problem remains to specify what "coherence" precisely means: mere consistency is clearly not sufficient, since any collection of unrelated facts is logically consistent.

Model construction can be seen as a trial-and-error process, where different variations are generated, but only those variations are retained which "fit in" with the rest of the experiential material. Thus, the process is *selectionist* (Cziko, 1995; Bickhard & Ter Veen, 1995) rather than *instructionist*: instead of instructing the subject on how to build a model, the (inside or outside) environment merely helps it to select the most "fit" models among all of the subject's autonomously generated trials.

Many constructivists tend to emphasize the role of social interaction in this selection process: those models will be retained about which there is a consensus within the community. This is the social constructivist position, which is popular especially in the social sciences and humanities. Psychologists like von Glasersfeld (1984) and Piaget (1937), on the other hand, emphasize the individual subject, who tries to find coherence between his or her different models and perceptions. The selectionists inspired by Popper's (1959) evolutionary epistemology and theory of falsification, finally, emphasize the role of the outside environment in weeding out inadequate models. For obvious reasons, this more "realist" position is most popular in the natural sciences. My own philosophy is pragmatic, and acknowledges the combined role of individual, social and physical ("objective") factors in the selection of knowledge (Heylighen, 1993, 1997).

Although constructivists usually also accept some form of a coherence view of truth, few people have proposed concrete mechanisms that show how the dynamic process of construction (variation) can result in the static requirement of coherence (selective retention). The originality of Pask's Conversation Theory is that it provides a detailed formal model of such mechanisms. The metaphor of *conversation* is aptly chosen to describe such a process of cognitive interaction, in which concepts are exchanged, combined and recombined (the construction phase), with the aim of achieving agreement about shared meanings (the coherence phase). Although the conversation metaphor seems to favor the social construction of knowledge, Pask is quick to point out that his theory applies equally well to interactions between different roles or perspectives ("p-individuals") within a single individual. I have suggested elsewhere (Heylighen, 1990c) that the conversational perspective could even be extended to interactions between observers and objects. Pask might not have disagreed with this generalization, since he saw his THOUGHTSTICKER program as a way to turn a computer into a conversational partner.

To describe my own solution to the problem, however, I prefer the metaphor of *bootstrapping* (cf. Heylighen, 1990a, 1990d, 1992). As said, the problem with correspondence epistemologies is that they lack grounding: everything is built on top of the symbols, which constitute the atoms of meaning; yet, the symbols themselves are not supported. The advantage of a coherence epistemology is that there is no need for a fixed ground or foundation on which to build models: coherence is a two-way relation. In other words, coherent concepts *support each other*. The dynamic equivalent of this mutual support relation may be called "bootstrapping": Model A can be used to help construct model B, while B is used to help construct A. It is as if I am pulling myself up by my own bootstraps: while my arms (A) pull up my boots (B), my boots at the same time—through my legs, back and shoulders—push up my arms. The net effect is that more (complexity, meaning, quality, ...) is produced out of less. This is the hallmark of *self-organization*: the creation of structure without need for external intervention.

I will now show how this bootstrapping philosophy can be applied to the practical problem of knowledge representation, first by reviewing Pask's entailment meshes, then by extending the underlying formalism to my own entailment nets.

3. Entailment Meshes and the THOUGHTSTICKER program

During the development of Pask's Conversation Theory, it turned out that many of its statements could be succinctly expressed by graphical schemes that were called "entailment meshes". These informal representations were later interpreted as expressions in a formal language of distinction and coherence, which Pask called Lp. The manipulation of Lp expressions was facilitated by an interactive computer program called THOUGHTSTICKER (Pask, 1984; Pask & Gregory, 1986). As the name implies, the main aim of THOUGHTSTICKER is to help users exteriorize their thoughts in the form of a stable and explicit knowledge representation. (The same aim underlies my own CONCEPTORGANIZER prototype (Heylighen, 1989, 1991a). I will here summarize the representation of knowledge through entailment meshes, and the rules that are embedded in the THOUGHTSTICKER software for manipulating and eliciting these meshes.

The basic elements of an entailment mesh are called *topics*. They are "public" concepts, in the sense that their meaning is shared by a number of conversational participants. These concepts are connected through *coherences*. A coherence is a collection or cluster of topics which are so interrelated that the meaning of any topic of the coherence can be derived from the meaning of the other topics in the cluster. In other words, the topics in a coherence *entail* or define each other. A simple example of a coherence is the cluster <**pen, paper, writing**>. This means that we can somehow start from the concepts of **writing** and **paper** and produce the concept of an instrument that allows you to write on paper: a **pen**. Complementarily, we can start from the notions of **pen** and **paper** and derive from them the activity you do when applying a pen to paper: **writing**.

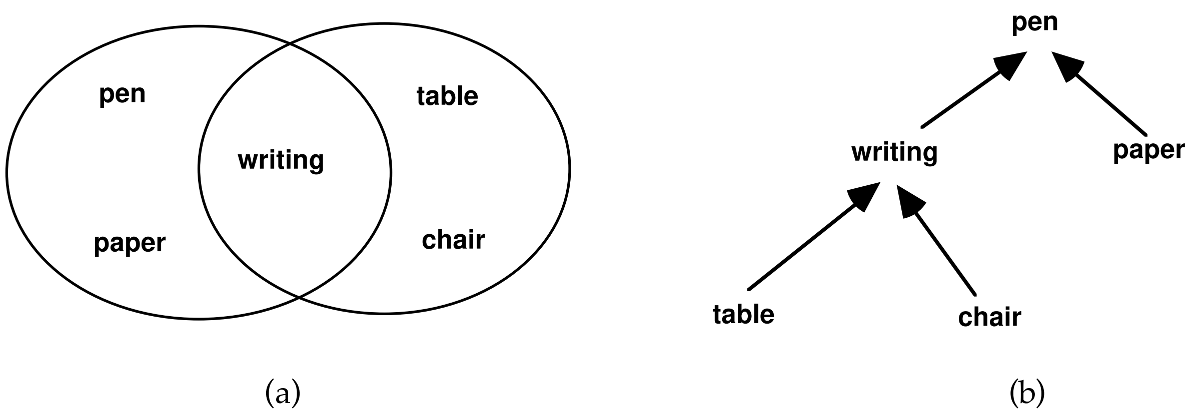

Fig. 1: (a) a simple entailment mesh consisting of two coherences overlapping on the topic **writing** ; (**b**) a 'prune' unfoldment of the same mesh from the point of view of the topic **pen.**

The same topic, e.g. **writing**, can belong to different coherences, for example: **<pen, paper, writing>** and **<table, chair, writing>** (see Fig. 1a). In other words, coherences can overlap in one or more topics. An *entailment mesh* is then a complex of overlapping coherences, that is, a collection of topics and coherences such that every topic belongs to at least one coherence (see Fig. 2). Every topic in an entailment mesh should be entailed by, or unambiguously derivable, from the other topics in the mesh. This derivation can be represented in THOUGHTSTICKER by the operation called 'Prune'. Prune produces an unfoldment of the mesh from the perspective of the concept you want to derive. For example, Fig. 1b shows a pruning of the mesh in Fig. 1 from the point of view of **pen**. With symbols this derivation could be represented as: **pen** ← {**writing**, **paper**}, **writing** ← {**table**, **chair**}.

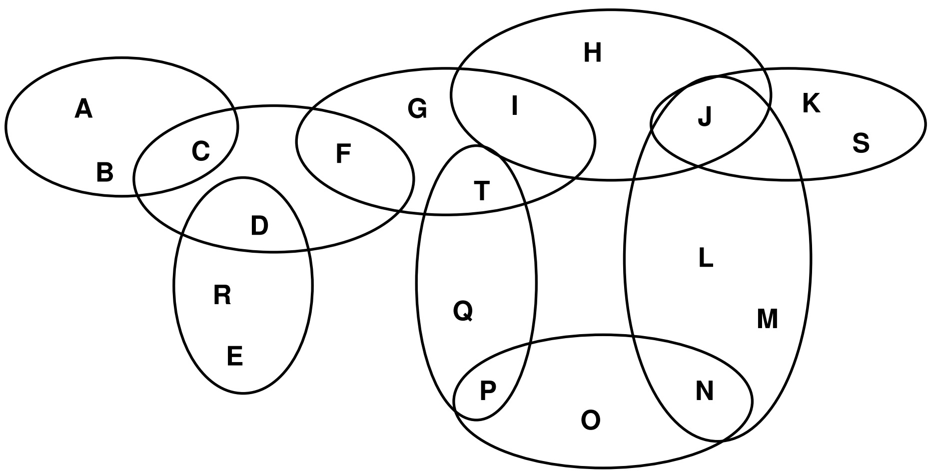

Fig. 2: a more complex entailment mesh

The main function of the Prune operation is to discover structural ambiguities or conflicts in a mesh. An ambiguity arises when different topics are derived in the same way. In that case, there is no way to distinguish the topics within the mesh. Consider for example the two coherences **<pen, paper, writing>** and **<pencil, paper, writing>**. In this case, it is not clear which of the two topics **pencil** or **pen** should be produced from **paper** and **writing**: **pen** ← {**writing**, **paper**} and **pencil** ← {**writing**, **paper**} both hold. This poses a problem for the knowledge representation, which can be resolved in one of the following ways:

- *merge topics*: the ambiguous topics may be really equivalent, in the sense that either they are synonyms (e.g. **persons** and **people**), or they are two tokens (e.g. **pen** and **pencil**) standing for the same basic concept (e.g. **writing device**). The resolution is to merge the topics, thus producing the single new coherence: **<writing device, paper, writing>**
- *add topic*: if the topics are really distinct, this must be reflected in the entailment mesh by changing the coherences that define them. The simplest way to do that is to add a topic to one of the two coherences. For example, one could replace **<pencil, paper, writing>** by **<pencil, paper, writing, erasable>**. The derivation of **pencil** from its coherence cluster is now different from the derivation of **pen**, and thus the ambiguity is resolved.
- *split topic*: another way to create distinction where there is none is to split or "bifurcate" one of the topics occurring in both coherences into two distinct topics. For example, **writing** could be split into **writing letters** and **writing notes**, resulting in the two coherences **<pen, paper, writing letters>** and **<pencil, paper, writing notes>**. Since the two new topics derive from the same underlying idea, they remain related in THOUGHTSTICKER by a relation of *analogy*.
- *merge coherences*: another way to eliminate the ambiguity is to replace the two coherences by one new coherence including all topics, e.g. **<pen, pencil, paper, writing>**. Now, **pen** and **pencil** have each a different derivation.

