Collective Intelligence and its Implementation on the Web: algorithms to develop a collective mental map

Francis HEYLIGHEN*

Center "Leo Apostel", Free University of Brussels

Address: Krijgskundestraat 33, B-1160 Brussels, Belgium
E-mail: fheyligh@vub.ac.be
home page: http://pespmc1.vub.ac.be/HEYL.html

ABSTRACT.

Collective intelligence is defined as the ability of a group to solve more problems than its individual members. It is argued that the obstacles created by individual cognitive limits and the difficulty of coordination can be overcome by using a collective mental map (CMM). A CMM is defined as an external memory with shared read/write access, that represents problem states, actions and preferences for actions. It can be formalized as a weighted, directed graph. The creation of a network of pheromone trails by ant colonies points us to some basic mechanisms of CMM development: averaging of individual preferences, amplification of weak links by positive feedback, and integration of specialised subnetworks through division of labor. Similar mechanisms can be used to transform the World-Wide Web into a CMM, by supplementing it with weighted links. Two types of algorithms are explored: 1) the co-occurrence of links in web pages or user selections can be used to compute a matrix of link strengths, thus generalizing the technique of "collaborative filtering"; 2) learning web rules extract information from a user's sequential path through the web in order to change link strengths and create new links. The resulting weighted web can be used to facilitate problem-solving by suggesting related links to the user, or, more powerfully, by supporting a software agent that discovers relevant documents through spreading activation.

1. Introduction

With the growing interest in complex adaptive systems, artificial life, swarms and simulated societies, the concept of "collective intelligence" is coming more and more to the fore. The basic idea is that a group of individuals (e.g. people, insects, robots, or software agents) can be smart in a way that none of its members is. Complex, apparently intelligent behavior may emerge from the synergy created by simple interactions between individuals that follow simple rules.

* Research Associate FWO (Fund for Scientific Research-Flanders)

To be more accurate we can define intelligence as *the ability to solve problems.* A system is more intelligent than another system if in a given time interval it can solve more problems, or find better solutions to the same problems. A group can then be said to exhibit collective intelligence if it can find more or better solutions than the whole of all solutions that would be found by its members working individually.

1.1. Examples of collective intelligence

All organizations, whether they be firms, institutions or sporting teams, are created on the assumption that their members can do more together than they could do alone. Yet, most organizations have a hierarchical structure, with one individual at the top directing the activities of the other individuals at the levels below. Although no president, chief executive or general can oversee or control all the tasks performed by different individuals in a complex organization, one might still suspect that the intelligence of the organization is somehow merely a reflection or extension of the intelligence of its hierarchical head.

This is no longer the case in small, closely interacting groups such as soccer or football teams, where the “captain” rarely gives orders to the other team members. The movements and tactics that emerge during a soccer match are not controlled by a single individual, but result from complex sequences of interactions. Still, they are simple enough for an individual to comprehend, and since soccer players are intrinsically intelligent individuals, it may appear that the team is not really more intelligent than its members.

Things are very different in the world of social insects (Bonabeau et al. 1997; Bonabeau & Theraulaz 1994). The way that ants map out their environment, that bees decide which flower fields to exploit, or that termites build complex mounds, may create the impression that these are quite intelligent creatures. The opposite is true. Individual insects have extremely limited information processing capacities. Yet, the ant nest, bee hive or termite mound as a collective can cope with very complex situations.

What social insects lack in individual capabilities, they seem to make up by their sheer numbers. In that respect, an insect collective behaves like the self-organizing systems studied in physics and chemistry (Bonabeau et al. 1997): very large numbers of simple components interacting locally produce global organization and adaptation. In human society, such self-organization can be found in the “invisible hand” of the market mechanism. The market is very efficient in allocating the factors of production so as to create a balance between supply and demand (cf. Heylighen 1997). Centralized planning of the economy to ensure the same balanced distribution would be confronted with a “calculation problem” so complex that it would surpass the capacity of any information processing system. Yet, an efficient market requires its participating agents to follow only the most simple rules. Simulations have shown that even markets with “zero intelligence” traders manage to reach equilibrium quite quickly (Gode & Sunder 1993).

The examples we discussed show relatively low collective intelligence emerging from highly intelligent individual behavior (football teams) or high collective intelligence emerging from “dumb” individual behavior (insect societies and markets). The obvious question is whether high collective intelligence can also emerge from high individual intelligence. Achieving this is everything but obvious, though. The difficulty is perhaps best illustrated by the frustration most people experience with committees and meetings. Bring a number of very competent people together in a room in order to devise a plan of action, tackle a problem or reach a decision. Yet, the result you get is rarely much better than the result you would have got if the different participants had tackled the problem individually. Although committees are obviously important and useful, in practice it appears difficult for them to realize their full potential. Let us therefore consider some of the main impediments to the emergence of collective intelligence in human groups.

1.2. Obstacles to collective intelligence

First, however competent the participants, their individual intelligence is still limited, and this imposes a fundamental restriction on their ability to cooperate. Although an expert in his own field, Mr. Smith may be incapable to understand the approach proposed by Ms. Jones, whose expertise is different. Even if we assume that Mr. Smith would be able to grasp all the ramifications and details of Ms. Jones's proposal, he probably would still misunderstand what she is saying, simply because he interprets the words she uses in a different way than the one she intended. Both verbal and non-verbal communication are notoriously fuzzy, noisy and dependent on the context or frame of reference. Even if everyone would perfectly understand everyone else, many important suggestions during a meeting would never be followed up. In spite of note taking, no group is able to completely memorize all the issues that have been discussed.

Another recurrent problem is that people tend to play power games. Everybody would like to be recognized as the smartest or most important person in the group, and is therefore inclined to dismiss any opinion different from his or her own. Such power games often end up with the establishment of a "pecking order", where the one at the top can criticize everyone, while the one at the bottom can criticize no one. The result is that the people at the bottom are rarely ever paid attention to, however smart their suggestions. This constant competition to make one's voice heard is exacerbated by the fact that linguistic communication is *sequential*: in a meeting, only one person can speak at a time.

It seems that the problem might be tackled by splitting up the committee into small groups. Instead of a single speaker centrally directing the proceedings, the activities might now go on in parallel, thus allowing many more aspects to be discussed simultaneously. However, now a new problem arises: that of *coordination*. To tackle a problem collectively, the different subgroups must keep close contact. This implies a constant exchange of information so that the different groups would know what the others are doing, and can use each other's results. But this again creates a great information load, taxing both the communication channels and the individual cognitive systems that must process all this incoming information. Such load only becomes larger as the number of participants or groups increases.

For problems of information transmission, storage and processing, computer technologies may come to the rescue. This has led to the creation of the field of Computer-Supported Cooperative Work (CSCW) (see e.g. Smith 1994), which aims at the design of Groupware or "Group Decision Support Systems". CSCW systems can alleviate many of the problems we enumerated. By letting participants communicate anonymously via the system it can even tackle the problem of pecking order, so that all contributions get an even opportunity to be considered. However, CSCW systems are typically developed for small groups. They are not designed to support self-organizing collectives that involve thousands or millions of individuals.

But there is a technology which can connect those millions: the global computer network. Although communities on the Internet appear to self-organize more efficiently than communities that do not use computers, the network seems merely to have accelerated existing social processes. As yet, it does not provide any active support for collective intelligence. The present paper will investigate how such a support could be achieved, first by analysing the mechanisms through which collective intelligence emerges in other systems, then by discussing how available technologies can be extended to implement such mechanisms on the network.

2. Collective Problem-Solving

To better understand collective intelligence we must first analyse intelligence in general, that is, the ability to solve problems. A *problem* can be defined as a difference between the present situation, as perceived by some agent, and the situation desired by that agent. Problem-solving then means finding a sequence of actions that will transform the present state via a number of intermediate states into a goal state. Of course, there does not need to be a single, well-defined goal: the agent's "goal" might be simply to get into any situation that is more pleasant, interesting or amusing than the present one. The only requirement is that the agent can *distinguish* between subjectively "better" (preferred) and "worse" situations (Heylighen 1988, 1990).

To generalize this definition of a problem for a collective consisting of several agents it suffices to aggregate the desires of the different agents into a collective preference and their perceptions of the present situation into a collective perception. In economic terms, the aggregate desire becomes the market "demand" and the aggregate perception of the present situation becomes the "supply" (Heylighen, 1997). It must be noted, though, that what is preferable for an individual member is not necessarily what is preferable for a collective (Heylighen & Campbell, 1995): in general, a collective has emergent properties that cannot be reduced to mere sums of individual properties. (Therefore, the aggregation mechanism will need to have a non-linear component.) In section 3, we will discuss in more detail how such an aggregation mechanism might work.

