

CAHOOTS: A SOFTWARE PLATFORM FOR ENHANCING INNOVATION AND FACILITATING SITUATION TRANSFER

TONY MCCAFFREY

*Mechanical & Industrial Engineering, University of Massachusetts Amherst
Amherst, Massachusetts 01003, U.S.A
mccaffrey@ecs.umass.edu*

SUNDAR KRISHNAMURTY

*Mechanical & Industrial Engineering, University of Massachusetts Amherst
Amherst, Massachusetts 01003, U.S.A
skrishna@ecs.umass.edu*

XUZHANG LIN

*Mechanical & Industrial Engineering, University of Massachusetts Amherst
Amherst, Massachusetts 01003, U.S.A
xuzhang@student.umass.edu*

We introduce a new software platform called *Cahoots* for enhancing innovation in both groups and individuals. The word *cahoots* means working together. *Cahoots* possesses two main features. First, *Cahoots* embodies a visualization method that is intuitive for users and matches the pattern of successful problem solving. Second, a standardized problem solving grammar is natural for users to express themselves and guides them to express goals in a function-focused manner—a manner that is suited for the automated finding of analogous solutions (i.e. situation transfers). Both visual and linguistic representations meet the requirements of being intuitive to use and robust at modeling the task—and thus are what we call *ergosemantic* (i.e. from the ergonomic study of semantic representations). *Cahoots* is a software platform that enhances innovation by facilitating both human-human collaboration and human-machine collaboration.

Keywords: Innovation; creativity; situation transfer; problem solving; computational creativity.

1. Introduction

There is growing research and experimentation in developing a software platform that can assist groups in innovation (Greene, Thomsen, & Michelucci, 2011; Nielsen, 2012). Greene et al. (2011) describes two such platforms: PARCEL and *ePluribus*. In his book, *Reinventing Discovery*, Michael Nielsen surveys the various attempts to “amplify collective intelligence” (Nielsen, 2012). Nielsen then goes on to articulate the features that a successful software platform would require.

Although the most common group creativity technique, brainstorming (Osborn, 1953), is still popular, all evidence suggests that it does not work as well as nominal

brainstorming (Taylor, Berry, & Block, 1958; Gurman, 1968; Renzulli, Owen, & Callahan, 1974; Chatterjea & Mitra, 1976; Diehl & Stroebe, 1991; Mullen, Johnson, & Salas, 1991; Stroebe & Diehl, 1994; Paulus, Larey, & Ortega, 1995; Sutton & Hargadon, 1996). Nominal brainstorming has people generate ideas in isolation; then people gather to share their ideas, possibly build on them, rank them, and finally select some ideas to move ahead with.

In our view, the next generation of group innovation processes needs to be grounded in a visualization method that is intuitive to use and robustly models the task of problem solving (i.e. *ergosemantic*). In this way, the new tool should be doubly adapted (Clark, 1997), both to the user and the task of problem solving. The visual representation needs to spatialize the task in ways that lessen the cognitive burdens of the users. For example, to reduce the burden on human memory, the history of all the activity related to solving a problem should be visible in one glance. Spatial features of the representation should align with the meanings humans naturally give to spatial directions. For example, a breadth of diverse solutions should be horizontally spread out in space and thus represented using the breadth (i.e. width) dimension. The top-down direction is often naturally thought of as moving from the general to the specific, which should be reflected spatially as a goal becomes more specified. These spatial aspects are intuitive for users to understand. Our overall intent is to off-load as much knowledge as possible onto the representation scheme, which reduces the cognitive load on the user and thus frees up mental resources to focus on the content of the problem.

A visual representation should also be able to guide users in working together by structuring how their contributions relate to each other. Traditional brainstorming sessions—even ones using Post-It notes on a wall (Kelley & Littman, 2001)—lack this added level of structure that visually shows how the contributions relate to each other. Further, the visualization tool should encourage *microcontributions* (Nielsen, 2012). Traditional brainstorming in person as well as in on online blogs often contain so much text for each contribution that it is very difficult to process it all (Nielsen, 2012). A visualization tool should encourage *microcontributions* by limiting the amount of text that can accompany an entry. We go further by providing a structured grammar that is machine-friendly and thus permits greater machine assistance in the innovation process.

After presenting our visualization method, we then work through two example problems to illustrate the *Cahoots* process in action. The first example problem is a “closed” problem, which means that all the resources needed to solve the problem are already known. The second example problem is an “open” problem, which means that a crucial part of solving the problem is finding promising resources to use. Our second example will also illustrate, *Analogy Finder* (McCaffrey, 2012b), our software for partially automating the search for ideas that can be adapted from other situations (i.e. situation transfers).

2. Visualizing Problem Solving

Problem solving is generally conceived of as a back-and-forth process between top-down problem framing and bottom-up problem solving (Rittel & Webber, 1984; Simon, 1995). Although conceived of as bi-directional activity, current visual aids focus only on uni-directional activity (Greene et al., 2011; Nielsen, 2012). Top-down problem framing refines the expression of the goal. In our approach, bottom-up problem solving involves two sub-processes: (1) uncovering the features of the available resources (2) followed by interacting them together to accomplish the effects expressed by the goal (McCaffrey & Spector, 2012, 2013). Anchored in cognitive science, the overall motif is visually simple and judged to be intuitive to use by our pilot test groups. In essence, as shown in Figure 1, two networks grow toward each other: (1) a top-down network of goals grows into sub-goals and (2) a bottom-up network of interacting resources produces effects. When the two networks connect, the goal is achieved and the problem is solved.

This visualization method matches how problem solvers (i.e. solvers) generally interleave their problem solving activity (McCaffrey & Spector, 2012). Solvers may work top-down for a while to refine and rephrase their goal. They may then switch to work bottom-up by adding new resources or uncovering features of existing resources. Finally, they may interact several of the resources together to produce effects that may contribute to satisfying the goal. They interweave their efforts among these three activities until the gap between the networks is closed, which announces that the problem has been solved.