Another type of ambiguity can arise when coherences are nested, i.e. when one coherence, e.g. **<pen**, **pencil**, **ball-point>**, is a subset of another coherence, e.g. **<pen**, **pencil**, **ball-point, paper, writing>**. In that case, it is not clear how to derive a topic belonging to both coherencies, e.g. is it **pen** ← {**writing**, **paper, ball-point, pencil**} or **pen** ← {**ball-point, pencil**}? Such a nested construction is illegal in THOUGHTSTICKER. The way to resolve it is to "condense" the topics of the inner cluster (**<pen**, **pencil**, **ball-point>**) into a new, "generalized" concept, e.g. **writing device**. This leaves a single new cluster, **<writing device, paper, writing>** (see Fig. 3).

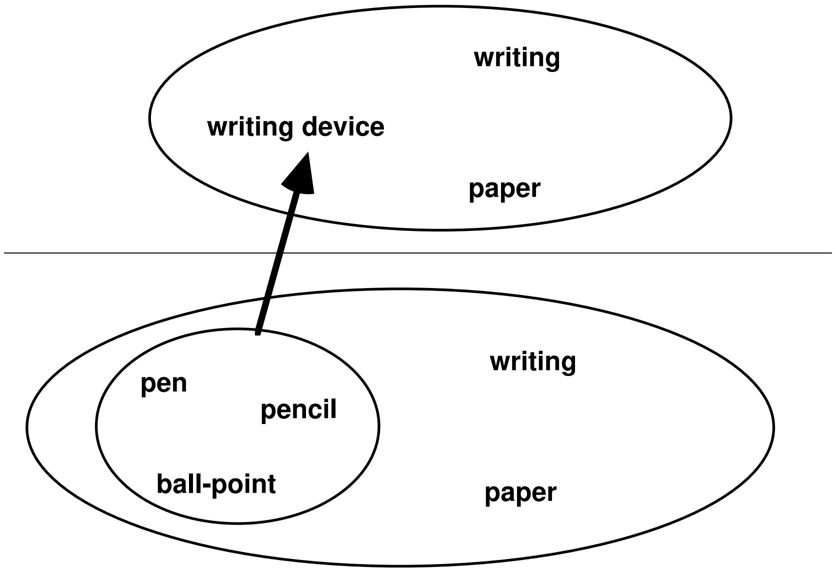

Fig. 3: a condensation of the subcoherence <**pen**, **pencil**, **ball-point**> into the new, more general topic **writing device.**

THOUGHTSTICKER will assist the user in representing her implicit knowledge more explicitly, and in searching for conflicts and proposing resolution. The latter will elicit new knowledge by pointing to gaps and ambiguities in the knowledge that is already there. Moreover, THOUGHTSTICKER can directly suggest possible expansions of the knowledge base through the 'Saturate' operation. This will produce new candidate coherences that would not create structural ambiguities. These suggested combinations of topics may be completely random, or restricted to certain by the user defined ranges of topics. It is up to the user to decide whether the proposed cluster is meaningful or not. Thus, THOUGHTSTICKER constantly interacts or "converses" with the user, helping to construct an ever more complete and well-balanced system of concepts.

4. Bootstrapping in graphs

4.1. ENTAILMENT MESHES AS NON-DIRECTED GRAPHS

Although THOUGHTSTICKER is more flexible than traditional AI knowledge representation systems, its underlying Lp logic has a number of inherent shortcomings. The meaning of a coherence is not very clear and is likely to be interpreted differently by different users. Moreover, Lp's structure is too loose to support "inferences" in the stronger sense of logical deductions showing the truth or falsity of certain expressions. Entailment meshes also do not have any obvious connection with the—admittedly little—knowledge we have about cognition and brain functioning. The most important limitation, however, seems to me the inherent *non-directionality* of its entailment meshes.

Pask's motivation for non-directionality was to replace the traditional hierarchical structures—where higher order concepts are always reduced to (non-grounded) primitives—by a heterarchical ("bootstrapping") structure—where concepts mutually produce each other. Directionality does not imply hierarchy or reducibility, though. When I started working on my structural language (Heylighen, 1990a), I made it intrinsically directional in order to model processes and their "arrow of time", not hierarchical dependencies. The advantage of directed structures is that they encompass non-directed structures as a special case, whereas the converse is not true. In the mathematics of relations, non-directionality corresponds to *symmetry*. A relation R is symmetric if for every pair for which the relation holds, $(\mathbf{a}, \mathbf{b}) \in R$, the inverse pair also satisfies the relation: $(\mathbf{b}, \mathbf{a}) \in R$. General relations are asymmetric, though, which means that the inverse pair can either be there, or not. So, a non-directed structure or graph can always be created by limiting the allowed relations to symmetric ones. On the other hand, there is no clear way to create a directed structure if only symmetric relations are available.

Let me show how such a relational representation generalizes the structure underlying entailment meshes. Two topics **a**, **b** in a coherence are connected by the reflexive and symmetric relation C: "belongs to the same coherence as", which we might also read as "is coherent with". Symmetry means that for all **a** and **b**, $\mathbf{a}\ C\ \mathbf{b} \Rightarrow \mathbf{b}\ C\ \mathbf{a}$. Moreover, within one coherence the relation is also transitive: if $\mathbf{a}\ C\ \mathbf{b}$ and $\mathbf{b}\ C\ \mathbf{c}$, then $\mathbf{a}\ C\ \mathbf{c}$. However, the relation is no longer transitive if we consider topics belonging to different but overlapping coherences. Consider the two coherences **<a, b, c>** and **<c, d, e>**. In that case we have $\mathbf{a}\ C\ \mathbf{c}$, and $\mathbf{c}\ C\ \mathbf{d}$, but not $\mathbf{a}\ C\ \mathrm{d}$. The combination of reflexivity, symmetry and transitivity defines an equivalence relation. The individual coherence clusters could then be viewed as the equivalence classes induced by the relation C. Since C is only locally transitive, its equivalence classes can overlap, instead of partitioning the set of all topics into separate clusters, like a full equivalence relation would.

The representation of an entailment mesh through its collection of coherences is thus equivalent to its representation by the symmetric relation C defined on the set of all topics $T = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, ...\}$: $C \subset T \times T$. The fundamental requirement of the avoidance of structural ambiguity can then be formulated in the following way. Every topic **a** is derived from the other topics it is coherent with: $\{\mathbf{x} \neq \mathbf{a} \mid \mathbf{a}\ C\ \mathrm{x}\} = \mathrm{I}(\mathbf{a})$. Ambiguity is the situation where we have two distinct topics $\mathbf{a} \neq \mathbf{b}$ but such that $\mathrm{I}(\mathbf{a}) = \mathrm{I}(\mathbf{b})$.

4.2. ENTAILMENT NETS AND THE BOOTSTRAPPING AXIOM

Let us generalize the above scheme by considering a general, non-directed, i.e. asymmetric relation. I will call this relation "entailment" and its instances "links", denoting them by right arrows: $\rightarrow$. Instead of "topics", I will call the elements connected by the entailment relation "nodes", or "concepts". The graphical representation of nodes and links will take the form of an unlabelled directed graph, with the only constraint that there can be at most one link between any two nodes (see Fig. 3 for an example). In analogy to the entailment meshes, we may call such graphs *entailment nets*.

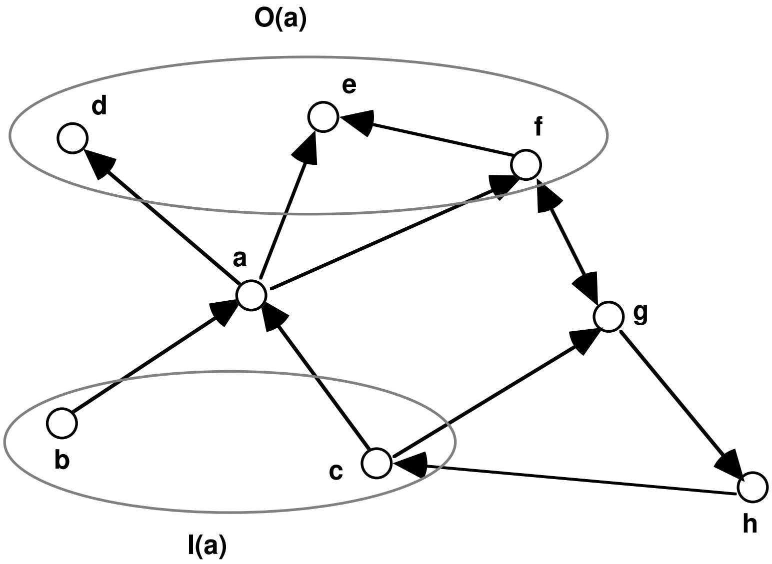

Fig. 3: an entailment net consisting of nodes (**a**, **b**, **c**, ...) connected by links. I(**a**) denotes the set of input nodes of a, O(**a**) the set of its output nodes.