On way to solve a problem is by trial-and-error in the real world: just try out some action and see whether it brings about the desired effect. Such an approach is obviously inefficient for all but the most trivial problems. Intelligence is characterised by the fact that this exploration of possible actions takes place mentally, so that actions can be selected or rejected "inside one's head", before executing them in reality. The more efficient this mental exploration, that is, the less trial-and-error needed to find the solution, the more intelligent the problem-solver.

2.1. Mental maps

The efficiency of mental problem-solving depends on the way the problem is *represented* inside the cognitive system (Heylighen 1988, 1990). Representations typically consist of the following components: a set of problem states, a set of possible actions, and a preference function or "fitness" criterion for selecting the most adequate actions. The fitness criterion, of course, will vary with the specific goals or preferences of the agent. Even for a given preference, though, there are many ways to decompose a problem into states and actions. Changing the way a problem is represented, by considering different distinctions between the different features of a problem situation, may make an unsolvable problem trivial, or the other way around (Heylighen 1988, 1990).

Actions can be represented as operators or transitions that map one state onto another one. A state that can be reached from another state by a single action can be seen as a neighbor of that state. Thus, the set of actions induces a topological structure on the set of states, transforming it into a problem *space*. The simplest model of such a space is a network, where the states correspond to the nodes of the network, and the actions to the edges or links that connect the nodes. The selection criterion, finally, can be represented by a preference function that attaches a particular weight to each link. This problem representation can be seen as the agent's *mental map* of its problem environment.

A mental map can be formalized as a weighted, directed graph $M = \{N, L, P\}$, where $N = \{n_1, n_2, ..., n_m\}$ is the set of nodes, L $N \times N$ is the set of links, and P: $L \rightarrow [0, 1]$, is the preference function. A problem solution then is a connected path

$C = (c_1, ..., c_k)$ N such that c_1 is the initial state, c_k is a goal state, and for all i $\{1, \ldots, k\}$: (c_i, c_{i+1}) L.

To solve a problem, you need a general heuristic or search algorithm, that is, a method for selecting a sequence of actions that is likely to lead as quickly as possible to the goal. If we assume that the agent has only a local awareness of the mental map, that is, that the agent can only evaluate actions and states that are directly connected to the present state, then the most basic heuristic it can use is some form of "hill-climbing" with backtracking. This heuristic works as follows: from the present state choose the link with the highest weight that has not been tried out yet to reach a new state; if all links have already been tried, backtrack to a state visited earlier which still has an untried link; repeat this procedure until a goal state has been reached or until all available links have been exhausted. The efficiency of this method will obviously depend on how well the nodes, links and preference function reflect the actual possibilities and constraints in the environment.

The better the map, the more easily problems will be solved. Intelligent agents, then, are characterized by the quality of their mental maps, that is, by the knowledge and understanding they have of their environment, their own capacities for action, and their goals. Increasing problem-solving ability will generally require two complementary processes: 1) enlarging the map with additional states and actions, so that until now unimagined options become reachable; 2) improving the preference function, so that the increase in total options is counterbalanced by a greater selectivity in the options that need to be explored to solve a given problem.

2.2. Coordinating individual problem-solutions

Let us apply this conceptual framework to collective problem-solving. Imagine a group of individuals trying to solve a problem together. Each individual can explore his or her own mental map in order to come up with a sequence of actions that constitutes part of the solution. It would then seem sufficient to combine these partial solutions into an overall solution. Assuming that the individuals are similar (e.g. all human beings or all ants), and that they live in the same environment, we may expect their mental maps to be similar as well. However, mental maps are not objective reflections of the real world "out there": they are individual constructions, based on subjective preferences and experiences (cf. Heylighen 1999). Therefore, the maps will also be to an important degree different.

This diversity is healthy, since it means that different individuals may complement each others' weaknesses. Imagine that each individual would have exactly the same mental map. In that case, they would all find the same solutions in the same way, and little could be gained by a collective effort. (In the best case, the problem could be factorized into independent subproblems, which would then be divided among the participating individuals. This would merely speed up the problem-solving process, though; it would not produce any novel solutions).

Imagine now that each individual would have a different mental map. In that case, individuals would need to communicate not only the (partial) solutions they have found, but the relevant parts of their mental maps as well, since a solution only makes sense within a given problem representation. This requires a very powerful medium for information exchange, capable of transmitting a map of a complex problem domain. Moreover, it requires plenty of excess cognitive resources from the individuals who receive the transmissions, since they would need to parse and store dozens of mental maps in addition to their own. Since an individual's mental map reflects that individual's total knowledge, gathered during a lifetime of experience, it seems very unlikely that such excess processing and storage capacity would be available. If it were, this would mean that the individual has used only a fraction of his or her capacities for cognition, and this implies an individual who is very inexperienced or simply stupid. Finally, even if individuals could effectively communicate their views, there is no obvious mechanism to resolve the conflicts that would arise if their proposals contradict each other. It seems that we have come back to our problem where we have intelligent individuals but a dumb collective.

Let us see whether investigations of existing intelligent collectives can help us to overcome this problem of coordination between individuals.

2.3. Stigmergy

While studying the way termites build their mounds, the French entomologist Pierre Grassé (1959) discovered an important mechanism, which he called "stigmergy". He observed that at first different termites seem to drop mud more or less randomly. However, the presence of a heap of mud incites other termites to add mud to that heap, rather than start a heap of their own. The larger the heap, the more attractive it is to further termites. Thus, the small heaps will be abandoned, while the larger ones will grow into tall columns. Since the bias to add mud in those places where the concentration of mud is highest continues, the columns moreover have a tendency to grow towards each other, until they touch. This produces an arch, which will itself grow until it touches other arches. The end result is an intricate, cathedral-like structure of interlocking arches.

This is obviously an example of collective intelligence. The individual termites follow extremely simple rules, and have no memory of either their own or other individual's actions. Yet, collectively they manage to coordinate their efforts so as to produce a complex, seemingly well-designed structure. The trick is that they coordinate their actions without direct termite-to-termite communication. The only "communication" is indirect: the mud left by one termite provides a signal for other termites to continue work on that mud. Thus, the term *stigmergy,* whose Greek components mean "mark" (*stigma*) and "work" (*ergon*).

The fundamental mechanism here is that the environment is used as a shared medium for storing information so that it can be interpreted by other individuals. Unlike a message (e.g. a spoken communication) which is directed at a particular individual at a particular time, a stigmergic signal can be picked up by any individual at any time. A spoken message that does not reach its addressee, or is not understood, is lost forever. A stigmergic signal, on the other hand, remains, storing information in a stable medium that is accessible by everyone.

The philosopher Pierre Lévy (1997) has proposed a related concept to understand collective intelligence, that of a shared "object". For example, a typical object is the ball in a soccer game. Soccer players rarely need to communicate directly, e.g. by shouting directions at each other. Their activities are coordinated because they are all focused on the position and movement of the ball. The state of the ball incites them to execute particular actions, e.g. running toward the ball, passing it to another player, or having a shot at the goal. Thus, the ball functions as a stigmergic signal, albeit a much more dynamic one than the mud used by termites. Another typical "object" discussed by Lévy (1997) is money. It is the price, i.e. the amount of money you get for a particular good, which incites producers to supply either more or less of that good. Thus, money is the external signal which allows the different actors in the market to coordinate their actions (cf. Heylighen 1997).

The difference between Lévy's "object" and Grassé's stigmergic signal, perhaps, is that the former changes its state constantly, while the latter is relatively stable, accumulating changes over the long term. The stigmergic signal functions like a *long-term memory* for the group, while the object functions like a *working memory,* whose changing state represents the present situation. In fact, you do not even need an external object to hold this information. The soccer players are not only influenced by the position and movement of the ball, but also by the position and movement of the other players. This perceived state of the collective functions as a shared signal that coordinates the actions of the collective's members. The coordinated actions exhibited by the individuals in a swarm (flocks of birds, shoals of fish, herds of sheep, etc.) are similarly based on a "real-time" reaction to the perceived state of the other individuals.

2.4. Collective Mental Maps

In the examples of stigmergy or shared objects we discussed until now, the problem-solving actions seem to be purely physical: amassing mud, kicking a ball towards the goal, producing goods. We might wonder whether stigmergy could also be used to support problem-solving on the mental plane, where sequences of actions are first planned in the abstract before they are executed in reality. Again, insect societies can provide us with a most instructive example. Ants that come back from a food source to their nest leave a trail of chemical signals, *pheromones*, along their path. Ants that explore the surroundings, looking for food, are more likely to follow a path with a strong pheromone scent. If this path leads them to a food source, they will come back along that path while adding more pheromone to the trail. Thus, trails that lead to sources with plenty of food are constantly reinforced, while trails that lead to exhausted sources will quickly evaporate.