Other candidate platforms such as PARCEL and *ePluribus* only work in one direction—the top-down direction (Greene et al., 2011). Further, traditional Artificial Intelligence (AI) systems grounded in the work of Newell and Simon (1972) also work in only one direction. For traditional AI systems, a start state, which keeps track of all relevant properties, is manipulated by all known operations (i.e. interactions) to produce a new set of possible states. This process is repeated until a goal state is reached. The result

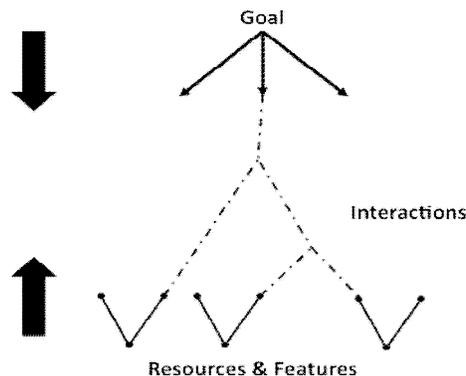


Figure 1. Bi-Directional (Bi-Net) network for problem solving.

is a tree of possible states that grows in one direction.

Based on initial pilot testing in our lab, our bi-directional network (i.e. *Bi-Net*) model in Figure 1 is intuitive for humans to understand. Individuals have worked with the *Bi-Net* model using paper and pencil. Groups worked together using Post-It Notes and string on a wall. Participants reported the following observations in our preliminary pilot work.

- the model presented a visual history of the problem solving activity that can be assessed in one glance
- the top-down growth of goals to sub-goals was intuitive to understand
- the bottom-up growth of features of resources was intuitive
- the model allowed a quick assessment of the status of the problem (i.e. how close it was to being solved)
- the model suggested directions for future problem solving activity: some people would choose to work on areas with lots of previous activity while others would choose to work on underexplored areas
- the model was structured enough to focus a group's activity but flexible enough to allow the introduction of new sub-goals and resources
- the model did not allow any single person to dominate the activity, as is often the case in brainstorming sessions, because people could simultaneously work on different parts of the bi-directional network
- it was easy to build upon each other's work: one person could add a feature, another could involve that feature in an interaction

In the next section, we step through the *Bi-Net* model applied to an example closed problem (i.e. one where all necessary resources are already known). This example illustrates the dynamics of the *Bi-Net* model, the simplicity of the problem solving grammar (which is presented more formally in Section 4), and how software can assist solvers to uncover overlooked features of the problem at key steps in the problem solving process.

3. Closed Problem: An Insight Problem

An insight problem, used in psychology experiments, presents solvers with everything they need to solve a problem but require solvers to notice at least one commonly overlooked feature of the problem (McCaffrey, 2011, 2012a). When solvers notice the key overlooked feature, they usually experience a small insight—often called an *aha moment*. All insight problems used in psychology experiments are amenable to visualization through the *Bi-Net* diagram (McCaffrey, 2011). Further, all insight problems are closed problems in the sense that the problem is solvable using only the resources that are presented in the description of the problem. In this section, we show how the *Bi-Net* model structures an insight problem called the *two rings problem* (McCaffrey, 2012a). The goal of the *two rings problem* is to fasten together two steel rings (each weighing three pounds) in a figure-eight configuration (Figure 2) so that if you pick up the top ring and move it about, the bottom ring will stay securely fastened.



Figure 2. Steel rings in a figure-eight configuration.

All you have to work with is a long candle, a match, and a 2-inch cube of steel. Most people try to bind the rings together with melted wax. However, the rings are too heavy for a wax bond. The solution’s key is to notice that the candle’s wick is a string that if extricated from the candle could tie the rings together in a secure manner.

Using the *Bi-Net* model, as shown in Figure 3, a solver might proceed to work on the problem in the following manner. A solver first articulates the goal *fasten ring to ring* at the diagram’s top and adds the objects that are available to solve the problem at the bottom. Next, a solver might work top-down by rephrasing the goal in various ways (e.g. *staple*, *clip*, and *tie*). This rephrasing can be partially automated (McCaffrey & Spector, 2012) since the online thesaurus *WordNet* (Miller, 1995) contains specific verbs, which express 61 different concrete ways to fasten things together. Humans by themselves can only list on average about 8 ways (McCaffrey & Spector, 2012).

As shown in Figure 4, a solver might then work bottom-up and examine the objects in order to look for overlooked features. This feature-uncovering process can also be partially automated as the *generic parts technique* (McCaffrey, 2011, 2012a) has been shown to help people counter *functional fixedness* (Duncker, 1945) by getting beneath

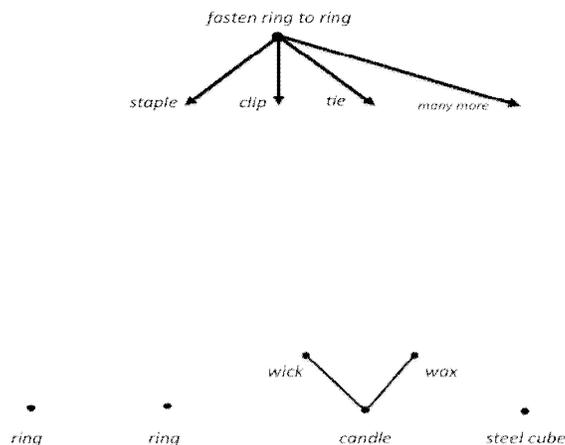


Figure 3. Bi-Net diagram for the two rings problem.