For every node or concept **a**, there will now be *two* relevant sets, depending on the direction in which we follow the relation, i.e. depending on whether we look at the concepts that **a** entails, or that are entailed by it. We can thus define the input and output sets of a concept **a** (see Fig. 3):

Input: $I(\mathbf{a}) = \{ \mathbf{x} \mid \mathbf{x} \rightarrow \mathbf{a} \}$
Output : $O(\mathbf{a}) = \{ \mathbf{x} \mid \mathbf{a} \rightarrow \mathbf{x} \}$

Like in an entailment mesh, the meaning (definition, distinction) of **a** can be interpreted as derived or produced from these two sets. The requirement of ambiguity avoidance can be formulated most generally as the following *bootstrapping axiom* (Heylighen, 1990a,d, 1991b):

two concepts are distinct if and only if their input and output sets are distinct:
$\mathbf{a} \neq \mathbf{b} \Leftrightarrow I(\mathbf{a}) \neq I(\mathbf{b})$ and $O(\mathbf{a}) \neq O(\mathbf{b})$

Thus, the concept **a** is unambiguously defined or determined by the other concepts it is connected with by the entailment relation. This definition is "bootstrapping" because the elements in I(**a**) and I(**b**) are of course themselves only distinguished by virtue of their own connections with distinct elements, including the original **a** and **b**. It is not recursive in the conventional sense, because there is no privileged set of primitive elements in terms of which all others are defined, like in a traditional symbol-based knowledge representation. Note that the axiom implies that concepts with empty input and output sets (i.e. independent, disconnected nodes) cannot be distinguished at all.

Although the bootstrapping axiom is formulated as a static, logical requirement, its practical value lies in the dynamic construction of new concepts and entailments. Suppose that a user is developing a knowledge representation consisting of different concepts together with their entailment relations. A computer program like CONCEPTORGANIZER would then constantly analyse the resulting network looking for ambiguities, i.e. nodes that are different yet have the same input and output sets. If it would find such an ambiguity, it would suggest a number of methods for resolving it, analogous to the methods proposed by THOUGHTSTICKER. These obviously divide in two main strategies: either merge the ambiguous nodes if they are really equivalent, or differentiate their respective input and output sets by adding, splitting or deleting nodes (Heylighen, 1991a).

The THOUGHTSTICKER strategy of merging coherences could be translated in an entailment net by adding a two-way entailment between the ambiguous nodes. For example, if $I(\mathbf{a}) = I(\mathbf{b})$ and $O(\mathbf{a}) = O(\mathbf{b})$ then adding the entailments $\mathbf{b} \to \mathbf{a}$ and $\mathbf{a} \to \mathbf{b}$ would add $\mathbf{a}$ to $I(\mathbf{b})$ and $O(\mathbf{b})$, and add $\mathbf{b}$ to $I(\mathbf{a})$ and $O(\mathbf{b})$, thus distinguishing the respective input and output sets. The two-way entailment could be interpreted as a "similar, yet different" relation. (However, if the entailment would also be reflexive, i.e. include the links $\mathbf{a} \to \mathbf{a}$ and $\mathbf{b} \to \mathbf{b}$, then we would be back to ambiguity since the input and output sets would again be identical.)

The richer structure of entailment nets makes it less likely to encounter ambiguities in an arbitrary net, since *both* input and output sets must be identical to have an ambiguity. However, there are different types of "approximate" ambiguities, which do not strictly transgress the bootstrapping axiom, but where the difference is not very large. For example, the two input sets of a pair of nodes could be identical, while their output sets differ only in one element. In such cases, it is possible that the single difference is accidental, and that the nodes should better be merged. Or, it is likely that there is more than one difference, and therefore the computer system will suggest to the user to add a difference in the input sets which reflects the existing difference in the output sets.

4.3. NODE INTEGRATION

A system based on entailment nets would also allow an equivalent of THOUGHTSTICKER's "condense" operation for creating a more general concept, by integrating a number of more specific concepts. This may be called node integration, as distinguished from the node identification (merging) suggested in the case of ambiguity (cf. Bakker, 1987; Stokman & de Vries, 1988).

A cluster of concepts that are strictly distinct according to the bootstrapping axiom may still be indistinguishable from outside the cluster, because they have the same input and output links with nodes outside the cluster: consider a set of nodes $A = \{\mathbf{a}_i \mid i = 1, ..., n\}$, with the property that for all $1 \leq i, j \leq n$: $I(\mathbf{a}_i) \setminus A = I(\mathbf{a}_j) \setminus A$ and $O(\mathbf{a}_i) \setminus A = O(\mathbf{a}_j) \setminus A$ (where "\" stands for the set theoretic difference). In that case, the nodes belonging to A can be *integrated* and replaced by a single new node A, resulting in a much simplified graph, without affecting the distinctions between nodes outside of A. See Fig. 4 for an example.

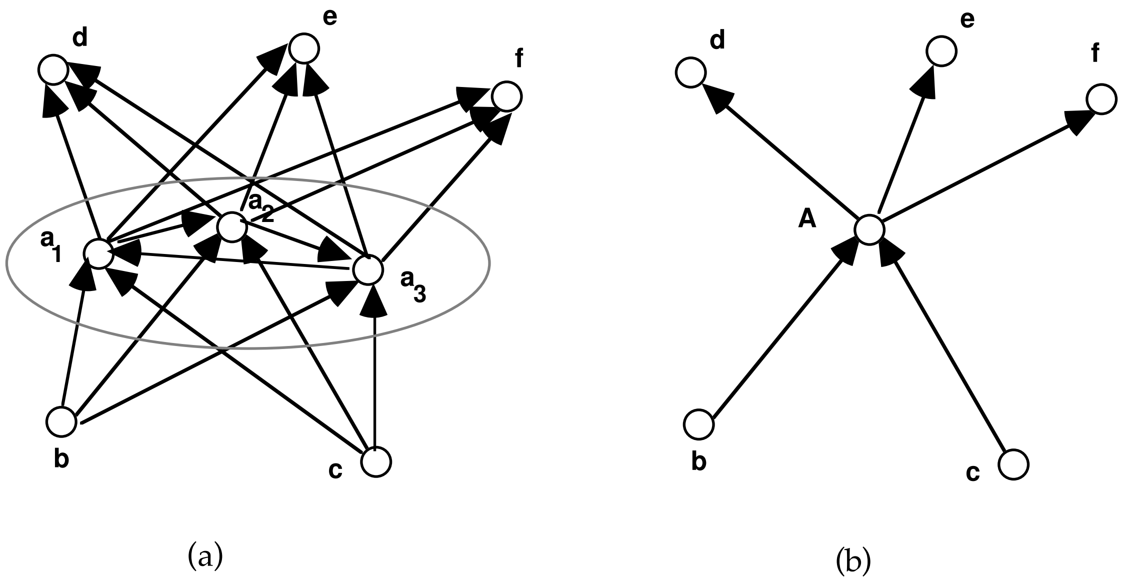

Fig. 4: (a) a cluster {a_1, a_2, a_3} of nodes that are externally indistinguishable; (**b**) the graph resulting from the replacement of that cluster by the single node A.

Again, the property of external indistinguishability of a cluster can be relaxed to *approximate* indistinguishability. This means that the external input and output sets of the cluster elements are not completely identical, but have most elements in common. This would be the case with concepts that have a "family resemblance", with many common properties, but some idiosyncratic differences. Fisher (1987) has proposed an algorithm for clustering together such concepts on the basis of the number of properties —in an entailment net, this would mean input and output elements—they have in common. (The algorithm uses the conditional entropy as a measure of difference between sets of properties). Such conceptual clustering is a basic form of what is usually called "machine learning", "knowledge discovery" or "data mining", i.e. the retrieval of regularities in large sets of interrelated data.

5. Bootstrapping semantic networks

A shortcoming of both Pask's entailment meshes and the entailment nets as I have presented them until now is that there is no clear interpretation of the "entailment" that keeps the concepts together. This very abstract relation tells us that concepts somehow depend on each other, but it does not tell us how.

Another knowledge representation scheme from AI, the *semantic network* (Brachman, 1977; Shastri, 1988; Sowa, 1991), is based on similar nets of interdependent concepts, but here the dependencies are classified into distinct types with specific interpretations. For example, different types of relations might specify that a concept **a** "causes" another concept **b**, that **a** "is a part of" **b**, or that **a** "is a special case of" **b**. The motivation underlying semantic networks is that concepts get their meaning through the semantic relations they have with other concepts. This is similar to the bootstrapping philosophy underlying entailments meshes and entailment nets.

However, semantic networks do not solve the "symbol grounding" problem; they merely push it to another level of description. Although the nodes are supposed to mutually define each other, the relations or link types do not. Like the symbols in a more traditional representation, the link types used in a semantic network are primitives, which are more or less arbitrarily imposed by the system designer. In practice, this has produced confusion, with different researchers using different link types, or interpreting what seem to be the same link types in a different manner. Moreover, since the number of link types is not *a priori* limited, there is a tendency to solve problems by creating a new *ad hoc* link type each time it is not clear how a particular relation can be expressed using the existing link types. Because of this semantic confusion, the empirical verification of semantic network-based theories of cognition has produced ambiguous results, sometimes seeming to confirm the theory, sometimes seeming to contradict it. Yet, a well-chosen set of link types can produce a kind of intuitive recognition, which may help the user to understand knowledge formulated as a semantic network more easily than knowledge expressed in a more abstract or sparse formalism.

For this reason I have tried to reconstruct a semantic network-like structure within my entailment nets (Heylighen, 1991b), while keeping to the fundamental requirement that all meaning or distinction only be justified by bootstrapping *within* the system, not by appealing to external reality as an invisible arbiter. The trick to bootstrap link types is simply to reduce them to node types, which themselves are reduced to nodes. Since all nodes are subject to the bootstrapping axiom, this indirectly subjects link types to bootstrapping as well. Let me explain this procedure in more detail.

The first step is to propose a general interpretation of the existing nodes and links in an entailment net. Until now, I have merely sketched the representation formalism and pointed out its analogies with Pask's Lp expressions, without clarifying what its elements concretely stand for. Since nodes are defined by the way they are distinguished, it is natural to interpret them as basic cognitive *distinctions* (cf. Spencer Brown, 1969; Heylighen, 1990b,d), that is, as classes of phenomena that are separated or distinguished by an observer from all other phenomena that do not belong to the class. It can be argued that all perceptual and cognitive entities, such as concepts, patterns, categories, or experiences, are distinctions. They are all triggered by certain phenomena but not by others. They thus separate the universe of phenomena into two complementary classes: those that fit the concept (the "indication" or "marked" side of the distinction, according to Spencer Brown, 1969), and those that do not. A distinction can be seen as the most fundamental unit of cognition.

