Distributed Analogical Idea Generation with Multiple Constraints

Lixiu Yu, Robert E. Kraut, Aniket Kittur
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{lixiuyu, robert.kraut, nkittur}@cs.cmu.edu

ABSTRACT

Previous work has shown the promise of crowdsourcing analogical idea generation, where distributing the stages of analogical processing across many people can reduce fixation, identify inspirations from more diverse domains, and lead to more creative ideas. However, prior work has only considered problems with a single constraint, while many real-world problems involve multiple constraints. This paper contributes a systematic crowdsourcing approach for eliciting multiple constraints inherent in a problem and using those constraints to find inspirations useful in solving it. To do so we identify methods to elicit useful constraints at different levels of abstraction, and empirical results that identify how the level of abstraction influences creative idea generation. Our results show that crowds find the most useful inspirations when the problem domain is represented abstractly and constraints are represented more concretely.

Author Keywords

Constraint; inspiration; problem-solving; idea generation

ACM Classification Keywords

H.5.3 Group and Organization Interfaces

INTRODUCTION

The use of analogy has historically driven innovation in science, technology, and design, in which inspirations from distant domains help problem solvers identify mechanisms that would not be apparent in the target problem's original domain [4, 5, 10, 15, 16]. For example, in 2013, a group of engineers and a world-renowned origami expert designed a large solar array to be carried by a narrow rocket. Using origami-folding techniques they were able to fold the array to a tenth of its deployed size, solving a 50-year old problem [20]. A rich literature in psychology, engineering, and design has investigated the methods by which individuals and small groups use analogies to innovate new approaches, solutions, and products [5, 6, 7, 11, 15, 17].

One thread of this research has introduced systematic methods for distributing the analogical idea generation process to crowds [28, 29]. For example, even the non-expert workers on Amazon's Mechanical Turk can search product databases (such as the thousands of unselected product submissions to Quirky) to find inspirations that contain useful mechanisms for solving a target problem in a different domain [28, 29]. A key enabling factor in this approach is abstracting the target problem and its constraints into a schema: a structured representation that removes domain-specific information while keeping the essential structure of the original problem. Such domain-agnostic representations reduce fixation on surface features, and can help focus search on inspirations that have structural similarity to the target problem.

While successful, such approaches have to date been limited by reliance on a single schema embodying a single set of constraints, i.e., requirements that a solution must satisfy to be successful. In contrast, many product design and engineering problems involve multiple, often conflicting constraints [3, 21, 23, 24, 26]. For example, the design of a kindergarten chair might require multiple constraints, such as its safety (e.g., preventing it from tipping over or pinching fingers) and flexibility (e.g., making it easy to move or stack). Each of these constraints could be represented as its own schema and at multiple levels of representation (e.g., safety vs. preventing pinched fingers). Generalizing distributed idea generation to more complex real-world problems requires a deeper understanding of how to elicit and represent multiple constraints.

In this paper, we propose a distributed process for eliciting the constraints inherent in a problem at multiple levels of abstraction and using those constraints to find inspirations useful in solving that problem. In doing so we explore how to operationalize constraints in ways that are useful for problem solving and how this can be done using non-expert crowds. We test our approach through two randomized experiments in which crowd workers generate constraints, find inspirations, and solve problems while we manipulate

CSCW '16, February 27-March 02, 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-3592-8/16/02...$15.00
DOI: http://dx.doi.org/10.1145/2818048.2835201

the level of abstraction of the problem description (i.e., domain independent or dependent descriptions) and of the constraints (i.e., more abstract or more concrete descriptions of the constraints). Our results suggest that crowd workers can identify important constraints for a product and can find inspirational examples that are useful for satisfying those constraints. The best results occur by representing the problem domain at a relatively abstract level but the constraints at a more concrete level. Under these conditions, people find the most diverse set of inspirational examples, yet ones that are most relevant to the initial problem. People shown these inspirational examples generate more original and practical ideas to solve the original problem.