Imagine two parallel trails, A and B, leading to the same source. At first, an individual ant is as likely to choose A as it is to choose B. So, on average there will be as many ants leaving the nest through A as through B. Let us assume that path B is a little shorter than A. In that case, the ants that followed B will come back to the nest with food a little more quickly. Thus, the pheromones on B will be reinforced more quickly than those on A, and the trail will become relatively stronger. This will entice more ants to set out on B rather than A, further reinforcing the gains of B relative to A. Eventually, because of this positive feedback, the longer path A will be abandoned, while the shorter path B will attract all the traffic. Thus, the ants are constantly tracing and updating an intricate network of trails which indicate the most efficient ways to reach different food sources. Individual ants do not need to keep the locations of the different sources in memory, since the collectively developed trail network will always be there to guide them.

This example may seem similar to the mud collecting termites. The difference is that the ants leaving pheromone are not making any physical contribution to the solution of their problem (collecting food), unlike the termites whose actions directly contribute to the mound building. They are merely providing the collective with a map to guide them through the terrain. In fact, the trail network functions like an external mental map, which is used and updated by all ants. We will call such an exteriorized, shared, cognitive system a *collective mental map* (CMM). Let us investigate this concept in more detail.

A collective mental map functions first of all as a shared *memory*. Various discoveries by members of the collective are registered and stored in this memory, so that the information will remain available for as long as necessary. The storage capacity of this memory is in general much larger than the capacities of the memories of the individual participants. This is because the shared memory can potentially be inscribed over the whole of the physical surroundings, instead of being limited to a single, spatially localized nervous system. Thus, a collective mental map differs from cultural knowledge, such as the knowledge of a language or a religion, which is shared among different individuals in a cultural group but is limited by the amount of knowledge a single individual can bear in mind.

In human evolution, the first step towards the development of a CMM was the invention of writing. This allowed the storage of an unlimited amount of information outside of individuals brains. Unlike a real CMM, however, the information in books is shared only to a limited extent. Not all books can be accessed by all individuals. This was particularly true before the invention of printing, when only a few copies of any given book existed in the world. Although libraries now provide a much wider access for people wishing to *read* books, there is still a very limited access for *writing* books. Although everybody could in principle write a book, very few books actually get published in such a way that they become accessible to a large number of people.

In a CMM, such as the ants' trail network, on the other hand, all individuals can equally contribute to the shared memory. They can in particular build on each others' achievements by elaborating, reinforcing or providing alternatives for part of the stored information. Books, on the other hand, are largely stand-alone pieces of knowledge, with very limited cross-references. It would be very difficult for me to take an existing book and start commenting, correcting or reinforcing on its content. If I want to add to the state of the art, I would rather need to write and publish a book from scratch, a very difficult and time-consuming affair.

The need for a universally and dynamically shared memory has been well understood by researchers in Computer-Supported Cooperative Work (e.g. Smith 1994). Discussions over a CSCW system will typically keep a complete trace of everything that has been said, which can be consulted by all participants, and to which all participants can at any moment add personal annotations. This collective register of activities is often called a shared "blackboard", "white board" or "workspace". However, a record of all communications does not yet constitute a mental map. The more people participate in a discussion and the longer it lasts, the more the record will grow, and the more difficult it will become to distil any useful guidelines for action out of it. Of course, you can allow the participants to edit the record and erase notes that are no longer relevant, as you would do with scribbles on a blackboard. But this again presupposes that the participants would have a complete grasp of all the information that is explicitly or implicitly contained in the record. And that means that the size of the "controlled" content of the blackboard cannot grow beyond the cognitive capacities of an individual. This obviously makes the shared blackboard a poor model for an eventual Internet-based support for collective intelligence.

A mental map is not merely a registry of events or an edited collection of notes, it is a highly selective representation of features relevant to problem-solving. The pheromone network does not record all movements made by all ants: it only registers those collective movements that are likely to help solve the ants' main problem, finding food. A mental map consists of problem states, possible actions that lead from one state to another, and a preference function for choosing the best action at any moment. These are all implicit in the pheromone network: a particular patch of trail can be seen simultaneously as a location or problem state, as an action linking to other locations, and as a preference, measured by the concentration of pheromone, for that action over other available actions. As it is clear that a CMM cannot be developed by merely registering and editing individual contributions, we will need to study different methods to collectively develop a mental map.

3. Mechanisms of CMM Development

3.1. Averaging preferences

Probably the most basic method for reaching collective decisions and avoiding conflicts is voting. This method assumes that all options are known by all individuals, and that the remaining question is to determine their aggregate preference. In the simplest case, every individual has one vote, which is given to the options that this individual prefers above all others. Adding all the votes together determines the relative preferences of the different alternatives for actions. (Usually, after a vote only the highest scoring option is kept, but this is not relevant for our model, where all options remain available). This is to some degree similar to the functioning of ant colonies, where the pheromone trail left by a particular ant can be seen as that ant's "vote" in the discussion of where best to find food.

In a more sophisticated version of the voting mechanism, individuals can distribute their voting power over different alternatives, in proportion to their individual preference functions. For example, alternative A might get a vote of 0.5, B 0.3, C 0.2 and D 0.0. In that case, the collective preference function P^{col} becomes simply an average of the n individual preference functions P^i:

(1) $$P^{col}(l_j) = \frac{1}{n}\sum_{i=1}^{n} P^i(l_j) = \frac{1}{n}\sum_{i=1}^{n} p_j^i$$

Johnson's (1998; see also Johnson et al. 1998) simulation of collective problem-solving illustrates the power of this intrinsically simple averaging procedure. In the simulation, a number of agents try to find a route through a "maze", from a fixed initial position to a fixed goal position. The maze consists of nodes randomly connected by links. In a first phase, the agents "learn" the layout of the maze by exploring it in a random order until they reach the goal. They do this by building up a preference function which attaches a weight to every link in the network they tried, but such that the last link used (before exiting the maze) in any given node gets the highest weight. In a second, "application" phase, they use this knowledge to find a short route, which now avoids all needless loops and dead-ends encountered during the learning phase. Since different agents have learned different preference functions, they will not all be successful to the same degree, and their best routes can greatly differ in length. However, Johnson (1998) showed that if the preference functions for a large number of agents are averaged, the route selected by that "collective" preference was significantly shorter than the average route found by a typical individual agent. In fact, if the collective consisted of a sufficiently large number of agents, the collective solution was better than the best individual solution.

This phenomenon might be explained by assuming that the different routes learned by the agents are all variations on the globally shortest route through the maze. Because of the time spent learning how to avoid poor routes, the preferred option at any node is more likely to be the optimal choice than any other choice. However, since the agents have no global understanding of the maze, most of their choices will still be less than optimal. These deviations from the optimum are caused by random factors, and therefore have no systematic bias in any particular direction. Because of the law of large numbers, we can expect these "fluctuations" to cancel each other out when many different preferences are averaged. This will only leave what the different routes have in common, namely their bias towards the optimal solution. It seems that a similar mechanism may apply to human decision-making, under the same condition that there is no systematic bias away from the optimum. Therefore, we may expect the average of the choices made by a large group to be effectively better than the choice made by a random individual.

Although Johnson's simulation exhibits collective intelligence the way we have defined it, the solutions found by a large collective (of the order of 100 individuals or more) are only somewhat better than the solutions found by a single agent. In real life, it would seldom seem worth the trouble employing a hundred people to solve a problem if one person could find a solution that is almost as good. As could be expected from the law of large numbers, the averaging mechanism provides *decreasingreturns*: the improvement produced by adding a fixed number of individuals to the collective will become smaller as the size of the collective increases.

3.2. Feedback

One reason that the collective intelligence produced by averaging adds relatively little to the intelligence of the participating individuals is that the procedure is very redundant: every individual must build up a mental map of the whole domain before any CMM can be initiated. Although Johnson's (1998) simulation may seem similar to the ant example, trail laying by ants is governed by a more sophisticated mechanism. The discovery of a large food source by a single ant is in general sufficient to start a trail which is then used and reinforced by growing numbers of other ants. No individual ant needs to explore the whole surroundings. Yet, collectively, the ants are likely to have explored the complete surroundings, and their trails are likely to provide a near complete map of these surroundings.

The reason is that collective trails are not simply the superposition of trails laid independently by different individuals. Trails interact in a *non-linear* way: a trail leading to a good source will be reinforced through a *positive feedback* loop, while a trail leading to an empty source will spontaneously decay (cf. Dorigo et. al 1996). Thus, a variety of local, individual contributions suffices to let a global map emerge. This is the hallmark of self-organization (cf. Bonabeau et al. 1997). The net effect is that exploration by the collective is much more efficient, since small, individual efforts that turn out to be successful are amplified into large, collective results by the feedback mechanism (in a sense, we might say that the decreasing returns of averaging are complemented by the increasing returns produced by positive feedback). Different applications (e.g. Dorigo et al. 1996, Schoonderwoerd et al. 1996, Bonabeau & Theraulaz 1994) have shown that such "ant-based" algorithms provide a powerful heuristic to solve a variety of problems.