Whereas the distinctions cut up the phenomenal universe, the entailment relation connects everything back together. If distinction **a** entails distinction **b**, we might say that given **a** we can somehow expect to see **b** as well. This is a kind of a generalized or weakened form of "if **a**, then **b**". In order to recover more of the properties of the traditional "if...then" in logic (implication), we should at least demand transitivity of the relation. Indeed, "if **a**, then **b**" and "if **b**, then **c**" together imply "if **a**, then **c**". The entailment relation does not have any a priori properties such as transitivity, symmetry or reflexivity. Yet, we can always choose to focus attention on that part of the entailment relation that is transitive, and to interpret it as an "if...then". (One way to mathematically generate the transitive part is to take the intersection of an entailment relation E with its second power: $E_{\text{trans}} = E \cap E^2$).

The interpretation of this "if **a**, then **b**" between distinctions is that the class **a** is somehow subsumed in or followed by the class **b**. For example, *if* a phenomenon is a **dog**, *then* it is also a **mammal**: **dog** → **mammal**. It means that a phenomenon denoted by the first concept cannot be present or actual, without a phenomenon denoted by the second one being (simultaneously) or becoming (afterwards) actual. As the example shows, a primary instantiation of entailment is the relation between an instance or subclass and the more general class to which it belongs. The more general class (e.g. mammal) can be seen as grouping a number of related, more concrete concepts (e.g. dog, cat, mouse, deer, etc.). With such an interpretation we can reinterpret the input of a concept **x**, I(**x**), as its "extension", i.e. the set of its instances, and its output O(**x**) as its "intension", i.e. the conjunction of its defining features. The meaning of **x**, as expressed by the bootstrapping axiom, can then be interpreted as determined by the disjunction of its input elements, and the conjunction of its output elements.

This interpretation suggests yet another heuristic for distinguishing "similar" nodes. Suppose you have two nodes **a** and **b**, such that I(**a**) ⊂ I(**b**), while O(**b**) ⊂ O(**a**). In that case the system might suggest to the user to create a link from **a** to **b**, **a** → **b**, assuming that a concept **a** with a smaller extension and a larger intension than **b** is likely to be a special case or subcategory of **b**.

5.1. ONTOLOGICAL DISTINCTIONS AS BASIC NODE TYPES

If we follow the entailment relation "upward" between classes we will reach ever more abstract or more general distinctions, e.g. **dog** → **carnivore** → **mammal** → **vertebrate** → **animal** → **organism** → **object**. The most abstract classes can be interpreted as the philosopher's "categories", the most basic or universal distinctions which underly our understanding of the universe. These fundamental concepts can be said to define an "ontology", i.e. a theory of the most fundamental categories of existence, such as time, space, matter, truth, cause and effect. These ontological distinctions can be interpreted as basic *node types*, which allow a classification of other, more concrete nodes.

Note that the word "ontology" has recently received a broader and more practical meaning in the domain of knowledge representation, where it denotes the complete system of concepts, with their definitions and relationships, that support a shared conceptualization of a domain (Uschold and Gruninger, 1996). The original philosophical meaning, which I use here, is closer to what Uschold and Gruninger call a "meta-ontology", i.e. a set of allowable types of concepts. What they call "an ontology" would correspond to the whole system of concepts in an entailment network. In that sense, CONCEPTORGANIZER could be viewed as a tool for building ontologies. As an illustration, the Principia Cybernetica Project has developed an ontology of basic systems theoretic concepts in the form of a semantic network (Joslyn, Heylighen & Bollen, 1997).

In order to develop a skeleton (meta-)ontology, or list of fundamental node types, I have posited two basic dimensions of distinction: stability (or time) and generality (Heylighen, 1991b). Stability distinguishes more from less temporally variable phenomena, while generality distinguishes abstract, universal classes from their concrete, particular instances. Although these distinction dimensions can in principle have continuous values, it is simpler to consider discrete classes. Stability can be seen as having three possible values: transitional (i.e. with no distinguished duration), temporary (with a finite duration), and stable (with an indefinite duration), and generality two: particular - general. The combination of these 3 x 2 values leads to 6 basic types of distinction (see table 1).

Time\generality	**General**	**Particular**
Stable	Class	Object
Temporary	Property	Situation
Transitional	Change	Event

Table 1: 6 basic node types, generated as combinations of 3 values for the time dimension, and 2 values for the generality dimension.

For example, an object is a distinction that is stable (it is not supposed to appear or disappear while we are considering it), and particular (it is concrete, there is only of it). A property is a distinction that is general (several phenomena may be denoted by it, it represents a common feature), and temporary (it may appear or disappear, but normally it remains present during a finite time interval). An event is instantaneous (it appears and disappears within one moment), and particular (it does not denote a class of similar phenomena, but a specific instance). Events can be seen as discrete changes, as transitions in which something was present and suddenly is no longer present, or was absent and suddenly appears. Changes can be seen as classes of events or as "transformations" (in the mathematical sense of a function mapping a set of initial states onto a set of subsequent states).

Some examples will further clarify this classification. "Dog" is a *class*, and so is "mammal". My own dog "Fido", on the other hand, is an *object* that belongs to the class of dogs. "Barking" is a *property* that Fido may or may not have at any particular moment. "Being able to bark", on the other hand, is a class that mostly overlaps with the class of "dog". "Fido is barking" is a *situation*, a temporary state of affairs, which in this case involves a particular object. "The 2nd World War" is another situation. "The beginning of the 2nd World War", on the other hand, is an *event*. "Starting a war" is a general *change*.

These ontological distinctions are meant to be an aid to analysis, not an absolute statement about the structure either of the world or of the formal system. They are not at the same level of importance as the bootstrapping axiom. They should rather be seen as a checklist (cf. Heylighen, 1991a) of distinctions that are likely to be relevant. Their use is not mandatory. Indeed, it is easy to conceive of a concept in an entailment network that cannot be readily classified along either of the ontological dimensions.

Consider for example the concept of "the American democracy". Does this concept refer to a particular institution that exists in one country, or to a more general method of governing that could be applied to many instances? Does it represent the situation at this moment in time, or a system that may survive indefinitely? Different people are likely to answer these questions differently in different contexts. Yet the concept of "American democracy" is sufficiently clear that it can be meaningfully used in communicating and representing knowledge. It can therefore be represented in an entailment network, perhaps with entailments linking it to "the American senate" and "the Bill of Rights", but without needing to be linked to one of the ontological concepts of "object", "class", "situation", etc. Still, the creation of more specialised distinctions such as "the American democracy as method of governing" (class) or "the American democracy as an institution during the Reagan years" (situation) may be helpful to elucidate semantic confusions between different (p-) individuals who use the same words but speak about different things.

5.2. BASIC LINK TYPES

With these generic node types we can now produce a number of corresponding link types by considering the possible combinations of nodes between which an "if... then" link can exist. There is one constraint on these combinations, though: a more "invariant" (more stable or more general) distinction can never entail a less invariant one. Otherwise, the second would be present each type the first one is present, contradicting the hypothesis that it is less invariant than the first one. For example, a class cannot entail an object and a situation cannot entail an event. Yet, two concepts with the same type of invariance (e.g. two objects) can be connected by an entailment relation. The remaining possible combinations are summarized in figure 5.

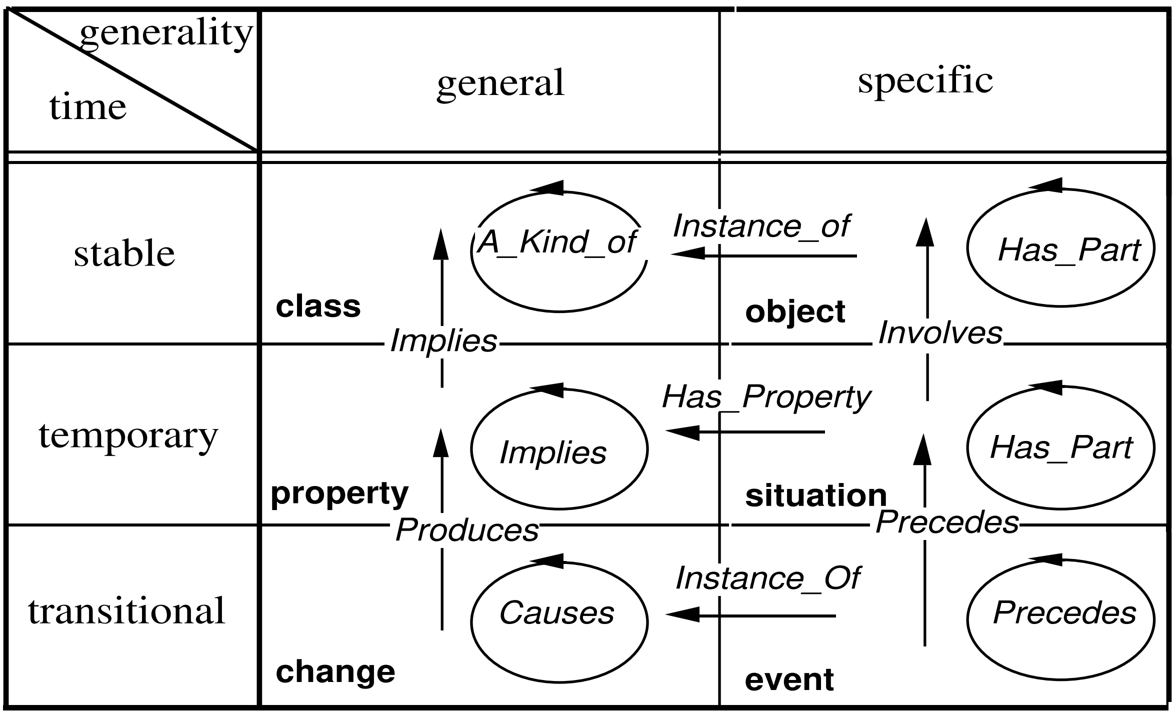

Fig. 5: Link types derived from the allowed combinations between node types; the straight arrows represent entailment from one type to another (more invariant) one, the circular arrows represent entailment from a concept of one type to a concept of the same type

Let us discuss the most important link types in this scheme. When an object **a** entails a class **b**, **a** → **b**, then **a** is an "instance of" **b**. When a class **a** entails another class **b**, then **a** is "a kind of" **b**. The union of the relations "a kind of" and "instance of" may be called "is a". This is the most popular link type used in traditional semantic networks. For example, "a dog *is a* mammal" and "Fido *is a* dog". Note that the traditional "has property" relation cannot be used to link an object with a property. If you wish the express the fact that an object has a property (which because of its temporary character cannot be expressed as a class), e.g. "Fido is tired", then you must create a situation which involves (only) the object "Fido" and has the property "tired".