LEVELS OF ABSTRACTION

Our research goal is to identify the levels of abstraction that are useful for product design in terms of both problem context (e.g., designing a kindergarten chair) and problem constraints (e.g., safety, flexibility). Abstracting a problem context into a schema in order to find analogical solutions from distant domains can be a highly effective approach for increasing innovation in domains ranging from scientific discovery to creative problem-solving to product development [5, 6, 17]. For example, people are better able to solve Duncker's radiation problem [12], in which a doctor needs to destroy a tumor with a large dose of X-rays without destroying surrounding tissue, if they are given as inspiration the case of a military general who conquers a castle surrounded by mined roads by dividing his army into small groups that are light enough to evade the mines and who attack from different directions. However, this inspiration is most useful if the problem solvers represent the solution abstractly, in terms of a divide-and-converge strategy, which maps the high-level structure shared by the problem and solution description [13]. Even though the medical and military domains differ radically in surface detail, an abstract divide-and-converge schema maps the general's strategy of dividing troops in transit into small groups before converging at the castle with the tumor problem, where the doctor divides the x-ray into multiple small beams converging on the tumor. Although this example was taken from cognitive science research, both empirical observation and experimental studies in product development find a positive relationship between using analogical inspiration and the originality of the product concept [e.g., 17, 22]. Recently, Yu et al. [29] developed an approach to distributing the schema abstraction process across multiple individuals, showing that searching for inspirations and generating ideas with crowds who had not seen the original problem domain or description decreased fixation and increased solution creativity.

However, the level of abstraction of the problem constraints is also likely to affect the diversity and relevance of inspirations found. Searching for inspirations relevant to "safety" might generate more diverse inspirational examples than "preventing pinched fingers", but might also lead to solutions less applicable to the design of kindergarten chairs. Real world idea generation often requires solving "wicked problems" [2], which can involve satisfying multiple constraints simultaneously, rather than developing a single insight that characterizes Duncker's x-ray problem. Stripping away too much detail during abstraction may cause problem solvers to consider mechanisms that are not relevant to their problem or prevent them from recognizing the precise mechanisms they need to solve a constraint in a particular domain. For example, in Duncker's radiation problem, the relevance of solutions is constrained because x-rays travel in straight lines; analogies from ant colonies involving ants who can follow arbitrary paths might thus not be relevant to reasoning about x-rays. Even though abstracting away problem context allows problem solvers to identify a wider range of potential solutions, actually applying the solutions to the original problem domain may require more detailed knowledge of the constraints that must be met.

Thus we hypothesize that while describing a problem context may be most useful if the domain or context is abstracted away (to reduce functional fixation), the representation of constraints may be most useful when they are concrete enough to suggest specific potential mechanisms (e.g., "prevent tipping over" rather than "safety") while simultaneously being general enough to allow multiple possible mechanisms (e.g., preventing something from tipping over by attaching it to a stable object or introducing a low center of gravity). To test these hypotheses, we first developed a procedure to elicit constraints from crowd workers for designing a kindergarten chair. Using these constraints, we conducted an experiment in which participants found inspirations using either an abstract or concrete representation of the problem context and constraints (in a 2x2 design), and then subsequent participants used these inspirations to generate new solutions to the design problem. Below we describe how we elicited constraints using crowds, and then describe the experiment procedure and results.

ELICITING CONSTRAINTS

One key question we aim to address here is how to elicit and represent constraints in a way that can be used by non-expert crowd workers to effectively search for inspirations, which in turn would be useful in solving a target problem. We conducted a series of pilot studies asking workers from Amazon's Mechanical Turk [18] to generate constraints for designing a good kindergarten chair, using the probe question: "*We are planning to design creative chairs for kindergarten kids. What aspects or constraints should be considered?*" The responses we got back varied significantly in their level of detail. Some of them were quite abstract, such as "*Safety. As we all know kids is very fragile so the safety is the most important thing to have*". Others were much more concrete, such as "*chairs should have to be shockproof*". We found two prevalent types of constraints, which corresponded to the level of abstraction workers used to describe the constraint. Abstract constraints such as "they should be safe for kids to sit on", can be satisfied in many different ways. In contrast, more concrete constraints, such as adding stabilizers, round off sharp corners, implied a much smaller number of ways they could be satisfied.