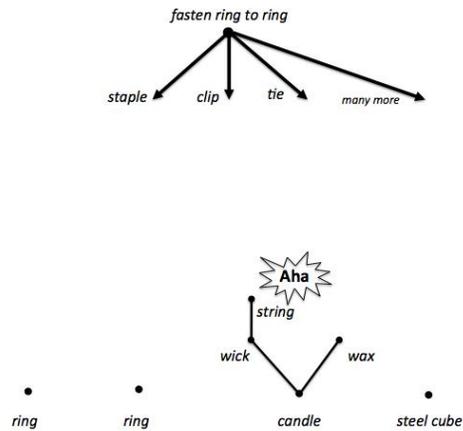


Figure 4. An aha moment of uncovering an obscure feature.

words that imply a common use of the object (e.g. *wick* implies burning to emit light) to more generic descriptions that open up the possibility of new uses (e.g. *string* implies tying things). In this case, noticing that a wick is made of string is the key to solving this problem and once this fact is noticed the problem becomes trivial (McCaffrey, 2011). The key obscure feature is tagged as an *aha moment* in Figure 4 because uncovering the key feature is often accompanied by the experience of a sudden insight (i.e. an *aha moment*).

Figure 5 shows the sequence of interactions that solvers easily construct to solve the problem. In this case, the string is freed from the wax by scraping the wax on the steel cube. Once freed, the string can achieve the goal by tying the rings together. More formally, the wax interacts with the steel cube to *free string from wax*. The string then interacts with the two rings to *tie ring to ring*, which achieves the overall goal of *fasten ring to ring*.

This sample problem illustrates the interweaving of top-down and bottom-up activity in problem solving. The *Bi-Net* model easily captures this bi-directional activity. This example further illustrates the three levels of a *Bi-Net* diagram: (1) goals and sub-goals, (2) resources and their features, and (3) interactions among the resources. Finally, this example shows how each verbal contribution to the *Bi-Net* model can be expressed in short phrases—often just single words are sufficient for the resources and features. The exact grammar for these contributions is formalized in the next section.

Before moving to the next section, we note that software can help connect the two networks by discovering the sequence of interactions needed to solve the problem (McCaffrey & Spector, 2011a, 2011b). In the case of the *two rings problem*, once the string is uncovered, our software program suggests a possible solution “a candle’s wick is made of string, which might be able to tie ring to ring.” Basically, the synonyms of the

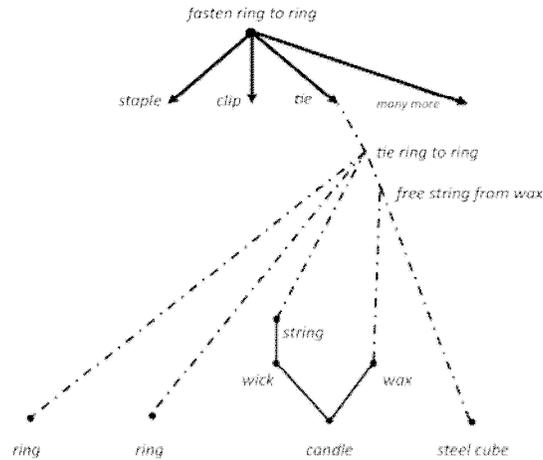


Figure 5. Sequence of interactions to solve the two rings problem.

goal verb *fasten* are intersected with the verbs associated with known resources and their parts. Since *tie* is a synonym of *fasten* and *tie* is also associated with *string*, the program can easily make the connection. In essence, the verbs associated with the goal express the actions that are needed to solve the problem. Verbs associated with the resources express the effects that the resources are known to produce. In this example, since the goal-verb *fasten* is a synonym of the resource-verb *tie*, this implies that the resource *string* might be able to help solve the problem.

More generally, as goals and sub-goals are entered in the top network, software can help look for resources that accomplish those goals. Vice versa, as resources and features are entered in the bottom network, software can also help look for goals that the resources/features might help accomplish.

4. Problem Solving Grammar

Each linguistic entry in Figures 3, 4, and 5 is expressed in short, standardized phrases. The resulting grammar makes the entries quick to read and easy to understand for humans while allowing software to easily parse them so that it can help humans make connections. By being quick to read, the grammar facilitates human-human collaboration and, by being easy to automatically parse, it facilitates human-machine collaboration.

All engineering goals can be expressed by a verb (e.g. *reduce* and *destroy*) that describes the desired change (Hirtz, Stone, McAdams, Szykman, & Wood, 2002). Building on this knowledge of goals, two grammatical components are added to the verb to more completely describe a goal: a noun phrase (e.g. *vibration* and *cancer cells*) to describe what needs changing and a list of prepositional phrases to describe any important constraints and relations (e.g. *over 1800 Hz* and *without damaging healthy cells*)

verb nounPhrase prepositionalPhrases

Figure 6. Grammar for goals and effects.

(McCaffrey & Spector, 2012, 2013; McCaffrey et al., 2013). Putting the three components together results in succinctly expressed goals such as *reduce vibrations over 1800 Hz* and *destroy cancer cells without damaging healthy cells*. Figure 6 shows the grammatical form of a goal.

Conveniently, the syntax for goals in Figure 6 can also be used to express the effects of interactions. A goal is a desired effect while an interaction produces an actual effect. For example, once a goal is achieved, the desired effect (e.g. *free string from wax* from Figure 5) becomes an actual effect that is accomplished by an interaction. Thus, an effect can also be expressed in the form *verb nounPhrase prepositionalPhrases*. The verb expresses the change that was accomplished. The noun phrase names that which is changed and the prepositional phrases describe any important constraints or relations.

In recent work, we have designed a simple grammar that can articulate everything that might appear in a *Bi-Net* diagram (McCaffrey & Spector, 2012). Using the Extended Backus-Naur Form (EBNF: Aho, Sethi, & Ullman, 1986), which is a compact notation mostly used to define the syntax of programming languages, in Figure 7, we propose a simple problem solving syntax. In EBNF, the “:=” symbol means “is defined as.” An item superscripted with a “*” means that there can be zero or more occurrences of that item. Straight brackets (i.e. []) indicate that what is inside the brackets is optional. A vertical line expresses a logical OR.