The danger with positive feedback is that it can become *too* strong, so that a few reasonably good courses of action immediately attract all the activity, preventing the exploration of avenues that might produce even better solutions. This risk can be controlled by adjusting the individuals' sensitivity to the collective preference. If ants would always choose the path with the strongest pheromone scent, then all ants would immediately concentrate on a single food source. It would take a very long time after that food source was exhausted before they would start exploring other regions. In order to guarantee a more or less exhaustive covering of the surroundings, ants should deviate from the strongest trail with a non-zero probability. Increasing this probability will lead to more thorough exploration, but reduced efficiency of food gathering. In ants, this probability has probably evolved to a nearly optimal value because of natural selection: ant varieties with a too high or too low value for this parameter would have lost the competition with ant varieties characterized by a better tuned value.

Some simulation results confirm this intuition. In Dorigo et al.'s (1996) search algorithm inspired by ants, the best results were achieved when the trail sensitivity parameter was set to 1, that is, when the "ant's" probability to choose a link was proportional to the trail strength (on the assumption that all possible paths start out with the same non-zero strength). In the simulation by Chialvo and Millonas (1995) of how ants build "cognitive maps" it was shown that real trails only emerge for a particular range of values of the "osmotropotaxic sensitivity" parameter.

When we transpose the feedback model from ant colonies to human society, we are reminded of a more typically human mechanism for collective problem-solving: *discussion*. When people in a group state their preferences, so that others who have not yet made up their mind may be enticed to follow them, they usually do more than just make an evaluation: they give arguments for why they believe a particular option is better. These arguments may convince others that this is really the best option, or incite them to produce counter-arguments. In the best case, these arguments and counter-arguments will illuminate the broader implications of the different options, or even suggest a new option that combines the best aspects of the previous options. Thus, not only the preference function, but also the state space and set of actions can develop interactively.

The general process is elegantly demonstrated by the Policy Delphi method for collaborative problem-solving (Linstone & Turoff 1975). This method has been implemented as a group decision support system (Kenis & Bollaert 1992), which in turn has been extended for use in a Web environment (Naydenova 1995). In a computer-supported Policy Delphi session, the participants are first asked to state their preferred action to tackle a given problem. The participants are then asked to evaluate each of the (anonymous) proposals on a 5 point scale, and to give arguments for their agreement/disagreement. In the next round, they see the distribution of opinions and the additional arguments for each of the options. In the light of this new information, they get the chance to change their mind, by making a new evaluation and giving new arguments. This discussion can go on for several rounds until the opinions have stabilized. At that moment, the group can decide to choose the option that has gathered the highest average evaluation.

3.3. Division of labor

Both ant trail laying and voting implicitly assume that all participating individuals contribute evenly to the different aspects of the problem-solving. This only makes sense if all individuals are equally competent on all issues. Human society, however, is based on the *division of labor*: different individuals have different forms of expertise, and they typically limit their contributions to the domains they are most competent in. Cognitive specialization emerges spontaneously through a positive feedback mechanism: as illustrated by Gaines' (1994) simulation, individuals who were successful in solving a particular type of problem are likely to get more problems of that type delegated to them, and thus will develop a growing expertise in the domain. Specialization helps to overcome individual limitations: since not everybody can know everything, a group where different individuals know different things will collectively cover a much larger domain.

Formally, division of labor is characterized by the fact that the mental maps $M^i = \{N^i, L^i, P^i\}$ are different for the different individuals i. The collective mental map M^{col} can then be reconstructed by taking the union of the respective sets of nodes and links: $N^{col} = \bigcup_i N^i$, $L^{col} = \bigcup_i L^i$. The collective preference function P^{col} can again be expressed as the average of the individual preferences as in equation (1), with the additional assumption that $P^i(l_j) = 0$ if $l_j \notin L^i$, or developed by a sequential, feedback-based algorithm, as discussed in section 3.2.

The problem with specialization is that in order to collectively tackle a problem the different specialists need to communicate. If the specialists' mental maps are too different, they will have great difficulty understanding each other. This is a classical issue in multidisciplinary or interdisciplinary research. One way to bridge the gap is to make sure that there is always some overlap between different mental maps, so that two specialists from very different disciplines (say, a chemist and a biologist) will be able to communicate via one or more "interpreters" who belong to an intermediary discipline (say, molecular biology) that overlaps with both. In our formal model this would mean that the overall graph M^{col} is connected. The more paths there exist between two arbitrary locations in the network, the higher the probability of fruitful information exchange between the corresponding domains. If there are sufficient "interdisciplines" to cover all the gaps between specializations, diffusion of ideas from one domain of expertise to another will always be possible. This is Campbell's (1969) "fish scale model of omniscience".

The disadvantage of this method is that diffusion between distant domains of expertise can be very inefficient, since many "interpretation" processes will slow down and degrade the communication. An alternative approach is the development of a universal language or, rather, metalanguage (Heylighen 1990) , that would allow different mental maps to be expressed in the same, more abstract language, so that their similarities and difference become clear. This is the approach proposed by General Systems Theory (Boulding 1956) to integrate the different scientific disciplines, albeit with limited success until now. One way to apply this mechanism to the development of a CMM would be a hierarchical semantic network supporting different levels of abstraction. This could be developed by clustering similar nodes, and representing the resulting clusters by new, "higher order" nodes (Heylighen 1999).

A similar mechanism underlies decision-making in organizations, where the advice coming from different specialists is synthesized by a "generalist" manager who puts them in a broader perspective. This brings us back to the hierarchical model, where supervisors at the higher levels divide a task or problem into subproblems which are then delegated to specialists at the levels below. This assumes that at every stage, a single, "generalist" individual decides how the activities of the other individuals should be coordinated. The efficiency of the resulting problem-solving will therefore be limited by the information-processing capacity of the "executive".

3.4 Conclusion

The three basic mechanisms of averaging, feedback and division of labor give us a first idea of a how a CMM can be developed in the most efficient way, that is, how a given number of individuals can achieve a maximum of collective problem-solving competence. A collective mental map is developed basically by superposing a number of individual mental maps. There must be sufficient diversity among these individual maps to cover an as large as possible domain, yet sufficient redundancy so that the overlap between maps is large enough to make the resulting graph fully connected, and so that each preference in the map is the superposition of a number of individual preferences that is large enough to cancel out individual fluctuations. The best way to quickly expand and improve the map and fill in gaps is to use a positive feedback that encourages individuals to use high preference paths discovered by others, yet is not so strong that it discourages the exploration of new paths.

Let us now try to apply these general principles to a concrete medium, the global information network. This will help us to clarify, formalize and operationalize these notions.

4. From Web to Collective Mental Map

We noted that the information stored in books can be seen as a rudimentary CMM for human society, albeit one that is too static and fragmented. The present trend to move all written information on-line, so that it becomes immediately and universally accessible, is a first step towards removing these obstacles. The World-Wide Web, with its distributed hypermedia format, seems particularly well-suited as a medium to create a dynamic CMM. Let us review the main benefits of the Web.

First, the storage space provided by the millions of computers connected to the network is practically unlimited. Second, the stored information can be accessed virtually instantaneously, both for reading and for writing. If I have an idea that I would like to publicize, it suffices to write it down, save the document in HTML format on my server, and the text becomes immediately accessible to everyone in the world with an Internet connection. Unlike information in books, moreover, the HTML format allows different documents to be directly connected. Thus, I can comment on, or contribute to, other people's ideas, while having both my comments and the original documents immediately available to the readers. The hyperlinks which connect web pages turn the web into a huge directed graph, consisting of nodes and links. Apart from the preference function, this is the same structure as the one we postulated for a mental map.

However, as yet the Web provides little support for problem-solving. One difficulty is that the unlimited memory makes it easy to store everything: relevant as well as irrelevant information. As we saw when discussing the automatic registry provided by CSCW systems, memorizing too much information hinders rather than helps problem-solving. In order to tackle that information explosion, we need guidance in selecting what is relevant. The best known tools to filter out what is relevant are the *search engines*: they sieve through the whole of the web, but return only those documents that contain the keywords provided by the user. However, this "brute force" approach has serious shortcomings (Heylighen & Bollen 1996; Kleinberg 1998). First, with the on-going explosion in the number of documents, even quite specific keywords will result in hundreds or thousands of "hits", most of which are of poor quality. Second, in order to select relevant keywords, the user already needs to have a clear idea of how a potential solution would be formulated. The best document to solve the user's problem may actually use different keywords, and therefore the solution may never be found.