Based on these findings, we iteratively developed a process to elicit abstract and concrete design constraints (see Table 1). To elicit abstract constraints, we first recruited 27 crowd workers to brainstorm constraints for kindergarten chairs. As noted previously, their responses varied in abstractness. To make the response set more homogeneous and abstract, we asked a new group of crowd workers to summarize each response in a word or two: "*Which design aspect of the chair is the following suggestion referring to? Summarize it in one or two words*". We then treated these keywords as the abstract constraints, because they shifted the representation of the initially brainstormed constraints to a more abstract level. After a consolidation step, in which the experimenters eliminated duplicates and synonyms, there were seven abstract constraints: safety, flexibility, pedagogy aesthetics, cost, comfort, and ease of cleaning.

To elicit concrete constraints, we asked seven new groups of roughly 15 crowd workers each to generate concrete constraints for each abstract constraint, by framing them in terms of lower-level requirements. In this task, the crowd workers received one of the abstract constraint key words, such as safety, and described components of safety or another abstract constraint should be considered when designing a kindergarten chairs. The instructions were to eliciting more concrete version of the safety constraint were, "*We are planning to design kindergarten chairs for 5-8 year-old children. In this task, we would like you to brainstorm the constraints kindergarten chairs should meet. Please identify a constraint by filling out the following sentences. Note that constraints are requirements on certain dimensions rather than solutions.*

*Regarding safety, a kindergarten chair should meet the requirement that*____________".

This task returned a set of lower-level, more concrete constraints for each abstract constraint. For example, for the safety constraint, participants identified concrete constraints such as "chairs shouldn't tip over easily" and "chairs shouldn't have sharp edges". While more concrete than general categorical constraints such as safety, such constraints still support being instantiated in multiple ways; for example, a chair could be prevented from tipping over by lowering its center of gravity or fixing it to the floor. As each group described similar concrete constraints in somewhat different ways, we presented the lower-level constraints for an abstract constraint to a new group of crowd workers and asked them to select the ones that were similar and summarize them in terms of a single constraint.

Steps	Constraint examples
Step 1: Eliciting abstract constraints	1) chairs shouldn't hurt kids; 2) chairs should have to be shockproof → Safety 1) comfortable, because kids need to be healthy; 2) Cozy, we all like cozy furniture → Comfort
Step 2: Eliciting concrete constraints	Safety → 1) The chair should not have sharp edges; 2) The chair won't tip over Comfort →1) The chair should be easy for kids to get on and off; 2) The chair should have appropriate angle giving the correct strain free posture
Domain-independent condition	Safety: 1) The *object* should not have sharp edges; 2) The *object* won't tip over Comfort: 1) The *object* should be easy to get on and off; 2) The *object* should have appropriate angle giving the correct strain free posture.

Table 1. Workflow for generating constraints.

In the domain-independent condition we identified a list of domain related terms such as "chair", "kindergarten", "kids", and "classroom". The experimenters removed these words and replaced them with more general, domain-independent terms, such as "objects".

EXPERIMENT 1: SEARCHING FOR INSPIRATION

The previous section described a process through which crowd workers identified a variety of important constraints for a product category at multiple levels of abstraction. Moreover, simple transformations, in particular, substituting domain-specific vocabulary with more general terms, provided a method to express constraints independent of the problem domain. We used these more or less abstract constraints presented with and without reference to the problem domain as inputs for the experiment described below.

The goal of this experiment was to better understand conditions under which abstract representations of problems aid or hinder people's ability to identify solutions to them. We distinguish between the functional requirements of a problem or its constraints, on one hand, and the problem

Constraints / Domain	Abstract	Concrete
Abstract	Domain-independent abstract constraint	Domain-independent concrete constraint
Concrete	Domain-dependent abstract constraint	Domain-dependent concrete constraint

Table 2. Conditions in Experiment 1.

context or domain, on the other. We hypothesize that abstracting away the problem domain will improve search for useful inspiration by preventing functional fixation, leading searchers to find a richer and more diverse set of potentially relevant solutions. However, we hypothesize that concrete representations of constraints will be more useful than abstract representations, because they help searchers more completely understand the type of problem they are trying to solve and to evaluate the relevance of potential solutions for that problem.