Briefly, we will explain the lines of the grammar in Figure 7. First, since a goal is just a desired effect, goals and effects are grammatically identical. Second, an effect (and also a goal) is defined as the now familiar form *verb nounPhrase prepositionalPhrase**, where the superscripted “*” means that there are zero or more prepositional phrases to describe important constraints or relations. Third, a noun phrase consists of a determiner (i.e. *a, an, the*), one or more adjectives, and either a single noun (e.g. *cancer*) or a noun-noun

```

goal ::= effect
effect ::= verb nounPhrase prepositionalPhrase*
nounPhrase ::= [determiner] adjective* [noun] noun
prepositionalPhrase ::= preposition nounPhrase
resource ::= nounPhrase
feature ::= adjective | nounPhrase

```

Figure 7. Problem solving grammar.

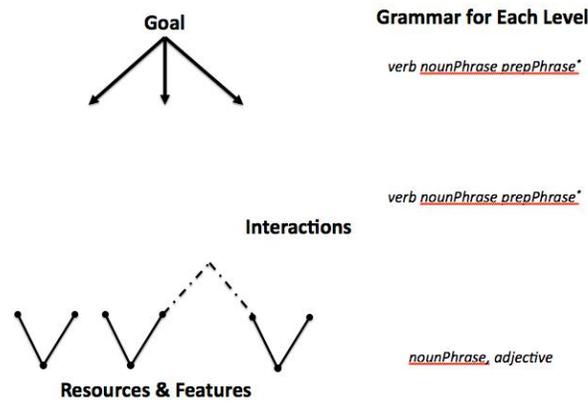


Figure 8. Grammar for each level of the Bi-Net diagram.

combination (e.g. *cancer cells*). Fourth, a prepositional phrase consists of a preposition (e.g. *on, to, over, without, etc.*) followed by a noun phrase. Fifth, a resource is a noun phrase (e.g. *steel cube*). Finally, a feature is either an adjective by itself (e.g. *long*) or a noun phrase (e.g. *wick*) which might also contain an adjective (e.g. *long wick*).

Figure 8 shows the relevant grammar for each level of the *Bi-Net* diagram. Using the grammar encourages brevity of expression (i.e. *microcontributions*), which is important to reduce the amount of reading necessary to comprehend an entry (Nielsen, 2012).

In the next sections, we will show how the problem solving grammar plays a crucial role in allowing software (i.e. *Analogy Finder*: McCaffrey, 2012b) to locate promising candidates for situation transfers (i.e. analogous solutions).

5. Situation Transfers in Closed and Open Problems

In a situation transfer during problem solving, a resource used in one situation is adapted to another situation. In the *two rings problem*, the closed problem illustrated in Section 3, a candle's wick is adapted to a new situation. In normal situations, a candle's wick is used for producing light. In the new situation of needing to fasten together two steel rings, the wick is repurposed to tie the rings together. Many insight problems used in psychology experiments involve repurposing an object, so they require a kind of situation transfer.

In contrast to a closed problem, for an open problem we do not know what resources will eventually solve the problem. A crucial aspect to solving open problems is being able to find promising resources. Often, a promising resource to solve a problem comes from another situation. Estimates are difficult to quantify but, based on our experience, perhaps upward of 90% of today's problems are solved by adapting an existing solution often from another field. Situation transfers between fields occur frequently.

Here are two examples. Francois Hennebique needed a stronger building material than concrete. While attending the Paris Exhibition of 1867, he found an exhibit for concrete flowerpots that contained a metal mesh. The need for reinforced concrete in the building industry, which made skyscrapers possible, had already been invented by Joseph Monier in the field of gardening.

Carbozyme Inc. needed to create an efficient way to remove carbon dioxide from the air so they could make filters for smokestacks. The key solution had been around for millions for years and was used every time anyone took a breath. The human lung efficiently removes carbon dioxide from the bloodstream and Carbozyme adapted this design from the field of human biology into an effective industrial smokestack filter.

In the next section, we focus on open problems, which require finding resources that can solve the problem. Since there is a high probability that the key resource already exists in another field, our goal is to devise a search method that can effectively find relevant resources in online sources such as patent databases, scientific journals, and in the files of a company's previous projects. Specifically, we will introduce a software method, *Analogy Finder* (McCaffrey, 2012b), for efficiently searching these electronic sources to find ideas that could be adapted to the needs of the current problem. In this way, *Analogy Finder* locates analogous solutions and engages in situation transfers. As illustrated in the next section, *Analogy Finder* leverages the grammar for a goal (i.e. *verb nounPhrase prepositionalPhrases*) so that it can recognize anything that accomplishes the goal—no matter what field it is from.

6. Open Problem: Automated Lawn Mowers

Suppose an engineering company is given the problem of needing to automatically shut off the cutting blades of a lawn mower when it does not detect grass. In this way, the mower shuts itself off when it is on gravel or pavement. This is the first step in the company's larger plan of making a completely automated mower that can guide itself to stay on the grass as it mows. Their initial goal is *detect grass on the ground*. As they begin to refine their goal, they create a level of sub-goals, each of which is under consideration as a possible way to achieve the main goal: *detect color*, *detect height of grass*, *detect surface types*, and *detect organic material* (Figure 9). As the goal network grows downward, the goals continue to be refined and specified. The goal network presents a visual history of all the possibilities that have been considered.

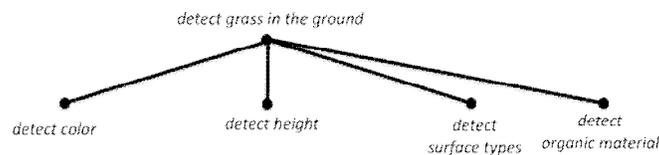


Figure 9. Initial goal network for lawn mower problem.