A more intelligent selection mechanism is implicit in the hypertext structure of the web. The author of a web document will normally only include links to other documents that are relevant to the general subject of the page, and of sufficient quality. Thus, locating one document relevant to your goals may be sufficient to guide you to further information on that issue. High quality documents, that contain clear, accurate and useful information, are likely to have many links pointing to them, while low quality documents will get few or no incoming links. Thus, although no explicit preference function is attached to a link, there is a preference implicit in the total number of links pointing to a document. This preference is produced collectively, by the group of all web authors. A first step towards turning the web into a CMM would consist in extracting this implicit information from existing web links.

Recently, two types of algorithms have been developed for this purpose: PageRank (Brin & Page 1998) and HITS (Kleinberger 1998). Both use a bootstrapping approach (cf. Heylighen 1999): they determine the quality or "authority" of a web page on the basis of the number and quality of the pages that link to it. Since the definition is recursive (a page has high quality if many high quality pages point to it), the algorithm needs several iterations to determine the overall quality of a page. Mathematically, this is equivalent to computing the eigenvectors of the matrix that represents the linking pattern in the selected part of the web. PageRank uses the linking matrix directly, HITS uses a product of the matrix and its transposed matrix. The latter method produces two types of pages: *authorities*, that are pointed to by many good "hubs" (indexes or lists of web pages), and *hubs*, that point to many good authorities. In combination with a keyword search, which restricts the pages for which the quality is computed to a specific problem "neighborhood", these methods seem to produce a much better quality in the answers returned for a query.

The disadvantage of these methods is that they are static: they merely use the (rather sparse) linking pattern that already exists; they do not allow the web to adapt to the way it is used, as a real CMM would do. To achieve this, other sources of implicit information can be "mined". People who merely browse the web by navigating from page to page express their preferences by the links they choose. The frequency of their link selections provides information about their subconscious preference function. The following sections will discuss algorithms to extract and use such implicit information.

4.1. Collaborative filtering

Recently a number of methods have been developed for the "collaborative filtering" or "social filtering" of information (Resnick et al. 1994; Shardanand & Maes 1995; Breeze et al. 1998). The main idea is to automate the process of "word-of-mouth" by which people recommend products or services to one another. If you need to choose between a variety of options with which you do not have any experience, you will often rely on the opinions of others who do have such experience. However, when there are thousands or millions of options, like in the Web, it becomes practically impossible for an individual to locate reliable experts that can give advice about each of the options. By shifting from an individual to a collective mode of recommendation, the problem becomes more manageable. Instead of asking opinions to each individual, you might try to determine an "average opinion" for the group, like in the voting and collective maze exploration examples we discussed before. This, however, ignores your particular interests, which may be different from those of the "average person". You would rather like to hear the opinions of those people who have interests similar to your own, that is to say, you would prefer a "division-of-labor" type of organization, where people only contribute to the domain they are specialized in.

The basic mechanism behind collaborative filtering systems is the following: 1) a large group of people's preferences are registered; 2) using a similarity metric, a subgroup of people is selected whose preferences are similar to the preferences of the person who seeks advice; 3) a (possibly weighted) average of the preferences for that subgroup is calculated; 4) the resulting preference function is used to recommend options on which the advice-seeker has expressed no personal opinion as yet. Typical similarity metrics are Pearson correlation coefficients between the users' preference functions and (less frequently) vector distances or dot products. The correlation coefficient between two users *a* and *b*, with p^a_i denoting *a*'s preference for option *i*, and $\overline{p}^a$ denoting the average preference of *a* over all options, is defined in the following way:

(2) $$R_{ab} = \frac{\sum_i (p_i^a - \overline{p}^a)(p_i^b - \overline{p}^b)}{\sqrt{\sum_i (p_i^a - \overline{p}^a)^2 \cdot \sum_i (p_i^b - \overline{p}^b)^2}}$$

If the similarity metric has indeed selected people with similar tastes, the chances are great that the options that are highly evaluated by that group will also be appreciated by the advice-seeker. The typical application is the recommendation of books, music CDs, or movies. More generally, the method can be used for the selection of documents, services or products of any kind.

The main bottleneck with existing collaborative filtering systems is the collection of preferences (cf. Shardanand & Maes 1995). To be reliable, the system needs a very large number of people (typically thousands) to express their preferences about a relatively large number of options (typically dozens). This requires quite a lot of effort from a lot of people. Since the system only becomes useful after a "critical mass" of opinions has been collected, people will not be very motivated to express detailed preferences in the beginning stages (e.g. by scoring dozens of music records on a 10 point scale), when the system cannot yet help them.

One way to avoid this start-up problem is to collect preferences that are *implicit* in people's actions (Nichols 1998). For example, people who order books from an Internet bookshop implicitly express their preference for the books they buy over the books they do not buy. Customers who have bought the same book are likely to have similar preferences for other books as well. This principle is applied by the Amazon web bookshop (www.amazon.com), which for each book offers a list of related books that were bought by the same people.

There are even more straightforward ways to collect implicit preferences on the web. One method is to register all the documents on a website that have been consulted by a given user (cf. Breeze et al. 1998). The list of all available documents, with preference 1 for those that have been consulted and preference 0 for the others, then determines a preference function for that user (cf. Breeze et al. 1998). Using a similarity metric on these preference vectors makes it possible to determine neighborhoods of users with similar interests.

4.2. Co-occurrence matrices

Since the documents consulted by users with similar interests are likely to be in a number of respects similar themselves, collaborative filtering makes it possible to determine clusters of related documents (cf. Breeze et al. 1998). The principle is the same as with the books that are assumed to be related to a given book, because they have been bought by the same people. However, we are now shifting our attention from similarities between users to similarities between options, as expressed implicitly by the users' preferences. This allows us to make abstraction of any specific users or groups in order to derive a collective preference function that describes associations *between* options, rather than merely evaluations *of* options. Thus, we can use this mechanism to develop a CMM.

Making abstraction of the users much simplifies the algorithms needed to calculate similarity. We can simply assume that two documents are more similar if more users have consulted both of them. However, the frequency of consultation depends not only on a document's similarity to other documents, but on its intrinsic value. Therefore, in order to determine the intrinsic strength of the relation between a document x and a document y we can use the *conditional probability* that a user would consult x given that that user also consulted y. That probability $P(x|y)$ determines a matrix M_{xy} which represents the connection strengths between documents:

(3) $$M_{xy} = P(x|y) = \frac{\#(x \& y)}{\#(y)}$$

$\#(x)$ stands here for the total number of users that consulted x, and $\#(x\&y)$ for the total number of users that consulted both x and y. The formula implies that the strength of the link from y to x is zero if there are no users that consulted both x and y, and reaches the maximum value of 1 if all users that consulted y also consulted x.

This formula is so general that we can apply it to other cases where two documents x and y appear together in a given selection. We will call this "appearing together" *co-occurrence.* An obvious source for co-occurrence data on the web are documents that contain lists of links to other documents. If two documents x and y appear in the same list, we can assume that the author of that list considered these documents to be equally relevant to the subject of that list, and therefore to be similar in some way. (This is similar to what is called "co-citation" in bibliometric research, see Small 1973, Pitkow & Pirolli 1997). The more often x and y appear together in another document, the more strongly we can assume them to be related, and therefore the more strongly the weight for the link that connects them. Co-occurrence data for web links are readily available. It suffices to use a search engine, such as AltaVista (altavista.digital.com), and search for all documents that contain links to either y, or x *and* y. The number of "hits" can then be entered into the formula for the conditional probability. (Another method is to collect bookmark lists from users, and analyse the co-occurrence of web pages in these lists. Rucker and Polanco (1997) have used this method in their Siteseer system to recommend particular pages to each user. Unfortunately, they did not specify the underlying algorithm.) The resulting values can be used to determine a list of weighted links connecting all web documents that have been examined in this way. This is a first step to turning the web into a CMM.

Note that such a procedure exploits the semantic topology which is implicit in the way different individuals' interests and expertises are distributed. For example, a document on cybernetics is likely to co-occur frequently with a document on complex adaptive systems, since people interested in cybernetics are usually also interested in complex adaptive systems. Similarly, a document on complex adaptive systems is likely to co-occur with a document on non-linear physics. However, non-linear physics will co-occur less frequently with cybernetics. This indicates that the field of complex adaptive systems must be situated somewhere in between cybernetics and non-linear physics in the semantic space of disciplines.