To test these hypotheses, we designed a 2X2 experiment with the four conditions shown in Table 2. In Experiment 1, searchers were given descriptions of the kindergarten chair design problem varying in abstractness and domain specificity and asked to find examples that could inspire solutions. The description of the constraints that solutions needed to meet were presented either more abstractly (e.g., ways to make objects safe) or concretely (e.g., ways to make objects that won't tip over). Similarly, the description of the problem domain was also presented relatively abstractly (i.e., in domain independent manner, without mentioning kindergarten chairs) or more concretely (i.e., in a domain dependent manner, mentioning kindergarten chairs). Henceforth, laid out in Table 2, for clarity we refer to the abstractness of constraints using the terms "abstract" and concrete, and refer to the abstractness of domains using the terms "domain-independent" to refer to more abstract domain and the term "domain-dependent" to refer to the more concrete domains.

Participants

Overall, 158 Amazon Mechanical Turk workers participated in the experiment. Forty-nine percent were women, and 91% were native English speakers. Their average age was 32 and ranged from 18 to 58.

Design and Procedure

After participants accepted the task, they were randomly assigned to one of the four experimental conditions in Table 2. They were asked to search for an inspirational source for product design tasks. The task charged them with satisfying two abstract constraints, safety and flexibility, or four concrete constraints: two related to safety (Tipping over, Pinching fingers) and the other two related to flexibility (Moving around, Folding).

The task in the *domain-dependent abstract constraint* condition was described as designing a kindergarten chair's that was either safe or flexible. For example, the instructions for this task with the safety constraint was. "*We are looking for inspirations for novel and useful designs for kindergarten chairs that are safe for kids to use. Go to the Internet and retrieve a picture or description of something that has been designed to be safe to use and that could be relevant to the design of kindergarten chairs.*"

The task in the *domain-dependent concrete constraint* condition was described as designing a kindergarten chair in terms of one of the following four concrete constraints (tipping over, pinching fingers, moving around, or folding). For example, "*Go to the Internet and retrieve a picture or description of something that has been designed to easily move around and that could be relevant to the design of kindergarten chairs.*"

Note that participants in all domain-dependent conditions were explicitly instructed to not restrict their search to the world of kids' chairs, and that their results would be judged partially on novelty: "*Please search broadly. The examples can come from the domains that are outside of kids chairs. Your search results will be judged on whether it can inspire novel and useful design.*"

In the domain-independent conditions, no mention was made of kindergarten chair. Specifically, we replaced domain related terms such as "chair" or "kids" with more general terms such as 'people" or "objects". For example, instructions in the *domain-independent abstract constraint* condition were, "*We are looking for inspirations for novel and useful designs for objects that are safe to use. Go to the Internet and retrieve a picture or description of something that has been designed to be safe to use*". The instructions in the *domain-independent concrete constraint* conditions were, *"Go to the Internet and retrieve a picture or description of something that has been designed to prevent tipping over. Please search broadly. The examples can come from any domains. Your search results will be judged on whether it can inspire novel and useful design."*

Participants were asked to provide a link to the inspiration they found, an explanation why they selected this inspirational example, and a description of how they searched for it. Participants returned roughly 40 examples in each condition.

Measuring Distance and Diversity

To measure the distance between the original problem domain of kindergarten chairs to the examples searchers returned, two judges blind to experimental condition rated each example on a 7-point Likert scale: "*How different is the above kindergarten chair design problem from the domain of the example?"* By this metric, for example, a glass was considered further from a kindergarten chair than was a bench.

The judges achieved high inter-rater reliability (Intraclass Correlation Coefficient (ICC)=0.85) [9]. The final distance score was calculated by averaging the scores of the two judges. Because the distribution of the scores was bimodal, we converted distance from the kindergarten-chair domain into a binary variable based on a median split. Examples above the median score were considered "far" (1) and those below the median were considered "near" (0).

Domain-dependent, abstract constraint	Domain-dependent, concrete constraint	Domain-independent, abstract constraint	Domain-independent, concrete constraint