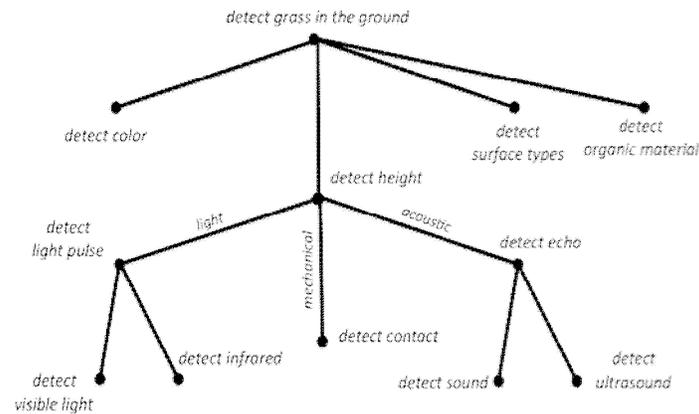


Figure 10. Refine goal network for lawn mower problem.

After discussion, the company engineers decide to first focus on trying to detect the height of the grass. If the mower detects zero height, then it knows that it is not on grass—but on pavement or gravel—and it should shut off the mower blades. Further, the engineers decide that if they can accurately detect the height of the grass, then they can also automate their mower to cut only sections of grass that need mowing and skip other sections that are already short enough. Figure 10 shows the expansion of the sub-goal network for *detect height*. Notice that the expanded sub-network is divided into energy types that might be used in the detection process (e.g. light, mechanical, and acoustic). This sub-network lists such things as *detect visible light*, *detect infrared*, *detect contact*, *detect sound*, and *detect ultrasound*.

If the goal network from Figure 10 is kept visible to the entire design team, then the team can make sure they will not forget or neglect promising avenues—if the options they are currently considering do not work out. For example, if their work on *detect height* does not solve the problem, the network in Figure 10 is a visual reminder that other options (e.g. *detect color*, *detect surface types*, or *detect organic material*) were proposed and should now be more fully explored. In this way, the goal network reduces the burden on human working memory. Further, the goal network made for the lawn mower project might also be useful for other company projects. For example, other projects might involve detecting the height of something—not necessarily just grass. Since the goals are expressed in a general manner (e.g. *detect height*), much of the goal network can be re-used for other problems—thus saving the company time and effort because it does not have to re-create a similar goal network for each new project. Even further, if this goal network was placed in a repository in the cloud, then anyone anywhere could benefit from the current information in this goal network and add to it. In other words, a *wiki* model could be used to save and refine goal networks for all to use and benefit from.

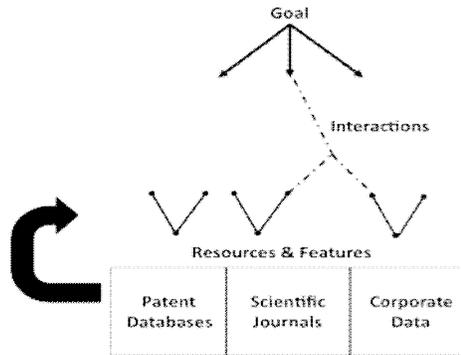


Figure 11. Populating resources from patent databases, journals, etc.

Once the goal network is sufficiently refined, then we can take the goals and search for actual solutions from diverse fields by searching patent databases, scientific journals, and corporate data. In contrast to closed problems where all the relevant resources are known beforehand, for open problems, we need to populate the bottom part of our *Bi-Net* model with promising resources. Figure 11 shows that solutions from various online sources can be used to populate the bottom level of the *Bi-Net* diagram.

Using *Analogy Finder* (McCaffrey, 2012b) to find possible solutions that could be transferred to our current situation, Figure 12 illustrates the results of using *Analogy Finder* for each sub-goal in the lower level of the goal network of Figure 10. Figure 12 indicates that a set of patents were found that achieved each sub-goal: *detect visible light*, *detect infrared*, *detect contact*, *detect sound*, and *detect ultrasound*. Any of these patents could be adapted to work for our lawn mower problem regardless of the field that they

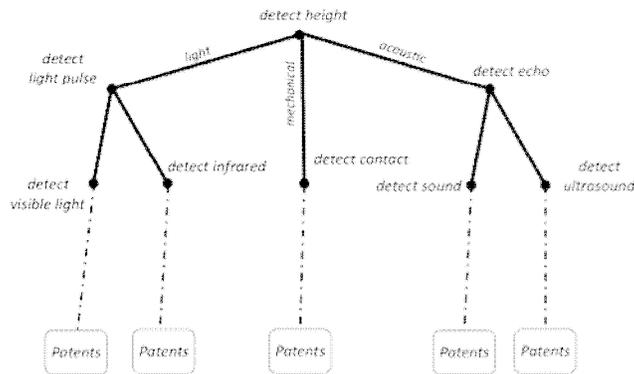


Figure 12. Patents to accomplish various goals.

originated from. For example, a solution for *detect ultrasound* might be adapted to our lawn mower problem from the fields of medical technology to detect pregnancy, physical therapy to relieve pain, submarine technology to detect underwater objects, security systems to detect movement, or industry to analyze the uniformity and purity of liquids—to name a few. Any of these applications of ultrasound from diverse fields may hold the key to adapting ultrasound to detect the height of grass in our lawn mower problem. The particular solution from any of these fields may be transferable to our situation.