It may seem that with the reduction from collaborative filtering to co-occurrence we have lost the possibility to make recommendations for an individual, rather than for a single node. However, the more complex personal recommendations *P*' can be recovered by representing an individual preference function *P* on the set of options as a vector **p** = (p_1 p_2 p_3 ... p_n) and calculating the product of this vector with the co-occurrence matrix:

$$p'_i = \sum_j M_{ij} p_j \qquad (4)$$

This formula remains valid if we replace the binary preference function (either an option occurs in a given selection, or it does not) by a numerically valued preference function, where each option can have a range of values. In that case, we can write the following more general formula for the "co-occurrence" matrix:

$$M_{ij} = \frac{\sum_{k=1}^{n} p_i^k p_j^k}{\sum_{k=1}^{n} p_j^k} \qquad (5)$$

Note that when there is only one preference function (n=1), then according to (4), its preference vector is an eigenvector of the corresponding matrix defined by (5). If the preferences are normalized as probabilities ($\sum_i p_i = 1$), then the corresponding eigenvalue is 1. This is as it should be: without other preference functions to supplement missing information, a given preference function should remain invariant under the collaborative filtering procedure. Also note that when preferences are restricted to binary values, $p_i \in \{0,1\}$, then the expression (5) reduces to the conditional probability formula (3).

Until now, the procedure as we have defined it has been motivated purely theoretically. It is simpler and seems more universal than the procedures based on correlation coefficients or vector distances. However, it still must be tested empirically on existing collaborative filtering data (cf. Breeze et al. 1998). A first indication that such a procedure would work at least as well as existing procedures can be found in the "artist-artist" algorithm tested by Shardanand & Maes (1995), which gave results similar to the more traditional collaborative filtering algorithms. This algorithm is based on a Pearson correlation coefficient like in (2), but applied to relations between "artists" (options, like in our proposal) rather than between users. The present formula is similar to the correlation formula, except for the normalization, and the fact that it uses the "raw" score p_i^a rather than its deviation from the average score $p_i^a - \bar{p}^a$. The advantage of the present normalization is that it is asymmetric: a link from a popular option *j* to a less popular option *i* ($\sum_{k=1}^{n} p_j^k > \sum_{k=1}^{n} p_i^k$) will get a lower connection strength than the inverse link, according to (5). The effect is similar to the "inverse user frequency" normalization which was found by Breeze et al. (1998) to increase the accuracy of recommendations.

4.3. Using sequential selection data

The co-occurrence or collaborative filtering procedures to construct a collective preference function are intrinsically parallel: the link between two nodes is reinforced only because these nodes are *simultaneously* present in some selection. However, the basic activity on the web is sequential: a user will select one node *after* another. This sequential browsing pattern too can provide us with information about the users' collective preferences. Moreover, since browsing is an on-going, real-time activity, this information can allow us to continuously update the CMM, thus supporting an interactive, feedback-based mechanism (section 3.2) rather than the non-interactive "averaging" (section 3.1) implied by co-occurrence.

To extract this sequential information, we may again consider ant trail laying as a source of inspiration. Each time an ant uses a trail to find food, the trail gets reinforced. Similarly, we might increase the weight of a link in the web by a small, fixed amount each time a user selects this link. Frequently used links would thus get a higher weight than less frequently used links. By renormalizing the link strenghts after each operation, the links that did not get reinforced will lose strength relative to the others. This is similar to the evaporation of pheromones along an ant trail. Since a seemingly promising link can still lead to an uninteresting document (the equivalent of a trail leading to an exhausted food source), the system should ideally increase the weight of a link only in proportion to the user's evaluation of the resulting document. If we want to avoid burdening the user by requesting an explicit rating, we can use implicit data, such as the time spent reading the document, which seems to correlate well with explicit evaluations (Nichols 1998).

There is a basic difference between the web and the terrain that ants explore to find food, though. An ant does not have to choose between existing trails: it can always deviate and start a wholly new trail. A web user, on the other hand, can only choose between the links that are available on the given web page. Therefore, the reinforcement of existing links by usage is intrinsically more constrained than the exploration used by ants.

There are different ways to add more "creativity" to the procedure. An obvious method to introduce new links is to provide the user with a list of suggested links that are not coded in the page's HTML content. In principle, we could let the user choose to go from the given page to any other page that exists somewhere in the Web. With hundreds of millions of Web pages, though, this method would be clearly impractical. We could also generate a small, random collection of web pages, and let the user choose between these. The probability that one of theses pages would be relevant to the user who has selected the given page seems very small, though. We can provide the user with a selection that is more likely to be relevant by using co-occurrence or keyword similarity to find pages related to the present one. However, this will only change the relative weights of links within the larger class of co-occurring or similar pages, and not create any really new links.

4.4. Learning web algorithms

My collaborator Johan Bollen and I have developed a heuristic algorithm that has both unlimited “creativity” in proposing new links, and is strongly selective in its suggestions (Bollen & Heylighen 1996, 1999, Heylighen 1999). When a user follows a path $a \to b \to c$, the algorithm not only increases the weight of the direct links $a \to b$ and $b \to c$ (this is the “frequency” rule), but also of the indirect link $a \to c$ (the “transitivity” rule), and of the inverse links $b \to a$, and $c \to b$ (the “symmetry” rule). In that way, a number of links that were not initially available on the page get the chance to gather a non-zero weight. All these links are considered potentially relevant to the user. From those, the links with the highest weights are added to the web page as suggestions, so that the user can now select these links immediately. When a link thus becomes directly available, we may say that it has turned from “potential” to “actual”.

The transitivity rule opens up an unlimited realm of new links. Indeed, one or several increases in weight of $a \to c$ may be sufficient to make the potential link actual. The user can now directly select $a \to c$, and from there perhaps $c \to d$. This increases the strength of the potential link $a \to d$, which may in turn become actual, providing a starting point for an eventual further link $a \to e$, and so on. Eventually, an indefinitely extended path may thus be replaced by a single link $a \to z$. 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 weights. 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.

This basic mechanism is extended by the symmetry rule. When a user chooses a link $a \to b$, 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 $b \to a$ gets a weight 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 joint effect of symmetry and transitivity is much more powerful than that of any single rule. For example, consider two links $a_1 \to b$, $a_2 \to b$. The fact that a_1 and a_2 point to the same node seems to indicate that a_1 and a_2 have something in common, i.e. are related in some way. However, none of the rules will directly generate a link between a_1 and a_2. Yet, the repeated selection of the link $a_2 \to b$ may actualize the link $b \to a_2$ by symmetry. The repeated selection of the already existing link $a_1 \to b$ followed by this new link can then actualize the link $a_1 \to a_2$ through transitivity. Similar scenarios can be conceived for different orientations or different combinations of the links.

A remaining issue is the relative importance of these three rules. In other words, how large should the increase in weight be for each of the rules? If we choose unity (1) to be the bonus given by the frequency 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. They play a role similar to the ants’ trail sensitivity (section 3.2): higher values of *t* and *s* mean a higher probability for the creation of new links, but also a higher probability for the selection of irrelevant links.

4.5. Learning web applications

In order to test these algorithms in practice, we set up two experiments (Bollen & Heylighen 1996, 1999), one using all three rules, another using all rules except symmetry. We built a network consisting of 150 nodes, corresponding to the 150 most frequent nouns of the English language. All of the potential 149 × 149 links between nodes were given a small random weight to initialize the web. Every node would show the 10 strongest links, ordered according to their weights. The link weights 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 "knowledge", a user would choose a link to an associated word, such as "education" or "experience", but not to a totally unrelated word, such as "face". 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 "book". However, when reaching the node "book", they might be able to select there another association, such as "education". Through transitivity, a new link to "education" might then appear in the node "knowledge", displacing the weakest link in the list, while providing a much better association than the previously best one, "book".

The development of the associative network was surprisingly quick and efficient. After only 2500 link selections (out of 22,500 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 links are gradually introduced and rewarded until their weight reaches an equilibrium value. Table 1 shows the self-organization of the list of 10 strongest links from the node "knowledge", in four subsequent stages: the initial random linking pattern, after 200 steps (link selections), after 800 steps, and after 4000 steps. The position of these associated words shifted upwards in the list until they reached a position that best seemed to reflect their relative strength.

Table 1: self-organization of the 10 strongest links from the node "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

The effect of these rules for creating new links and changing the weights of existing rules is a continuous reorganization of the web so that it more clearly reflects the users' implicit structure of associations or relative preferences between nodes. We might say that a web with these reorganization rules "learns" the preferences of its users. The more it is used, the more it learns, and the better its structure will reflect the users' collective preferences.

The fact that it learns so quickly can be explained by a positive feedback mechanism similar to the one that enhances the ants' trail network (Bollen & Heylighen 1996).