When an object **a** always entails the presence of another object **b**, then **b** must belong to or be a part of **a**. (If two objects entail each other, they are either identical or so tightly bound that they cannot be separated.) Note that the direction of the "has part" entailment may seem counter-intuitive, since it runs from a "larger" whole to a "smaller" part, whereas the "is a" entailment runs from a "smaller" class to a "larger" superclass, thus putting the conventional hierarchy on its head. The (physical) size of an object has nothing to do with the (logical) size of a class, though. One should beware of the intuitive tendency to see "is a kind of" as analogous to "is a part of", and remember that in terms of entailment it is equivalent to the inverse relation "has as part".

The entailment from property to class is a simple implication from temporary to stable features. E.g. if something falls it has a mass: **falling** → **mass**, yet objects that have mass (permanently) are not permanently falling. Property entailing property is again a simple implication (or succession), now between temporary features, e.g. **falling** → **moving**.

The application of the entailment relation to stable distinctions follows relatively closely the logical implication. However, when applied to distinctions that have an element of time, the "then" part of "if...then" can be interpreted as an indication of temporary succession. This makes it possible to express the flow of time in the formalism. The cognitive interpretation is that the main function of knowledge is anticipation: trying to foresee the future on the basis of the present (Heylighen, 1993; Turchin, 1993). The simplest case is the entailment between events, which can be interpreted as "precedes or is simultaneous with". With this interpretation, my original structural language for the foundations of physics (Heylighen, 1990a) corresponds to that subset of the present representation scheme in which only events and precedence relations are considered. As shown in (Heylighen, 1990a), the network structures present in this subset are sufficient to reconstruct the primitive geometry of relativistic space-time.

The generalization of an event to a class of similar events corresponds to a change. When a change **a** entails another change **b**, then **a** and **b** "covary" and since **a** is prior to **b**, we can interpret **a** as the cause of **b**. For example, "heating water (change) causes (entailment) boiling (change)". There is no direct way to express by a link that an event **a** causes another event **b**, except by noting that **a** precedes **b**, and that **a** and **b** are two instances of general change phenomena which are related by a causal relation. (This is not a shortcoming of the present knowledge representation, but a general property of causation, which can only be established after observing a repeated covariation.) A transitional phenomenon can also entail a more long lasting phenomenon. For example, "boiling (change) produces vapor (property or class)". Similarly, an event (e.g. "the 1939 invasion of Poland") can precede a situation (e.g. "the 2nd World War"). Sometimes this precedence can also be interpreted as a production, if event and ensuing situation can be interpreted as instances of a change producing a property (e.g. "invasions produce war"), but this is in general not the case (e.g. "the release of 'Gone with the Wind' preceded the 2nd World War") (see Fig. 6).

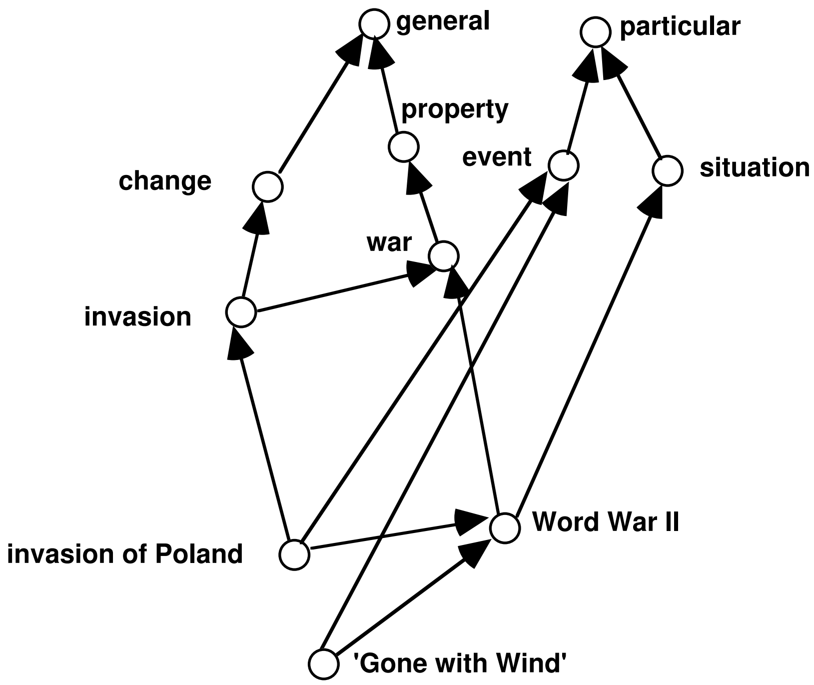

Fig. 6: an entailment network representing the semantic relations "the invasion of Poland produced World War II", and "'Gone with the Wind' preceded World War II".

The fact that a short-lived phenomenon (e.g. an event, change or situation) can entail a more long-lived one (e.g. a property, class or object), but not the other way around, is an illustration of the "principle of asymmetric transitions" governing systems evolution: a transition from an unstable configuration to a stable one is possible, but the converse is not (Heylighen, 1992b). Of course, we can formulate expressions such as "the gun (stable) produced a shot (transitional)", but this is just a shorthand for saying that a particular event (transitional) involving a gun, a finger and a movement of the gun's trigger, produced another event, a shot. This example illustrates how the simple constraints inherent in the present ontology (and in entailment nets in general) force the user to make implicit—but necessary—distinctions explicit, thus avoiding ambiguities and gaps in the knowledge system.

In conclusion, the advantage of this representation scheme is that most of the intuitive and often used semantic categories (objects, classes, causality, whole-part relations, temporal precedence, etc.) can be directly expressed in it, using a simple and uniform format. The resulting representation appears general, consistent and unambiguous. This allows us to reduce a complicated set of semantic categories to an extremely simple and flexible formal structure. The disadvantage is that many more links are needed in order to reduce various link types to nodes than if the links could be simply labeled by their types. However, the burden of keeping track of all the links will normally rest on the computer, and not on the user, who could still work with a higher level representation using typed links.

5.3. FRAMES AND INHERITANCE

Although the scheme above incorporates the most frequently used link types, it can never incorporate all possible types. Any two-place predicate or relation could be made into a link type, for example: "is the father of", "is greater than", "sits to the left of". This freedom will quickly overburden any semantic network scheme. Another AI representation scheme, a "frame", extends the semantic network representation by allowing custom-made relations to be associated with different types of concepts. A frame consists of a central concept related to other concepts by "attribute-value" pairs (also called "slots and fillers"). For example, the concept **car** has attributes (slots) such as **color**, **brand**, **model**, and **year**. A particular instance of that concept, such as my own car, will have particular values (fillers) for these respective attributes, such as **grey**, **Honda**, **Civic** and **1992**. The values of an attribute are in general concepts themselves and thus the attribute plays the role of a typed link connecting nodes in a semantic network.

The unrestricted proliferation of attribute-value pairs is kept in check by the mechanism of *inheritance*: concepts inherit most of their attributes (and sometimes also values) from the concepts higher up in the "is a" hierarchy, that is, from the classes to which they belong as subclasses or as instances. For example, a car is a vehicle, and therefore **car** will inherit a number of attributes such as **number of wheels** or **type of fuel** from the **vehicle** category. **Vehicle** itself will inherit attributes such as **size** and **weight** from the **object** category. This has the advantage that each attribute only needs to be represented once, in the concept situated at the highest level of the hierarchy where it is needed. Thus, the number of attributes or link types is controlled by associating each of them with one or a few high level concepts.

Such a "frame-like" knowledge representation can be easily expressed in an entailment net. Suppose I wish to express the fact that my car, which is an instance of the class **car**, has the value **grey** for its attribute **color**. Fig. 7 shows how this typical attribute-value pair can be represented with untyped entailment links between nodes typed through the categorical distinctions introduced earlier. The "is a" link between **my car** and **car** is obviously an entailment between an object and a class (see Fig. 5). The statement that every car has an attribute **color** is equivalent to the statement that the class **car** is a subclass of, and therefore entails, the class of colored things. Thus, the entailment link goes from **car** to **color**. Similarly my car, which is an object, belongs to the class of grey things which is itself a subclass of the class of colored things. Thus, the scheme linking an instance to its class and to an attribute-value pair can be reduced in essence to a distinction graph with two times two parallel arrows, in a configuration resembling what is called a "commutative diagram" in mathematics.

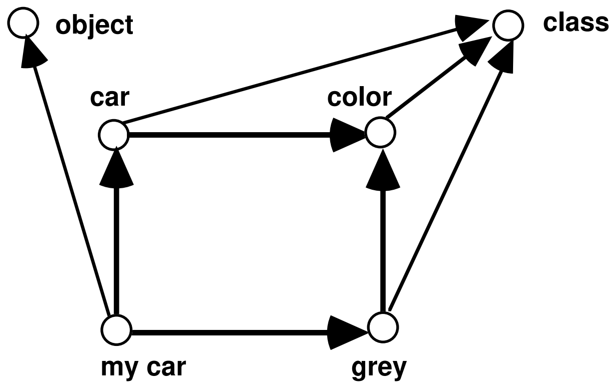

Fig. 7: representation of a frame (car) with attribute (color) and value (grey) for an instance (my car), as a commutative diagram (thick arrows) in an entailment network.