To find diverse solutions that may be adaptable, here is how *Analogy Finder* works (McCaffrey, 2012b). Whenever a goal is expressed as a *verb nounPhrase* (e.g. *detect contact*), a search can take place in patent databases, scientific journals, and corporate data for any entities that accomplish a similar function/effect. In fact, the grammar forces engineers to express the desired effect in the essential terms needed as input for an analogy search engine. Our patent pending analogy search process enacts the following steps (McCaffrey, 2012b; McCaffrey, Krishnamurty, Grosse, & Wileden, 2013). We generate the synonyms (i.e. both more general hypernyms and more specific troponyms/hyponyms) (Miller, 1995) of both the verb and the noun phrase (e.g. for both the verb *detect* and the noun *contact*) and allow the user to edit the lists for best results. For example, a more general synonym (i.e. hypernym) of *detect* is *recognize* and a more specific synonym (i.e. troponym) is *locate*. Including both the hypernyms and troponyms creates a expansive set of synonyms to encompass the many general and specific ways that inventors from different fields might phrase the same goal. We then combine the synonymous verbs (e.g. *sense*, *register*, *recognize*) and nouns (e.g. *touch*, *connection*, *interaction*) into many phrases (e.g. *sense touch*, *sense connection*, *sense interaction*, *register touch*, etc.) and search patent databases, scientific journals, and corporate data for the occurrence of these phrases—permitting a match to occur when a few words fall between the words of the search phrase (e.g. *sense physical touch*, *register a partial connection*). If a goal uses prepositional phrases to describe constraints and relations, we then use select words from these prepositional phrases to reduce the number of search results.

By searching in this manner, we cast a wide net across all the domains of a patent database and all fields of scientific journals for ways that contact can be detected. Solutions will be found which include trip-wires from the field of explosives and touch screens from the field of computer science. The engineers will have many diverse solutions to choose from and further evaluate. One of them may very well be adaptable to detect the grass that moves beneath our lawn mower.

Analogy Finder is accessible from the website www.innovationaccelerator.com and currently searches the U.S. Patent database. *Analogy Finder* will continually be expanded to search other patent databases as well as scientific journals and corporate data.

7. Comparison to Other Analogy Approaches

Our grammar for analogical search shares some characteristics with other analogy research (Hirtz et al., 2002; Chakrabarti, Sarkar, Leelavathamma, & Nataraju, 2005a,

2005b; Linsey, Wood, & Markman, 2008; Oriakhi, Linsey, & Peng, 2011; Linsey et al., 2012; IHS Goldfire, 2013; Oxford Creativity Effects Database, 2013), but differs in important ways. In our *Analogy Finder* approach, a goal is expressed in the form *verb nounPhrase prepositionalPhrase*^{*}. We first conduct our search based on the *verb nounPhrase* and then use keywords from the prepositional phrases to reduce the search results. In contrast, Hirtz and colleagues (2002) proposes a *verb flow* construct to describe a goal. The verb describes the desired change while the flow may describe the type of material (e.g. solid) or the type of energy used (e.g. mechanical, chemical, or thermal). When searching for analogies, however, the flow is initially too abstract. If an engineer's goal is to reduce vibrations, for example, they may not initially care whether the solution is a mechanical or chemical process. Forcing engineers to name the flow as part of their goal restricts their search too early and leads to the omission of promising solutions of diverse energy types (e.g. mechanical, chemical, thermal, etc.) and diverse material types (e.g. solid, liquid, plasma, etc.). Searching just by the verb and the flow forces users to use search terms such as *reduce* and *mechanical energy* when they really need to search for *reduce vibrations* regardless of the type of flow involved. The flow information should be used later to group the matches by flow type, but it should not be used in the initial search.

Linsey and colleagues (Linsey et al., 2008; Oriakhi et al., 2011; Linsey et al., 2012) use the *WordTree* method, which focuses on exploring the troponyms (i.e. more specific synonyms) and hypernyms (i.e. more general synonyms) of a verb that express the desired change. Oriakhi and colleagues (2011) has developed a software tool to automatically generate and visualize a verb's synonyms. Since Linsey and colleagues focus on the verb, however, they are not able to further automate the process of searching for adaptable ideas in online sources (e.g. patent databases). Searching just by verbs is not restrictive enough to precisely name the desired effect. For example, searching the U. S. Patent database for the verb *reduce* returns over 1.6 million patents (search performed on July 13, 2013, using the USPTO website at www.uspto.gov) and is too broad of an effect. Searching by *reduce vibration* (i.e. *verb nounPhrase*), however, returns just over two thousand patents (search also performed on the same date at the same website) and more precisely targets the desired effect. Adding constraints from the prepositional phrase of the goal (i.e. *reduce vibrations over 1800 Hz*) then further hones the desired effect to greater precision. In sum, the *Analogy Finder* method, which operates on the grammatical form *verb nounPhrase prepositionalPhrase*^{*}, leads to more precise searches than the *WordTree* method and thus makes possible a higher degree of automation in the search process for analogous solutions.

The Oxford Creativity Effects Database (2013) uses a *verb noun* construct but only returns results predetermined by experts so does not engage in live searches of the patent database. From the biomimicry literature, Chakrabarti and colleagues (2005a, 2005b) uses a more sophisticated and hence non-intuitive grammar that involves verbs, adjectives, and nouns to represent actions, states, physical phenomena, physical effects, and inputs. Our grammar has simple expressions and fewer constructs to represent:

goals/effects, resources, and features. We argue that our simpler grammar is more intuitive and thus more ergosemantic. A commercialized product, IHS Goldfire (2013), extracts the verb and noun from an entered sentence but does not permit interactive editing of the synonym lists nor the progressive reduction of search hits made possible by leveraging the words from our prepositional phrases.

8. Situation Transfer through Bi-Net Reusability

Suppose that our engineering company has now solved its lawn mowing problem and is now approached with a new problem. It now needs to make an automatic window washer that can move up and across a building washing windows as it goes. One of the capabilities needed for this window washer is the ability to distinguish glass from non-glass (e.g. steel, concrete, brick, etc.) so that it knows what to wash.