Indeed, as soon as a link has gathered a sufficient weight because of transitivity or symmetry, it becomes "actual" and can now be directly selected by the user. Such a direct selection boosts its weight, and makes it move up in the ordered list of suggested links. Since users consult lists from top to bottom, the higher the position of the link, the higher the probability that it will be selected by a following user and thus further increase its strength. However, the positive feedback is not so strong that if a new, much better link would appear at the end of the list, it would be ignored by the users. If the new last link is clearly better than the top link until then, it will be selected and start to move up, until it finally reaches the top position.

In this small 150 node experiment, there was no division of labor: all users were equally likely to visit a particular node, and were equally competent to select a particular link. However, if a similar learning web system would be implemented on the Web as a whole, we should expect extensive specialization. A user who does not like sport is very unlikely to consult a website about baseball. Similarly, a user who does not understand anything about physics will not browse through a quantum mechanics site. The more a person is interested and expert in a domain, the more frequently he or she will use web documents about that domain. Thus, link weights in a large learning web will be learned primarily from the most competent users. Therefore, our learning web algorithms should not be expected to suppress controversial or eccentric preferences while merely promoting the "lowest common denominator", as many people have suggested to us. "Fringe" documents will be consulted basically by "fringe" users, and the links that the web learns from them will reflect the preferences of this fringe group, not the ones of the majority. Thus, the learning web algorithms preserve the diversity of perspectives which is essential to true collective intelligence, while producing a much more complete and coherent tissue of links between related documents.

4.6. Problem-solving in the CMM web

Both the co-occurrence algorithms and the learning web algorithms have the potential to transform the web into a true collective mental map, which continuously self-organizes and adapts to the changing preference of its users. The two mechanisms are complementary. Co-occurrence of links in existing web documents, possibly complemented by similarity between documents computed from the keywords they contain, seems a good basis to produce an initial list of weighted links for each node in the web. This list can then develop interactively according to something like the learning web algorithms. The question now is how we could most efficiently use the wealth of collective knowledge represented by the resulting distribution of links and weights.

For the individual user, the benefits would be obvious. Instead of being limited to the few links present (or absent) in the document being consulted, a user would be able to choose from an extensive, but intelligently selected list of related documents, ordered by the probability that they would be relevant. Such a list of suggested links has already been implemented by the Alexa corporation (www.alexa.com) and incorporated in the Netscape and Internet Explorer browsers. (The algorithms used by Alexa to find suggestions have unfortunately not been published.) This list would continuously adapt, reflecting newly created documents as they become available. It would function like the collectively developed trail network that guides an individual ant in its search for food. This would make it much easier to find the documents the user is looking for.

However, this assumes that the user already has a good idea of where to start looking. If one does not have any relevant document to start with, one could use a traditional search engine to find documents that contain the relevant keywords. These documents, through their learned links, could lead the user to other relevant documents, which do not necessarily contain the same keywords. However, users may still have to spend a lot of time browsing the web before they would find the documents that really answer their questions.

One way to speed up the process is to develop a software agent that browses the web instead of the user. The agent could be provided with a list of keywords that defines the problem and start with a selection of documents that contain those keywords. It would then explore further, linked documents in the order of their link strength. The importance it attaches to a newly found document would be a function of the incoming link strength and the degree to which the document matches the keywords (by using more advanced methods such as Latent Semantic Indexing (Deerweester et al. 1990), a document may semantically match a query without actually containing the keywords). Since documents that match none of the keywords are most probably irrelevant, they should get overall preference zero. Documents that match the keywords partially but that are strongly linked to documents that do score high on the keyword match, on the other hand, *are* likely to be relevant.

One way to formalize this intuition is to calculate the overall relevance as a product of the keyword score $K(d_j)$ for document d_j and link strength $P(l_{ij})$ for the connection from document d_i to document d_j. The overall relevance of a document depends not only on a single link, but on all incoming links from relevant documents. The larger the number of high-weight links that point to a document from documents that have already been evaluated as relevant, the more relevant the new document can be expected to be. Thus, we could express the degree of relevance R of a document d_j as:

(6) $$R(d_j) = K(d_j).\sum_i P(l_{ij}).R(d_i)$$

This equation can be interpreted as representing a process of *spreadingactivation* (cf. Salton & Buckley 1988; Pirolli et al. 1996). Each node that the agent encounters is "activated" to a degree proportional to the document's relevance to the query. This activation then spreads to linked documents, in proportion to the strength of the link. The total activation arriving in a document is the sum of the activations carried by all incoming links. This activation can then continue to spread by following the outgoing links to reach a new collection of documents. We can thus let activation spread through the network, while keeping track of the documents that have received the highest activation. After a certain number of iterations, when no highly activated documents are being added to the list anymore (that is, when activation has become diffuse), the highest scoring documents are returned to the user as best candidates for solving the problem.

We have implemented such a spreading activation program on the 150 word network produced by 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.

The problem with such a spreading activation algorithm on the web is that in order to use the formula for calculating the new activation $R(d_j)$, we already need to know all the activations of the other nodes d_i. After a few iterations, the number of these nodes becomes huge, as activation diffuses and covers an ever growing subset of the web. This puts a heavy computational load on the agent. One way to tackle this problem is to limit the collection of nodes subjected to spreading activation to all nodes within a given web domain, or all nodes returned by a keyword query complemented by their immediate neighbors in web space (nodes connected by one incoming or outgoing link). Within such a restricted set, the solutions $R(d_i)$ to equation (6) can then be found relatively quickly by iteration. This is similar to the methods used by Pirolli et al. (1996) and by Kleinberg (1998).

Another solution may be to replace the parallel search characteristic of spreading activation by a sequential algorithm that to some degree mimics the spread of activation. This algorithm could function in the following way. The agent starts with a given list of nodes, their relevance or activation values, and the weights of their links to further nodes. It chooses the node with the highest activation and "spreads" that activation via its links to the connected nodes. Some of these connected nodes will be new, and therefore will be added to the agent's list, with their newly calculated activation. However, a connected node can already be part of the agent's list. In that case, the newly calculated activation is added to the already stored activation for that node. Then, the agent simply repeats the procedure, again choosing the node with the highest activation from the new list, excluding any node that has been explored before, and updating the list with newly discovered nodes and additional activation values. This procedure is repeated for as long as there are significant increases in activation for the highest scoring nodes on the list. After the agent has decided that no high scorers seem to be coming forward anymore, it returns the nodes with the highest final activation to the user as suggested solutions.

This algorithm seems like a good heuristic for discovering nodes that receive high overall activation from their neighbors, without need for exhaustively spreading activation through all existing links. The algorithm mimics parallel search because the nodes it sequentially explores are in general not connected by a sequential path in the network. The agent's list of nodes is ordered only by overall activation received until then, not by direct linking patterns. Thus, the agent gradually expands the activated domain in different directions, while focusing on those directions that seem most promising. This is something that is very difficult to do for a human user, whose memory for visited nodes and their apparent relevance is strongly limited. Thus, the software agent could be expected to be much more efficient in searching through the collective mental map for the best solutions to the user's queries, by harnessing the full power of the collective knowledge stored in the CMM's linking pattern.

The recommendations made by collaborative filtering systems can be seen as a special case of this general "spreading activation" procedure. The user's list of preferences, on the basis of which recommendations are computed, is formally equivalent to a problem definition consisting of a vector of activations that lists potentially relevant options. Instead of one global recommendation for a given user, the system can produce different "recommendations" for different "problem definitions" entered by the same user. For example, in the Siteseer system (Rucker & Polanco 1997), one list of recommended pages is computed for every subsection of the user's list of bookmarks (preferred pages).

Moreover, the fact that spreading activation can be iterated allows us to overcome one of the main limitations of traditional collaborative filtering: when there is little overlap between preference vectors (or when users simply express few preferences), the system can make very few recommendations. If activation continues to spread from these few suggestions, however, it is likely to find additional relevant suggestions in a second iteration, and even more in a third and in a fourth one. For example, if both *a*'s and *b*'s list of liked paintings contain impressionist paintings, but these lists do not overlap, then a traditional system will never recommend *a*'s choices to *b*. Yet, it is likely that all impressionist paintings in the system will be at least *indirectly* linked by co-occurrence relations. An iterated spreading activation algorithm, therefore, is likely to activate the complete cluster of impressionist paintings starting from one or a few paintings belonging to that cluster. This is similar to the way neural networks are able to recognize patterns even when most of the input information is missing.