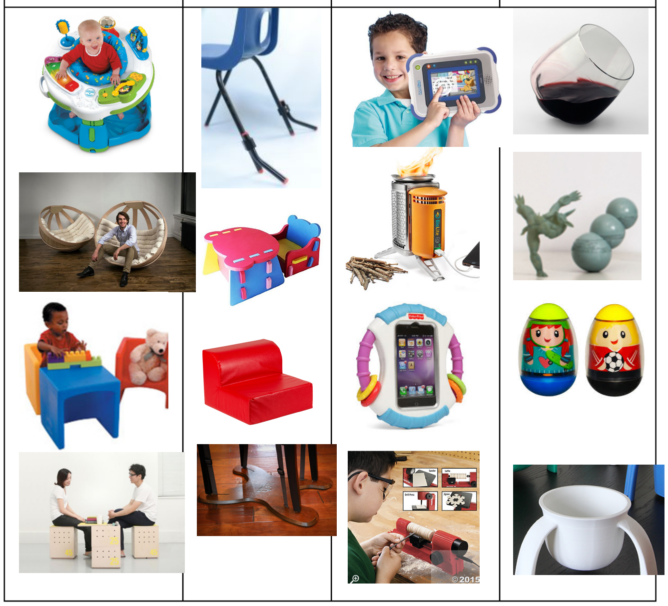


Table 3. Examples of inspirations found when the domains and constraints were presented concretely and abstractly.

We also calculated a diversity score by counting how many unique domains were found within in each condition. In counting unique domains we combined those from highly similar inspirations; for example, all the examples shown in the first column of Table 3 were classified as coming from the chair domain while the last two examples from the fourth column, (the Russian eggs and the cup), where from two different domains.

Analysis and Results

Summary statistics for the data are shown in Table 4. For analysis purposes, the four conditions were represented as two dummy variables: *domain abstractness* and *constraint concreteness*. Domain independent descriptions were coded as abstract (1) while domain-dependent descriptions were coded as 0. Problem descriptions with constraints described at a less abstractness level (e.g., tipping over, pinching fingers, moving around, and folding up) were coded as having concrete constraints (1), while descriptions with more abstract constraints (e.g., safety or flexibility) were coded as 0. We ran a logistic regression analysis with binary *Distance* as the dependent variable regressed on *domain abstractness*, *constraint concreteness*, and their interaction. The results show the odds of finding an example from a far domain were substantially higher when the domain was described abstractly rather than concretely (odd ratio = 214.86, $p<0.001$). This is consistent with previous literature showing that representing a problem abstractly reduces fixation and increases the likelihood of considering far domains. There was no significant effect of *constraint concreteness* on *domain distance* ($b=1.20$, $p=0.15$), nor a significant interaction effect ($b=-1.66$, $p=0.14$).

Conditions	**N**	**Avg Distance**	**% far**	**% unique domains**
Domain-dependent, abstract constraint	40	2.36	5%	30%
Domain-dependent, concrete constraint	40	2.69	15%	35%
Domain-independent, abstract constraint	37	5.80	92%	92%
Domain-independent, concrete constraint	41	6.20	88%	71%

Table 4. Distance of inspirations from the kindergarten furniture domain and their diversity in Experiment 1.

For domain diversity, we see a similar main effect of domain abstractness: the odds of an example coming from a unique domain were substantially higher when problems were described in an abstract, domain independent way (odds ratio=26.31, $p<0.001$). In addition, we also observe a significant negative interaction between domain abstractness and constraint concreteness, with the effect of domain abstractness significantly higher when constraints were also presented abstractly ($b=-1.77$, $p<0.05$). These results indicate that *Domain-independent abstract constraint* lead to the highest diversity of domains explored.

These results are visually reflected in Table 3, which shows examples of inspirations found in the four conditions. As these examples illustrate, in the domain-dependent conditions, most examples were about chairs and furniture. In contrast, the domain-independent conditions returned a wider variety of potential examples, including glasses, 3D printer techniques, and kids tools.

We also analyzed how participants searched for inspirations in the different conditions. In the domain-dependent conditions, most search involved domain terms. For example, participants described the ways they searched for examples, *"I was searching google for "fun kids chairs"", "creative seating for classrooms",* and *"search for tip proof chair"*. When participants searched with *domain-independent abstract constraints*, the ways of conducting search were vague. For example, "*I have search on Google for different kind of images related to novel inspiration for kids*" and "*I did an image search for adaptable images for inspiration*". In contrast, when people searched with domain-independent concrete constraints, the search terms were much more concrete while without the fixation on chairs, including "*I found the example by typing designs for objects that don't pinch fingers*", and "*I googled objects that don't tip over easily*". These different ways of searching resulted from different representations explained the findings.

Table 5. Typical chairs used in judging originality.