Normally, engineers would have to start analyzing the window washer problem from scratch. However, many of the same solutions considered for *detect grass* are also relevant for the new goal *detect glass*. Figure 13 borrows many of the sub-goals originally used in Figure 10. For example, for the window washer problem, many of the sub-goals (e.g. *detect visible light*, *detect infrared*, *detect contact*, *detect sound*, *detect ultrasound*) could be relevant to detect the difference between glass and non-glass. In this example, a whole sub-network might be applicable to the new problem. Engineers can borrow existing sub-networks from previous projects so they do not have to create the entire goal network anew for each project. Further, many of the patent searches conducted for the *detect grass* problem will not have to be repeated for the *detect glass* problem.

In this way, repositories of goal networks and patent/journal searches can be reused

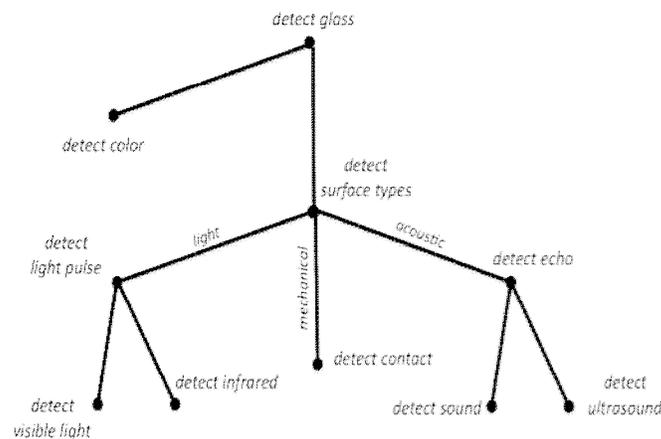


Figure 13. Different problems but similar goal networks.

over and over again as similar problems arise. The *Bi-Net* structure could develop into a repository of goals and a repository of possible solutions for each of those goals. Saving this information, which is adaptable to diverse problems, could save a company a great deal of time, effort, and money. Further, if shared in the cloud for the public to use, this information could save everyone a great deal of time and effort because people will no longer have to start each project from scratch.

9. Satisfying Nielsen's Requirements

After examining many attempts at online collaborative problem solving, Nielsen (2012) nicely summarizes the requirements known thus far for a collaborative problem solving platform. First, the system should encourage modularity by easily breaking down the problem down into small units. *Cahoots* breaks problem solving into its three main levels of activity: stating and refining goals, adding resources and uncovering their features, and interacting resources to produce effects. Each node that is added to one of the levels of a *Cahoots* network is a small unit: a goal, a sub-goal, a resource, a feature of a resource, an interaction, an effect of an interaction. Second, a collaborative platform should encourage small contributions (i.e. *microcontributions*) by individuals. *Microcontributions* lessen the amount of reading necessary to keep up to the current state of the problem as well as lowering the bar for contribution. If someone has a small insight, they should feel encouraged to contribute it instead of waiting until they have a large portion of the problem worked out. Another person might be able to quickly leverage the small insight into a further contribution. *Cahoots* encourages *microcontributions* both through its high modularity and the short phrases of its standardized grammar.

Third, the system should allow earlier work to be easily reused. Previous entries should be readily accessible so they can be examined and used as fodder for new contributions. The layout of *Cahoots* visually presents the history of the problem solving activity, so the history is readily available for reuse. Further, *Cahoots* allows work to be reused from previous projects as portions of goal networks and search results can be reused without alteration. Thus, earlier work is readily available to be built upon. Fourth, a collaborative platform should help people decide where to focus their attention. With *Cahoots*, the *Bi-Net* visualization method naturally directs solvers to where most of the problem solving activity has taken place. In this way, solvers can decide to contribute to the location where most others have already worked or re-direct their attention to an underexplored area of the problem space.

In sum, *Cahoots* meets each of the requirements specified by Nielsen (2012) in his book *Reinventing Discovery*.

10. Conclusions and Future Work

Cahoots is a strong candidate for a collaborative problem solving platform. *Cahoots* promotes modularity as it breaks problem solving at its natural joints of goals, resources, and interactions. Its *Bi-Net* diagram is ergosemantic: it is intuitive for solvers to understand and robustly models problem solving activity. Our problem solving grammar

promotes short, standardized expressions for all entries—which promotes *microcontributions*. *Cahoots* successfully models both closed problems and open problems. For open problems, *Cahoots* uses *Analogy Finder* to locate analogous solutions from diverse domains that can be transferred to the current situation. Much of the work performed for one problem can be reused for future problems thus saving solvers much time and effort.

Future research will use controlled experiments to test the many subjective claims made by our pilot test subjects. Specifically, we will compare *Cahoots* with brainstorming methods and measure the quantity of solutions, quality of solutions, and ease of use. Constructing a software prototype of *Cahoots* will allow us to test how it facilitates problem solving activity in remote, asynchronous settings. As the workforce of engineering companies becomes global, they require a collaborative platform that works online from remote locations and easily permits asynchronous additions to the *Bi-Net* diagram. In essence, a collaborative problem solving platform must work well across space (i.e. remote locations) and time (i.e. asynchronous contributions). *Cahoots* is a promising option to fulfill these important needs.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Grant Numbers 1127609 and 1261052. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF. The authors would also like to thank Craig Arthur James of Cat-Strat (www.Cat-Strat.com) for bringing to our attention the work of Greene et al. (2011), which showed us several initial attempts at creating a collaborative problem solving system and inspired us to flesh out our ideas to construct a new model.