5. Summary and Conclusion

We have defined collective intelligence as collective problem-solving ability. Problem-solving requires a mental map, which represents the different problem states, actions, and preferences. Collective problem-solving therefore requires a collective mental map. Such a CMM is an external, shared memory, to which all members of the collective have some degree of read/write access. However, to efficiently support problem-solving, a CMM must offer more than an edited collection of public notes. Cognitive limitations make it impossible for any individual(s) to fully control or oversee the development of a CMM. Therefore, we need a global, self-organizing mechanism. The development of pheromone trail networks by ants provided us with a paradigm for the emergence of a CMM from a variety of local, individual contributions. Generalizing from this example, we suggested the following mechanisms for the development of a complex CMM: 1) superposition of several individual contributions, to average out fluctuations away from the optimum; 2) positive feedback between subsequent contributions, to amplify weak signals and accelerate overall development; 3) division of labor with overlap in the domains of expertise, to allow a diversity of specialized mental maps to be integrated into an encompassing CMM.

We then set out to apply these mechanisms to the World-Wide Web. The web as a shared memory already has the node and link structure characteristic of a mental map, but lacks the preference weighting of links. We examined two complementary techniques to extract a collective preference function from the preferences that implicitly guide the web's authors and users.

Collaborative filtering is a technique that assumes a "division-of-labor" differentiation between users, and that averages the preferences of a subgroup of similar individuals for the different options. However, by considering the co-occurrence of options in different user selections, it is possible to transform this collection of preference functions on options (nodes) into a global preference function (co-occurrence matrix) on links between options. This transformation simplifies the mathematical expressions, apparently without loss of information. Its usefulness still needs to be tested out in practice, though.

The complementary technique of learning web algorithms extracts the sequential link information from the on-going paths followed by users through web space. It thus directly uses the feedback mechanism to quickly blaze new trails, while indirectly supporting the division of labor mechanism. It has been successfully applied in a small scale experiment, but needs to be tested further in more realistic web environments.

Both techniques, on their own or (preferably) together, would enrich the web with an extensive pattern of weighted links. This could be used either to suggest related links to a user, or to support a software agent that uses spreading activation to retrieve the pages that are most relevant to a user's interests. In either case, it seems likely that such a CMM would greatly aid individuals or groups to find the solutions to their problems, by relying on the collective wisdom of all other users.

In conclusion, it seems that such a collective system would indeed be much more intelligent than its members, while still making full use of the individual intelligence of its content-providers and users. It could be further extended with techniques such as typed links and node clustering (Heylighen 1999), discussion, workflow, and market mechanisms (Heylighen 1997). Perhaps the best metaphor for such a world-wide, intelligent network would be the "global brain" (Heylighen & Bollen 1996). Although the first commercial applications of some of these techniques are already appearing, it is clear that we still need to do a lot of research before we can be certain that the proposed algorithms are ready for the task. There are many possible variations on the methods we discussed, and there are many other sources of collective knowledge to be mined. The best combined method will likely be found by testing out a variety of approaches in a variety of circumstances. I hope that the present paper will inspire other researchers to take up this challenge and start experimenting with various algorithms to support collective intelligence.

References

Bollen J. and 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, Vienna, 911-916.

Bollen J. and Heylighen F. (1996b). "Finding words through spreading activation" [http://pespmc1.vub.ac.be/SPREADACT.html].

Bollen J. and Heylighen F. (1999): "A system to restructure hypertext networks into valid user models", *New Review of HyperMedia and Multimedia*

Bonabeau E. and Theraulaz G. (1994), *Intelligence collective* Hermès, Paris.

Bonabeau E., Theraulaz G., Deneubourg J.-L., Aron S. and Camazine S. (1997), "Self-organization in social insects", *Trends in Ecology and Evolution* 12, 188-193.

Boulding K.: (1956) "General Systems Theory - The Skeleton of Science", *General Systems Yearbook* 1, 11-17.

Breese J.S., Heckerman D. and Kadie C. (1998), "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", *Proceedings 14th Conference on Uncertainty in Artificial Intelligence*, Madison WI: Morgan Kauffman.

Brin S. & L. Page (1998): "The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Proceedings of the 7th International World Wide Web Conference*, April 1998.

Campbell D. T. (1969), "Ethnocentrism of Disciplines and the Fish Scale Model of Omniscience", in: M. Sherif and C.W. Sherif (eds.), *InterdisciplinaryRelationships in the Social Sciences*, Chicago: Aldine, 328-348.

Chialvo, D.R. and Millonas, M.M. (1995), "How Swarms Build Cognitive Maps", *The biology and technology of intelligent autonomous agents*. Luc Steels (Ed.) NATO ASI Series, (144), 439-450.

Deerweester S., Dumais S., Landauer T., Furnas G. and Harshman R. (1990): "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science* 41:6, 391-407.

Dorigo M., Maniezzo V. and Colorni A. (1996), "The Ant System: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(1), 1-13.

Gaines B.R. (1994), "The Collective Stance in Modeling Expertise in Individuals and Organizations", *International Journal of Expert Systems* 71, 22-51.

Gode D.J. and Sunder S. (1993), "Allocative efficiencies of markets with zero-intelligence traders", *Journal of Political Economy* 101, 119-127.

Grassé-P. (1959), "La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalis et Cubitermes sp. La théorie de la stigmergie", *Insectes Sociaux* , 6, 41-83.

Heylighen F. & Campbell D.T. (1995): "Selection of Organization at the Social Level: obstacles and facilitators of metasystem transitions", *World Futures: the Journal of General Evolution 45,* p. 181-212.

Heylighen F. (1988): "Formulating the Problem of Problem-Formulation", in: *Cybernetics and Systems '88*, Trappl R. (ed.), Kluwer Academic Publishers, Dordrecht, p. 949-957.

Heylighen F. (1990), *Representation and Change*, Communication and Cognition, Gent.

Heylighen F. (1997), “The Economy as a Distributed, Learning Control System”, *Communication and Cognition- AI* , 13(2-3), 207-224.

Heylighen F. (1999), “Bootstrapping knowledge representations: from entailment meshes via semantic nets to learning webs”, *International Journal of Human-Computer Studies*

Heylighen F. and 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, Vienna, 917-922.

Johnson N. (1998), “Collective Problem-Solving: functionality beyond the individual” (Los Alamos National Laboratory technical report: LA-UR-98-2227; URL: http://ishi.lanl.gov/Documents/NLJsims_AB_v11.pdf).

Johnson N., Rasmussen S., Joslyn C., Rocha L., Smith S. and Kantor M. (1998), “Symbiotic Intelligence: self-organizing knowledge on distributed networks driven by human interaction”, *6th Int. Conference on Artificial Life*, eds. C. Adami, et al., MIT Press, Boston.

Kenis D. and Bollaert L. (1992), “MacPolicy, a group decision support system”, *Revue des Systèmes de Décision*, 1, 305.

Kleinberg J. (1998): “Authoritative sources in a hyperlinked environment”, *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*.

Lévy (1997), *Collective Intelligence: Mankind’s Emerging World in Cyberspace* Plenum, New York.

Linstone H. and Turoff M. (eds.) (1975), *The Delphi Method: techniques and applications*, Reading MA: Addison-Wesley.

Naydenova Z. (1995), *Structured Discussion over the World-Wide Web*, Masters Thesis, Vrije Universiteit Brussel, Faculty of Applied Sciences.

Nichols D.M. (1998) “Implicit Rating and Filtering”, *Proc. Fifth DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary, 10-12 November 1997, ERCIM, 31-36.

Pirolli, P., J. Pitkow, R. Rao (1996): “Silk from a sow’s ear: Extracting usable structures from the web”. *Conference on Human Factors in Computing Systems*, CHI ‘96, Vancouver, Canada.

Pitkow, J. and P. Pirolli (1997). “Life, death, and lawfulness on the electronic frontier”, *Conference on Human Factors in Computing Systems, CHI ‘97, Atlanta*, GA, Association for Computing Machinery, 383-390.

Resnick P, Iacovou N., Suchak M., Bergstrom, and Riedl J. (1994), “GroupLens: An open architecture for collaborative filtering of netnews”, *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, NC: ACM, 175-186.

Rucker J. and Polanco M.J. (1997): “SiteSeer: personalized navigation for the web”, *Communications of the ACM* 40(3), p. 73-75.

Salton G. and 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), 147-160.

Schoonderwoerd, R., Holland, O.E., Bruten, J.L., and Rothkrantz, L.J.M. (1996) “Ant-based load balancing in telecommunications networks”, *Adaptive Behavior*, 5(2), 169-207.

Shardanand U. and Maes (1995), “Social information filtering: Algorithms for automating ‘word of mouth’”,*Proceedings of CHI’95 -- Human Factors in Computing Systems*, 210-217.

Small H. (1973): “Co-citation in the Scientific Literature: a new measure of the relationship between two documents”, *Journal of the American Society for Information Science* 24, 265-269.

Smith J.B. (1994), *Collective Intelligence in Computer-Based Collaboration* Erlbaum, New York.