EXPERIMENT 2: GENERATING CREATIVE SOLUTIONS

The results from Experiment 1 suggest that the level of abstraction of the problem domain is a key driver of exploration: the more abstract the domain description, the more distant and diverse the domains explored. However, simply exploring more distant and diverse domains is not sufficient to find useful inspirations; to be useful, the solutions suggested by the inspirations must also be relevant and applicable to the target domain (here, kindergarten chairs). Observation of the rightmost two columns of Table 3 suggests that while both domain-independent conditions led to inspirations in distant domains, those found in the abstract-constraint condition (third column) have various safety designs that do not seem relevant to chair safety. In contrast, the inspirations found with concrete constraints (fourth column) suggest a variety of relevant mechanisms to prevent tipping over.

We designed an experiment to test the usefulness of the inspirations in generating creative solutions to the target problem. In the experiment, we asked people to design kindergarten chairs after showing them four examples randomly selected from one of the four conditions in the previous experiment. There were four concrete constraints (Tipping over, Pinching fingers, Moving around, Folding) in the concrete-constraint conditions; therefore, we randomly selected one example from each constraint. Because there were only two constraints (safety and flexibility) in the abstract-constraint conditions, we selected two examples from each constraint. Thus all conditions received four examples split amongst the two abstract constraints, with a further split for the concrete-constraint conditions.

Participants

Overall, 147 Amazon Mechanical Turk workers participated in the experiment. Forty-one percent were women, and 91% were native English speakers. Their average age was 33 and ranged from 19 to 83.

Design and Procedure

The same four conditions in Table 2 were used in this experiment, with the addition of a baseline control in which participants were not shown any examples. Participants first saw the instructions given to Turkers searching for inspirations in the domain-dependent, concrete constraint condition in Experiment 1: "*This task asks you to design a flexible and safe kindergarten chair. The chair should be flexible enough to be used and stored in a variety of ways in a kindergarten class. It should also be safe for kids (4-5 year olds) to use. Specifically, a flexible and safe kindergarten chair should meet the following criteria:*

1) The chair can be easily folded up; 2) The chair can be moved around easily by 4-5 year olds; 3) The chair is stable and won't tip over easily; 4) The chair won't pinch kids' fingers."

After seeing the instruction, participants saw four examples and were asked to think about how these examples could be valuable: "*In a previous task, we asked other Turkers to search for examples that could provide inspiration for designing such a kindergarten chair. Below are four examples and the Turkers' explanations for why they provided them. Please look at these examples carefully and explain how the ideas from this example could help design a kindergarten chair.*"

Participants were then asked to "*sketch the design of a flexible and safe kindergarten chair. Use the above examples as inspiration if you think they are helpful*".

Rating

Did the examples found with different representations influence the quality of the new designs differently? We draw on previous research on operationalizing measures of creativity, which suggests that ideas are more creative the more original and practical they are [1, 8, 25, 27]. Two judges, the first author and an undergraduate student, blind to experimental condition rated each design on Likert scales measuring originality and practicality. Although neither judge was a professional designer, because the design challenge was about a consumer product we believe non-experts have sufficient knowledge to evaluate the novelty and practicality of the solutions. Prior work also shows high agreement between designers and non-designers in judging consumer products such as chair design [e.g., 30, page 50-51]. To achieve more objective judgment, we considered the requirements of kindergarten chairs, and, based on these requirements, created specific criteria for judging originality and practicality.

Originality was defined as an idea that was not obvious and differed from existing products on the market. To judge originality, we selected four most frequently found chair examples in the previous experiment as the typical chair design. The raters were asked to judge originality on 7-point Likert scales by comparing the designs to these four typical chairs, as shown in Table 5. Practicality was defined as how much the new design met the four constraint criteria and how realistic it would be to design, manufacture, and use the idea. To measure practicality we used a combination of five three-point Likert scales judging each of the four constraints and the realism constraint described above. Below shows examples of the specific instructions:

Tipping over: how much does the design solve the tipping over issue? *Not at all (1), sort of (2), completely (3)*

Realism: Can it be realistically designed, produced, and *used? Not at all (1), sort of (2), completely (3)*

The final practicality score was calculated by adding all the five scores together, creating a 15-point scale. After several rounds of training and discussion, the judges achieved ICC inter-rater reliabilities of 0.75 and 0.79 for originality and practicality respectively. The designs in Table 6 show two examples with their corresponding scores on practicality and originality. Because large differences in participants' sketching skills could distort judges' estimates of idea quality, the judges were told to use the sketches to assist understanding the design idea but not to base their judgments on sketch quality.