Since "if...then" entailments are supposed to be transitive, inheritance of attributes comes automatically. Whether **my car** gets its attribute **number of wheels** through the class **car** or through the higher class **vehicle** should not make a difference. In both cases, a commutative diagram can be drawn, with an entailment path running from **my car** to **number of wheels**, either through **car** or through **vehicle**. The inheritance of values in a frame has a more subtle feature, though, the *default mechanism*: a concept inherits its values from the class to which it belongs, *unless* there are values that are directly attached to the concept. The values filling the slots at the level of the concept override those filling the same slots at the higher levels from which it normally inherits. Inheritance of values only takes place if the values are not specified at the level of the concept itself.

The traditional example of inheritance by default is the concept of "penguin", which belongs to the class of birds, from which it inherits properties such as being warm-blooded, and laying eggs. However, by default the class of birds also has the property of being able to fly, which the penguin does not have. This is represented by having a specific attribute-value pair (**can_fly?, false**) attached to the concept **penguin**), while other subclasses of the bird class, such as robins or sparrows simply inherit their attribute value pair (**can_fly?, true**) from the parent class. This can be represented in an entailment net by the following simple graph (Fig. 8):

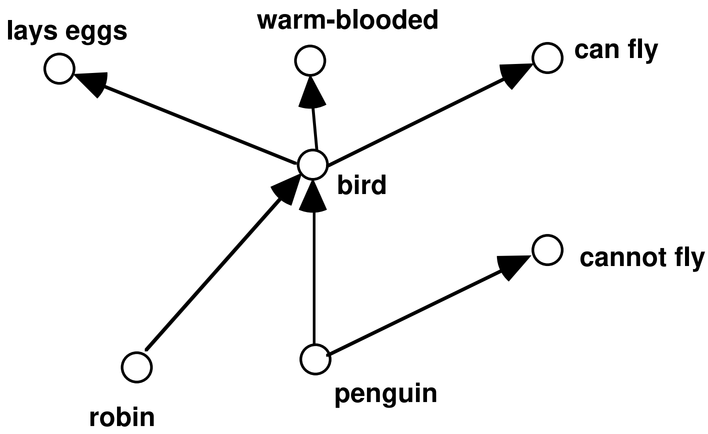

Fig. 8: representation by means of entailment links of a default inheritance (birds can fly) being overwritten for the exceptional case of "penguin" .

In order to interpret this entailment network correctly, the requirement of transitivity for the entailment relation must be relaxed, so that the direct entailment **penguin** → **cannot fly** can override the indirect path **penguin** → **bird**, **bird** → **can fly**. The general rule could be that if two entailment paths from the same node lead to contradictory results, the shortest path is to be preferred.

This somewhat *ad hoc* interpretation may be justified if we consider a more general interpretation of an entailment **a** → **b**, where it expresses a kind of "expectation" of **b**, given **a**, rather than "certitude" of **b**, given **a**. Expectations can come with different "strengths" or degrees of certainty. These may be expressed as a kind of conditional probability of **b**, given **a**: P(**b** | **a**). If we assume that typical entailment links have a strength of 1 or slightly less, paths of entailments will have a strength that is the product of the strengths of the individual links. This will be lower than or equal to the strength of the weakest link. Therefore, if the link from **penguin** to **bird** has maximal strength, P(**bird** | **penguin**) = 1, while P(**can fly** | **bird**) = 0.9, the combined path P(**can fly** | **penguin**) = 0.9, which is overriden by the stronger connection P (**cannot fly** | **penguin**) = 1. We will here not go into further detail about such calculations, but discuss an application of entailment nets which uses link strengths in a more general way.

6. Learning Webs

6.1. ASSOCIATIVE NETWORKS

When we consider links with strengths continuously varying between 0 and 1 (or even -1 and 1 if we want to express negative or inhibitory links), the interpretation must shift from a generalized "if...then" relation, to the weaker "is associated with" connection. This brings us from "semantic" to "associative" networks. Associative networks are in principle more general and more flexible, allowing the expression of different "fuzzy", "intuitive" or even "subconscious" relations between concepts. Such networks have been regularly suggested as models of how the brain works. They are similar to the presently popular "neural" networks, except that the latter are typically used as directed, information processing systems, which are given a certain pattern of stimuli as input and are supposed to produce the correct response to that pattern as output. In the present "bootstrapping" perspective, there is no overall direction or sequence leading from inputs to outputs; there are only nodes linked to each other by associations, in such a way that they are coherent with each other and with the user's understanding of the knowledge domain.

In principle, associative networks could be created by the same type of knowledge elicitation techniques underlying THOUGHTSTICKER or CONCEPTORGANIZER, where a user enters a number of concepts and links and is prompted by the system to add further links and concepts under the main constraint of avoiding ambiguity. These links must then be attributed some variable degree of strength. However, the very weak requirement of "associativity" allows virtually any pair of concepts to be linked, if only with a very small link strength. Moreover, there is no obvious generalization of the bootstrapping axiom, which is based on discrete linking patterns, to continously varying linkages. If everything is linked to everything, then all nodes become uniformly indistinguishable. Finally, it is in practice impossible to let users realistically estimate a strength value for each of the huge number of possible links. Rocha (1991) has suggested a method to "fuzzify" conversation theory, by calculating continously varying conceptual distances between nodes in an entailment mesh, on the basis of the number of linked nodes they share, but this approach has never been fully worked out.

In psychology, rudimentary associative networks have been created through experiments in which subjects were given a word (say "cat"), and were asked which other word first came to mind (e.g. "dog", "mouse", or "milk"). The more often a certain word **b** is given in response to the cue word **a**, the stronger the association from **a** to **b**. Since this approach usually only finds a small number of associations for any given word, association strengths for links between other words are calculated by taking into account indirect associations (e.g. knowing the strengths of "dog → cat" and "cat → mouse" would allow one to calculate the strength of "dog → mouse"). Note that such associations are in general asymmetric. For example, when cued with "penguin" the probability that you would say "bird" is not so small, whereas the probability to respond with "penguin", when cued with "bird" is virtually zero. This methodology, however, requires a lot of work from designers and users, and is only useful for simple, well-known items like common words.

6.2. ALGORITHMS FOR LEARNING HYPERTEXTS

My collaborator Johan Bollen and I have developed a more efficient way to create associative networks, by applying a bootstrapping philosophy to hypertext navigation (Bollen & Heylighen, 1996, 1997). The hypertext paradigm provides a very simple and natural interface for representing any type of knowledge network to a user (Heylighen, 1991a). A document in a hypertext may contain the description in natural language or other media of a concept, while the associations of that concept to other concepts are represented by hypertext links within the document's text or graphics. The selection of a link by the user calls up the linked document, thus allowing the user to “browse” or “navigate” through the network. The problem with present hypertexts is the same one as with semantic networks and other knowledge representations: the system of nodes and links is created in an *ad hoc* way by the system designer, without being justified by either an underlying theory, or an empirical method for testing the effectiveness of the resulting network. This typically results in labyrinthine, "spaghetti-like" meshes of interconnected data, so that the user quickly gets lost in hyperspace.

The method we developed allows an associative hypertext network to “self-organize” into a simpler, more meaningful, and more easily usable network. The term “self-organization” is appropriate to the degree that there is no external programmer or designer deciding which node to link to which other node: better linking patterns emerge spontaneously. The existing links bootstrap new links into existence, which in turn change the existing link patterns. The information used to create new links is not internal to the network, though: it derives from the collective actions of the different users. In that sense one might say that the network “learns” from the way it is used.

The algorithms for such a learning web are very simple. Every potential link is assigned a certain “strength”. For a given node **a**, only the links with the highest strength are actualized, i.e. are visible to the user. Within the node, these links are ordered by strength, so that the user will encounter the strongest link first. There are three separate learning rules for adapting the strengths.

1) Each time an existing link, say $\mathbf{a} \rightarrow \mathbf{b}$, is chosen by the user, its strength is increased. Thus, the strength of a link becomes a reflection of the *frequency* with which it is used by hypertext navigators. This rather obvious rule can only consolidate links that are already available within the node. In that sense, it functions as a *selector* of strong connections. However, it cannot actualize new links, since these are not accessible to the user. Therefore we need complementary rules that generate novelty or *variation*.

2) A user might follow an indirect connection between two nodes, say $\mathbf{a} \rightarrow \mathbf{b}$, $\mathbf{b} \rightarrow \mathbf{c}$. In that case the potential link $\mathbf{a} \rightarrow \mathbf{c}$ increases its strength. This is a weak form of *transitivity*. It opens up an unlimited realm of new links. Indeed, one or several increases in strength of $\mathbf{a} \rightarrow \mathbf{c}$ may be sufficient to make the potential link actual. The user can now directly select $\mathbf{a} \rightarrow \mathbf{c}$, and from there perhaps $\mathbf{c} \rightarrow \mathbf{d}$. This increases the strength of the potential link $\mathbf{a} \rightarrow \mathbf{d}$, which may in turn become actual, providing a starting point for an eventual further link $\mathbf{a} \rightarrow \mathbf{e}$, and so on. Eventually, an indefinitely extended path may thus be replaced by a single link $\mathbf{a} \rightarrow \mathbf{z}$. Of course, this assumes that a sufficient number of users effectively follow that path. Otherwise it will not be able to overcome the competition from paths chosen by other users, which will also increase their strengths. The underlying principle is that the paths that are most popular, i.e. followed most often, will eventually be replaced by direct links, thus minimizing the average number of links a user must follow in order to reach his or her preferred destination.

3) A similar rule can be used to implement a weak form of *symmetry*. When a user chooses a link $\mathbf{a} \rightarrow \mathbf{b}$, implying that there exists some association between the nodes **a** and **b**, we may assume that this also implies some association between **b** and **a**. Therefore, the reverse link $\mathbf{b} \rightarrow \mathbf{a}$ gets a strength increase. This symmetry rule on its own is much more limited than transitivity, since it can only actualize a single new link for each existing link.