References

- Aho, A., Sethi, R., & Ullman, J. (1986). *Compilers: Principles, techniques, and tools*. Reading, MA: Addison Wesley.
- Chakrabarti, A., Sarkar, P., Leelavathamma, B., & Nataraju, B. S. (2005a). A behavioural model for representing biological and artificial systems for inspiring novel designs. *Proceedings of the International Conference on Engineering Design*.
- Chakrabarti, A., Sarkar, P., Leelavathamma, B., & Nataraju, B. S. (2005b). A functional representation for biomimetic and artificial inspiration of new ideas. *AIEDAM*, 19, 113–132.
- Chatterjea, R. G., & Mitra, A. (1976). A study of brainstorming. *Manas*, 23, 23–28.
- Clark, A. (1997). *Being there: Putting brain, body, and world together again*. Cambridge, MA: MIT Press.
- Diehl, M., & Stroebe, W. (1991). Productivity loss in idea-generating groups: Tracking down the blocking effects. *Journal of Personality and Social Psychology*, 61, 392–403.
- Duncker, K. (1945). On problem-solving. *Psychological monographs*, 58(5), Whole No. 270.
- Greene, K., Thomsen, D., & Michelucci, P. (2011). Explorations in massively collaborative problem solving. *Proceedings of the IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing* (pp. 716–719).

- Gurman, E. B. (1968). Creativity as a function of orientation and group participation. *Psychological Reports*, 22, 471–478.
- Hirtz, J., Stone, R., McAdams, D., Szykman, S., & Wood, K. (2002). A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13(2), 65–82.
- IHS Goldfire. (2013). Retrieved July 11, 2013, from <http://inventionmachine.com/>
- Kelley, T., & Littman, J. (2001). *The art of innovation*. New York: Doubleday.
- Linsey, J., Markman, A., & Wood, K., (2012). Design by analogy: A study of the WordTree method for problem re-representation. *ASME Transactions, Journal of Mechanical Design*, 134(4) (1 p.).
- Linsey, J., Wood, K., & Markman, A., (2008). Increasing innovation: Presentation and evaluation of the WordTree design-by-analogy method. *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conferences* (pp. 1–12).
- McCaffrey, T. (2011). *The Obscure Features Hypothesis for innovation: One key to improving human innovation*. Unpublished doctoral dissertation, University of Massachusetts Amherst.
- McCaffrey, T. (2012a). Innovation relies on the obscure: A key to overcoming the classic *functional fixedness* problem. *Psychological Science*, 23(3), 215–218.
- McCaffrey, T. (2012b). *U.S. Patent Pending No. 61/728,924*. Analogy Finder. Washington, DC: U.S. Patent and Trademark Office.
- McCaffrey, T., & Spector, L. (2011a). How the *Obscure Features Hypothesis* leads to innovation assistant software. *Proceedings of the Second International Conference on Computational Creativity* (pp. 120–122).
- McCaffrey, T., & Spector, L. (2011b). Innovation is built on the obscure: Innovation-enhancing software for uncovering the obscure. In A. K. Goel, F. Harrell, B. Magerko, Y. Nagai & J. Prophet (Eds.), *Proceedings of the 8th ACM Conference on Creativity and Cognition* (pp. 371–372). New York, NY: Association for Computing Machinery.
- McCaffrey, T., & Spector, L. (2012). Behind every innovative solution lies an obscure feature. *Knowledge Management and E-Learning*. Special issue on Supporting, Managing, & Sustaining Creativity and Cognition through Technology, 4(2), 146–156.
- McCaffrey, T., & Spector, L. (2013). An approach to human-machine collaboration in innovation. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*. Special Issue on Computational Creativity. Manuscript submitted for publication.
- McCaffrey, T., Krishnamurty, S., Grosse, I., & Wileden, J. (2013). Analogy finder: An automated tool for finding analogous solutions. *Design Studies*. Manuscript submitted for publication.
- Miller, G. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11), 39–41.
- Mullen, B., Johnson, C., & Salas, E. (1991). Productivity loss in brainstorming groups: A meta-analytic integration. *Basic and Applied Social Psychology*, 12, 3–23.
- Newell, A., & Simon, H. (1972). *Human problem solving*. Oxford England: Prentice-Hall.
- Nielsen, M. (2012). *Reinventing discovery: The new era of networked science*. Princeton, NJ: Princeton University Press.
- Oriakhi, E. V., Linsey, J. S., & Peng, X. (2011). Design-by-Analogy using the WordTree method and a WordTree generating tool. *International Conference on Engineering Design, ICED11*. Technical University of Denmark.

- Osborn, A. (1953). *Applied imagination: Principles and procedures of creative problem solving*. New York: Scribner.
- Oxford Creativity Effects Database. (2013). Retrieved July 11, 2013, from www.triz.co.uk/cp12.php
- Paulus, P. B., Larey, T. S., & Ortega, A. H. (1995). Performance and perception of brainstormers in an organizational setting. *Basic and Applied Social Psychology, 17*, 249–265.
- Renzulli, J. S., Owen, S. V., & Callahan, C. M. (1974). Fluency, flexibility, and originality as a function of group size. *Creative Behavior, 8*, 107–113.
- Rittel, H., & Webber, M. (1984). Planning problems are wicked problems. In N. Cross (Ed.), *Developments in design methodology* (pp. 135–144). New York: John Wiley & Sons.
- Simon, H. (1995). Problem forming, problem finding and problem solving in design. In A. Collen & W. Gasparski (Eds.), *Design & systems* (pp. 245–257). New Brunswick, NJ: Transaction Publishers.
- Stroebe, W., & Diehl, M. (1994). Why groups are less effective than their members: On productivity loss in idea generating groups. *European Review of Social Psychology, 5*, 271–304.
- Sutton, R. I., & Hargadon, A. (1996). Brainstorming groups in context: Effectiveness in a product design firm. *Administrative Science Quarterly, 41*, 685–718.
- Taylor, D. W., Berry, P. C., & Block, C. H. (1958). Does group participation when using brainstorming facilitate or inhibit creative thinking? *Administrative Science Quarterly, 3*, 23–47.