In the experiment, participants were asked to explain how their ideas were inspired by the examples if they were so inspired. We used the information to judge whether a new design was inspired by an inspirational example. It was coded as 1 if the participant mentioned an inspiration; otherwise it was coded as 0.

Analysis and Results

The five conditions differ in terms of whether showing examples and the sources of the shown examples. We examined *practicality* and *originality* of the new designs as well as whether a new design was inspired by the examples shown to them, *inspiration*. The means, standard deviations of practicality and originality, and the percentage of designs that were inspired by an example are shown in Table 7.

The distribution of the *originality* scores was left skewed. We performed a square root transformation to normalize the data and used the transformed data for analysis. We first compared all the experimental conditions to the baseline condition on *practicality* and *originality* separately through two regression analyses. The results show that only the ideas produced by the *Domain-independent concrete constraint* condition had significantly higher *practicality* and *originality* than the baseline condition, as shown in Table 7.

To further examine the factors affecting the quality of ideas, we applied a similar dummy coding as in the previous experiment. Regression analyses, regressed on *domain abstractness*, *constraint concreteness*, and their interaction, showed that *domain abstractness* and *constraint concreteness* alone did not predict *practicality* (b=-0.64, p=0.19 for *domain abstractness*; b=-0.14, p=0.78 for *constraint concreteness*) nor *originality* (b=0.02, p=0.89 for *domain abstractness*; b=0.07, p=0.53 for *constraint concreteness*). However, we observed a significant positive interaction between *domain abstractness* and *constraint concreteness* for both *practicality* (b=3.10, p<0.01) and *originality* (b=0.38, p<0.05), as shown in Figure 1 and 2. A logistic regression analysis examining *inspiration* revealed a similar finding: while *domain abstractness* and *constraint concreteness* alone did not predict *practicality* (b=-0.80, p=0.14 for *domain abstractness*; b=-0.25, p=0.65 for *constraint concreteness*) nor *originality* (b=0.02, p=0.89 for *domain abstractness*; b=0.07, p=0.53 for *constraint concreteness*), we observed a significant positive

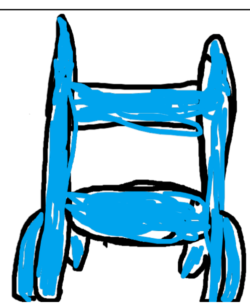
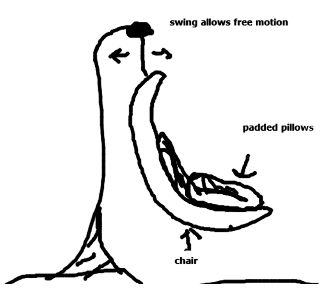
	The chair is made out of flexible material. The back part of the chair is made out of a soft cotton material and so is the cushion for where you sit. There is an addition of two more legs outside of the chair. It can easily be folded upright thanks to the chair being out of a more flexible material. The joints of the chair each have one of the protective joint covers mentioned by the other Turkers that prevents fingers from being pinched as the chair folds down. It was inspired by the first image, the second and the last because it seemed to make the most sense (Practicality: 13; Originality: 5) 
	The slide example reminded me how much fun I use to have as a child playing on the swing set. So I thought, "wouldn't it be great to have a swing chair?" The chair would be big enough for one child to sit and there would be enough soft pillow/padding in it to be very comfortable. The swing would be designed to allow enough freedom of movement to get a swinging motion but limited so that it doesn't swing to wide or to high. The chair ergonomics would function like a cradle and if desired a child could sleep in it like a mini hammock. (Practicality: 10; Originality: 7) 

Table 6 Experiment 2: Design examples

interaction between *domain abstractness* and *constraint concreteness* (b=1.95, p<0.01).