However, the collective effect of symmetry and transitivity is much more powerful than that of any single rule. For example, consider two links $\mathbf{a}_1 \rightarrow \mathbf{b}$, $\mathbf{a}_2 \rightarrow \mathbf{b}$. The fact that $\mathbf{a}_1$ and $\mathbf{a}_2$ point to the same node seems to indicate that $\mathbf{a}_1$ and $\mathbf{a}_2$ have something in common, i.e. are related in some way. However, none of the rules will directly generate a link between $\mathbf{a}_1$ and $\mathbf{a}_2$. Yet, the repeated selection of the link $\mathbf{a}_2 \rightarrow \mathbf{b}$ may actualize the link $\mathbf{b} \rightarrow \mathbf{a}_2$ by symmetry. The repeated selection of the already existing link $\mathbf{a}_1 \rightarrow \mathbf{b}$ followed by this new link can then actualize the link $\mathbf{a}_1 \rightarrow \mathbf{a}_2$ through transitivity. Similar scenarios can be conceived for different orientations or different combinations of the links.

A remaining issue is the relative importance of the three above rules. In other words, how large should the increase in strength be for each of the rules? If we choose unity (1) to be the bonus given by the first rule, there are two remaining parameters or degrees of freedom: t is the bonus for transitivity, s for symmetry. Since the direct selection of a link by a user seems a more reliable indication of its usefulness than an indirect selection, we assume $t < 1$, $s < 1$. The actual values will determine the efficiency of the learning process, but it seems that this matter cannot be settled by pure theoretical reasoning.

6.3. EXPERIMENTAL RESULTS

In order to test these ideas in practice we have set up two experiments. We built a network consisting of 150 nodes, corresponding to the 150 most frequent nouns of the English language. Every node was assigned 10 links to other nodes. These links were randomly selected from the 149 remaining nodes to initialize the web, but would then evolve according to the above learning rules (with $t = 0.5$ and $s = 0.3$). We made the web available on the Internet, and invited volunteers to browse through it, selecting those links from a given node which seemed somehow most related to it. For example, if the start node represented the noun “dog”, a user would choose a link to an associated word, such as “cat”, “animal”, or “fur”, but not to a totally unrelated word, such as “mathematics”. Of course, in the beginning of the experiment, there would be very few good associations available in the lists of 10 random words, and users might have to be satisfied with a rather weak association, such as “meat”. However, when reaching the node “meat”, they might be able to select there another association, such as “carnivore”. Through transitivity, a new link to “carnivore” might then appear in the node “dog”, displacing the weakest link in the list, while providing a much better association than the previously best one, “meat”.

Although there were some unexpected side effects, such as the 'attractor effect', the development of the associative network was surprisingly quick and efficient (Bollen & Heylighen, 1996). After only 2500 link selections (out of 22500 potential links) both experimental networks had achieved a fairly well-organised structure in which most nodes had been connected to large clusters of related words. This may be illustrated by a typical example of how connections are gradually introduced and rewarded until their strength reaches an equilibrium value (Table 2). The position of these associated words shifted upwards in the list until they reached a position that best seemed to reflect their relative strength.

KNOWLEDGE			
0	200	800	4000
trade	education	education	education

view	experience	experience	experience
health	example	development	research
theory	theory	theory	development
face	training	research	mind
book	development	example	life
line	history	life	theory
world	view	training	training
side	situation	order	thought
government	work	effect	interest

Table 1: self-organization of the list of 10 strongest links from the word "knowledge", in different stages: initial random linking pattern, after 200 steps, after 800 steps, and after 4000 steps. A step corresponds to a user selecting a link on one of the 150 nodes, in a web that evolves according to the direct, transitive and symmetric learning rules.

The net result of the experiment was a 150 x 150 matrix of association strengths, which reflected fairly well the most important intuitive associations existing among the concepts. For example, a cluster analysis performed on the matrix produced 9 general categories (Time, Space, Movement, Control, Cognition, Intimacy, Vitality, Society, Office), grouping most of the related words in a single class. Such statistical clustering techniques may provide the equivalent for associative webs of the discrete conceptual clustering we discussed earlier.

We are now trying to determine to what degree these results from our learning web correlate with different word associations derived by other means (e.g. free association experiments, or letting people judge the degree of synonymity). We also plan to test the usefulness of the self-organization, by checking in how far users find knowledge more effectively in a self-organized network, as compared to a network that did not undergo learning. This can be done by measuring the average number of steps needed to find a node, and the average time needed to choose a link. We are further considering additional learning rules, such as *similarity* (nodes sharing several links would get stronger cross-connections), that may make the learning more effective. Although this research is still in its initial stage, and will need much empirical testing to confirm its usefulness, it seems like a very promising approach to quickly and easily develop complex associative networks that are more adequate than hypertexts built manually.

7. Towards an Intelligent Web

Apart from simple knowledge elicitation, the most obvious practical application of the various network models for knowledge structuring which we have reviewed lies in the World-Wide Web, the hypermedia interface to the information stored on the global Internet computer network. As said, the hypertext organization maps directly onto the directed graphs used in entailment nets or semantic networks. The web is a perfect example of a distributed and collectively constructed system of knowledge. The wealth of information provided by the millions of web documents, however, is offset by the almost total lack of structure in the way these documents are linked. This makes it far from trivial to find the precise information one is looking for, even when that information is within easy reach.

The structuring algorithms which we discussed could be used to make the web simpler, more efficient, less redundant, and more complete, by pointing out lacking or superfluous links and nodes, and by suggesting better connection patterns. In particular the associative learning algorithms we sketched could be applied rather straightforwardly to the web as it now exists (Bollen & Heylighen, 1996; Heylighen & Bollen, 1996), resulting in a dramatic reduction in the average number of links a user needs to traverse in order to find a specific item. The proposed node merging and conceptual clustering techniques may prove useful in creating automatic indexes or review documents, grouping links to similar documents. The semantic network organization should also be able to simplify web browsing by providing a simple and unified set of link types, so that users can be more selective in which links they need to explore. For example, if you are interested in the general class to which a concept belongs, it would be meaningless to spend time exploring its instances.

While such a bootstrapping structuration of the web would help a user navigate through hyperspace, entailment nets may support information retrieval in an even more direct way. The directionality of entailment nets allows non-trivial inferences. Such inferences can be used to answer queries. Queries can be formulated either in a well-structured, formal or in an associative, fuzzy way (cf. Heylighen, 1991a), depending on how clear the problem is for the user. Formal queries are aimed at determining the presence of a particular semantic or entailment relation between two or more concepts. For example, "Can a penguin fly?", is a "yes-no" query that needs to be resolved by determining the presence or absence of an entailment from **penguin** to **can fly**. An open-ended query like "Which birds cannot fly?" should produce a list of all concepts that entail both **bird** and **cannot fly**. Associative queries, on the other hand, do not ask for the presence of specific relations, but for the concepts that are in the most general way associatively related with the concepts the user has in mind, e.g. **bird**, **ice**, **fish**, **cute**.

Both types of queries can be tackled in a knowledge network by the mechanism of *spreading activation* (Jones, 1986; Salton & Buckley, 1988; Chen & Ng, 1995): nodes or concepts that are linked to the concepts in the query are "activated". The activation spreads from those nodes through their links to neighbouring nodes, and the nodes which have received the highest activation are brought forward as candidate answers to the query. If none of the proposals are acceptable, those that seem closest to the answer are again activated and used as sources for a new process of spreading. This process is repeated, with the activation moving in parallel from node to node via their links, until a satisfactory solution is found.

In the case of associative queries, the only constraint on spreading activation is the strength of the intervening links: the activation passed through a link is proportional to its strength. The activation arriving in a node is the weighted sum of the activations arriving through all input links of that node. If the associative network is represented by a matrix of link strengths, spreading activation can be implemented by repeatedly multiplying an input vector representing the initial activation for all nodes in the network (typically with values of 1 for the query terms and 0 for all the others) with the matrix, and then selecting the nodes with the highest activation values from the different output vectors. (Chen & Ng, 1995, explore the usefulness of some other spreading activation algorithms for concept retrieval).

We have implemented such a spreading activation program that uses the associative data resulting from our learning web experiment (Bollen & Heylighen, 1996b). The program often manages to mimic the "intuitive" reactions of a human subject trying to guess a word from various clues. For example, the input of the clue words "control" and "society" produces the word "government" as most highly activated, while the words "building", "work" and "paper" produce "office". This is similar to the way thoughts diffuse in the brain, moving along intuitive, fuzzy pathways, rather than retrieving exact matches like traditional search engines. Such "inferences" could obviously never have been achieved through logical deductions, since there is no way in which "office" could have been defined by a boolean combination of the query terms above.

In the case of formal queries, spreading activation will not be a continuously diffusing intensity, but a discrete state of activation which can only follow certain paths. For example, a query looking for birds that cannot fly should not follow links of the "has part" or "causes" type, but only of the "is a" type. The typical implementation in a semantic network will activate the concepts in the query and let the activation follow all possible links of the right type until the activated paths intersect. This is quite inefficient for proving *negative* connections, though, since all possible paths need to be explored between **penguin** and **can fly** in order to show that none exist. For such situations, it seems better to allow negative entailments or "inhibitory" links, of the type "if **penguin**, then not (**can fly**)".

It seems that the introduction of these different types of flexible inference and discovery mechanisms could turn the rapidly developing World-Wide Web from a huge, static repository into an active processor and creator of knowledge (Heylighen & Bollen, 1996). The best metaphor for capturing the collective intelligence formed by millions of users interacting with such a self-organizing, "thinking" web may be the one of a global brain (Russell, 1995). Although many issues still need to be resolved, work is starting in different quarters to turn this science-fiction-like vision into a concrete reality (Goertzel & Pritchard, 1997; Mayer-Kress & Barczys, 1995; Heylighen, 1997b).

Such an implementation at the planetary scale was probably not envisaged by Gordon Pask. Yet, his conversational systems and the present vision of an intelligent web share their view of knowledge as a collective construction striving to achieve coherence, rather than a mapping of external objects. By abandoning the correspondence epistemology and its reliance on fixed primitives, bootstrapping approaches open the way to a truly flexible, adaptive and creative knowledge system. Of course, the systems sketched here are still in their infancy, and need to be thoroughly tested under diverse circumstances, and implemented on a sufficiently large scale to show their practical usefulness. This will obviously require a very large effort. I hope that the work of Gordon Pask, myself and our colleagues will provide sufficient inspiration for other researchers to take up these challenges.

Acknowledgments:

I have been supported during this research by the Fund for Scientific Research-Flanders (FWO), as a Senior Research Associate. I wish to thank Johan Bollen, Cliff Joslyn and others for collaboration and discussions on many of the ideas reviewed in this paper.

References

BAKKER R.R. (1987). *Knowledge Graphs: representation and structuring of scientific knowledge,* (Ph.D. Thesis, Dep. of Applied Mathematics, University of Twente, Netherlands).

BICKHARD M. & TERVEEN L. (1995). Foundational Issues in Artificial Intelligence and Cognitive Science, (Elsevier,).

BOLLEN J. & HEYLIGHEN F. (1996) “Algorithms for the Self-organisation of Distributed, Multi-user Networks. Possible application for the future World Wide Web”, in: *Cybernetics and Systems '96* R. Trappl (ed.), (Austrian Society for Cybernetics), pp. 911-916.

BOLLEN J. & HEYLIGHEN F. (1996b). Finding words through spreading activation [http://pespmc1.vub.ac.be/SPREADACT.html].

BOLLEN J. & HEYLIGHEN F. (1997). "Dynamic and adaptive structuring of the World Wide Web based on user navigation patterns". Proceedings of the Flexible Hypertext Workshop (ACM) (in press).

BRACHMAN, R. J. (1977). "What's in a Concept: Structural Foundations for Semantic Nets", Int. J. Man-Machine Studies 9, 127-152.

CHEN H. & NG T. (1995). "An Algorithmic Approach to Concept Exploration in a Large Knowledge Network (Automatic Thesaurus Consultation): Symbolic Branch-and-bound Search vs. Connectionist Hopfield Net Activation", Journal of the American Society for Information Science 46: 5, 348-369.

CZIKO G. (1995). Without Miracles. Universal Selection Theory and the Second Darwinian Revolution, (MIT Press, Cambridge, MA).

FISHER D.H. (1987). "Knowledge Acquisition via Incremental Conceptual Clustering", Machine Learning 2(7), 139-172.

FORD K.M. & HAYES P.J. (Eds.) (1991). Reasoning Agents in a Dynamic World: The Frame Problem , (JAI Press, Greenwich CT).

GOERTZEL B. & PRITCHARD J. (1997) "About webMind", comoc "white paper" #1 [http://www.comoc.com/About-Webmind.html].

HARNAD S. (1990). "The Symbol Grounding Problem", Physica D 42, 335-346.

HEYLIGHEN F. & BOLLEN J. (1996) “The World-Wide Web as a Super-Brain: from metaphor to model”, in: *Cybernetics and Systems '96* R. Trappl (ed.), (Austrian Society for Cybernetics).p. 917-922.

HEYLIGHEN F. (1987). "Towards a General Framework for Modelling Representation Changes", in: *Proc. 11th Int. Congress on Cybernetics*, (Assoc. Internat. de Cybernétique, Namur), pp. 136-141.

HEYLIGHEN F. (1989). "A Support System for Structuring Complex Problem Domains", in: *Proc. 12th Int. Congress on Cybernetics* (Assoc. Intern. de Cybernétique, Namur), pp. 1025-1032.

HEYLIGHEN F. (1990a). "A Structural Language for the Foundations of Physics", *International Journal of General Systems* 18, 93-112.

HEYLIGHEN F. (1990b). Representation and Change. An Integrative Metarepresentational Framework for the Foundations of Physical and Cognitive Science, (Communication & Cognition, Gent).

HEYLIGHEN F. (1990c). "Classical and Non-classical Representations in Physics I", *Cybernetics and Systems* 21, 423-444.

HEYLIGHEN F. (1990d). "Relational Closure: a mathematical concept for distinction-making and complexity analysis", in: *Cybernetics and Systems '90*, R. Trappl (ed.), (World Science, Singapore), pp. 335-342.

HEYLIGHEN F. (1991a). "Design of a Hypermedia Interface Translating between Associative and Formal Representations", *International Journal of Man-Machine Studies* 35, 491-515.

HEYLIGHEN F. (1991b). "Structuring Knowledge in a Network of Concepts", in: *Workbook of the 1st Principia Cybernetica Workshop*, Heylighen F. (ed.) (Principia Cybernetica, Brussels), pp. 52-58. [http://pespmc1.vub.ac.be/Papers/StructuringKnowledge.html]

HEYLIGHEN F. (1992a). "From Complementarity to Bootstrapping of Distinctions: A Reply to Löfgren's Comments on my Proposed 'Structural Language", *International Journal of General Systems* 21(1), 99.

HEYLIGHEN F. (1992b). "Principles of Systems and Cybernetics: an evolutionary perspective", in: *Cybernetics and Systems '92*, R. Trappl (ed.), (World Science, Singapore), pp. 3-10.

HEYLIGHEN F. (1993). "Selection Criteria for the Evolution of Knowledge", in: *Proc. 13th Int. Congress on Cybernetics* (Association Internat. de Cybernétique, Namur), pp. 524-528.

HEYLIGHEN F. (1997). "Objective, subjective and intersubjective selectors of knowledge", *Evolution and Cognition* [in press]

HEYLIGHEN F., ROSSEEL E. & DEMEYERE F. (eds.) (1990). Self-Steering and Cognition in Complex Systems. Toward a New Cybernetics, (Gordon and Breach Science Publishers, New York).

HEYLIGHEN F., JOSLYN C. & TURCHIN V. (1997). Principia Cybernetica Web [http://pespmc1.vub.ac.be/].

JONES W. P. (1986). "On the Applied Use of Human Memory Models", *International Journal of Man-Machine Studies* 25, 191-228.

JOSLYN C., HEYLIGHEN F. & BOLLEN J. (1997). Systems Concepts [http://pespmc1.vub.ac.be/SYSCONC.html].

JOSLYN C., HEYLIGHEN F. & TURCHIN V. (1993). "Synopsys of the Principia Cybernetica Project", in: *Proc. 13th Int. Congress on Cybernetics* (Association Internationale de Cybernétique, Namur), pp. 509-513.

LÖFGREN, L. (1991). "Complementarity in Language: Toward a General Understanding." In Carvallo, ed. *Nature, Cognition and System II: Complementarity and Beyond*. Dordrecht, Boston, London: Kluwer, pp. 73-104 .

MATURANA H. R., & VARELA F. J. (1992). The Tree of Knowledge: The Biological Roots of Understanding, (rev. ed.), (Shambhala, Boston).

MAYER-KRESS G. & C. BARCZYS (1995). "The Global Brain as an Emergent Structure from the Worldwide Computing Network", *The Information Society* 11 (1), 1-28.

PASK G. & GREGORY D. (1986). "Conversational Systems", in: Human Productivity Enhancement (vol. II), Zeidner R. J. (ed.), (Praeger), pp. 204-235

PASK G. (1975). Cognition, Conversation and Learning, (Elsevier, Amsterdam).

PASK G. (1976). Conversation theory: applications in education and epistemology, Elsevier : Amsterdam.

PASK G. (1980). "Developments in Conversation Theory—Part 1", Int. J. Man-Machine Studies 13, 357-411.

PASK G. (1984). "Review of Conversation Theory and a Protologic (or Protolanguage) Lp", *Educational Communications and Technology* 32, no. 1, pp. 3-40.

PASK G. (1990). "Some Formal Aspects of Conversation Theory and Lp", in: Heylighen F., Rosseel E. & Demeyere F. (eds.). *Self-Steering and Cognition in Complex Systems. Toward a New Cybernetics,* (Gordon and Breach Science Publishers, New York), pp. 240-247.

PASK G. (1991). "The Foundations of Conversation Theory, Interaction or Actors Theory, all Cybernetic and Philosophically so", in: Heylighen F. (ed.), *Workbook of the 1st Principia Cybernetica Workshop*, (Principia Cybernetica, Brussels), pp. 15-18. [http://pespmc1.vub.ac.be/Papers/Workbook.html]

PIAGET J. (1937). La Construction du Réel chez l'Enfant, (Delachaux & Niestlé, Neuchâtel).

POPPER K. (1959). The Logic of Scientific Discovery, (Hutchinson, London).

RESCHER N. (1973) Coherence Theory of Truth, London, Oxford University Press.

ROCHA L. (1991). "Fuzzification of Conversation Theory", (presented at the 1st Principia Cybernetica Workshop, Free University of Brussels).

RUSSELL, P. (1995). *The Global Brain Awakens: Our Next Evolutionary Leap* (Miles River Press).

SALTON G. & BUCKLEY C. (1988). “On the Use of Spreading Activation Methods in Automatic Information Retrieval”, *Proc. 11th Ann. Int. ACM SIGIR Conf. on R&D in Information Retrieval* (ACM), pp. 147-160.

SHASTRI L. (1988). Semantic Networks (Morgan Kauffman, Los Angeles).

SOWA J.F. (ed.) (1991). Principles of Semantic Networks: explorations in the representation of knowledge (Morgan Kauffman, Los Angeles)

SPENCER BROWN G. (1969). Laws of Form, (Allen & Unwin, London).

STOKMAN F.N. & DE VRIES P.H. (1988). "Structuring Knowledge in a Graph", in: *Human-Computer Interaction, Psychonomic Aspects*, G.C. van der Veer & G.J. Mulder (eds.), (Springer, Berlin).

THAGARD, P. (1989) Explanatory Coherence. Behavioral and Brain Sciences 12: 435-467.

TURCHIN V. (1993). "On Cybernetic Epistemology", Systems Research 10:1, 3-28.

VAN BRAKEL, J. (1992) The Complete Description of the Frame Problem. PSYCOLOQUY 3(60) frame-problem.2

VON FOERSTER H. (1996) Cybernetics of Cybernetics (2nd edition). (Future Systems, Minneapolis)

VON GLASERSFELD E. (1984). "An Introduction to Radical Constructivism", in: The Invented Reality, P. Watzlawick (ed.), (Norton, New York).