Overall these results suggest creative ideas are most likely to be inspired by examples found using an abstract problem context (i.e., removing domain specific information) but constraints that are more concrete and suggest mechanisms by which their requirements can be met (e.g., prevent from tipping over) rather than more abstract requirements (e.g., safety). The examples in Table 6 offer some insights into these findings. In the first example, the participant borrowed three mechanisms from three of the four examples: the extra supporting legs for stability (the first image), the buffer on the joint (the second image), and the folding method (the last image). These mechanisms can solve three constraints: tipping over, pinching fingers, and folding. Such successful transfer not only resulted in a practical design but also increased originality, because these mechanisms may have been widely used in the diverse domains people explored, but they were rarely seen in kindergarten chairs. In contrast, examples found in the domain-dependent conditions have no such benefits because most of them belong to the chair domains; examples found with domain-independent abstract constraints are not as useful because many of their mechanisms are not applicable to chairs.

In Experiment 2, we were also interested in the integration of multiple constraints. Applying multiple examples to different constraints is challenging for several reasons. One is that some constraints conflict with each other. Once you adapt a mechanism from one inspiration, it might become difficult to integrate a mechanism from another. For example, to prevent chairs from pinching, the chair can be designed without joints, but this design could make it difficult to integrate mechanisms involving folding, which are useful for satisfying the storage constraint. A second explanation for the difficulty in incorporating mechanisms from multiple inspirations is high cognitive load and limited working memory. Indeed, in Experiment 2, we found that even though some participants could successfully apply mechanisms from all four examples, the majority of participants used fewer than four, as illustrated in Table 6. We will return to this observation in the Discussion.

DISCUSSION

In this paper we examined how to extend previous crowdsourcing idea generation research beyond problems involving a single constraint to those involving multiple constraints. In particular, we contributed a process for eliciting multiple constraints for a problem and investigated how the level of abstraction of those constraints influenced creative idea generation. During the first step of the process, crowd workers transformed an ill-formed, open-ended problem (e.g., design a creative kindergarten chair) to a better-structured statement comprising concrete constraints (e.g., design a chair that is easily movable, stackable and won't tip over). Other crowd workers searched for inspirational examples of products in remote domains that could satisfy the constraints (e.g., mechanisms that prevent objects from tipping over). Yet other crowd

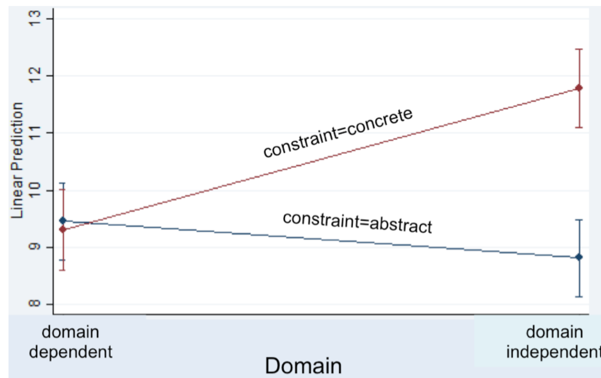


Figure 1 Practicality: Interaction between abstraction of the domain and constraints, with 95% CIs.

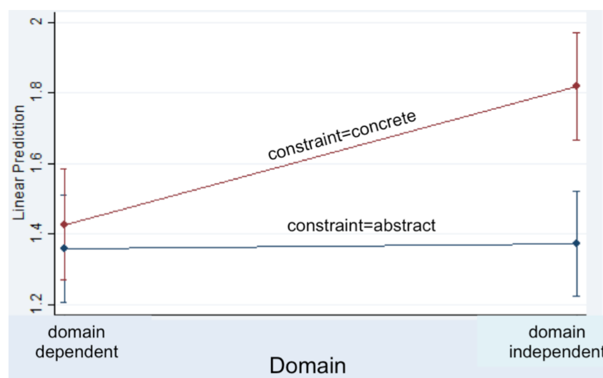


Figure 2 Originality: Interaction between abstraction of the domain and constraints, with 95% CIs.

Conditions	Practicality		Originality		% Inspiration	N
	Mean	SD	Mean	SD		
Baseline (no examples)	9.70	1.79	2.04	1.49		23
Domain-dependent, abstract constraint	9.45	1.78	2.03	1.17	26%	31
Domain-dependent, concrete constraint	9.31	2.22	2.20	1.32	31%	29
Domain-independent, abstract constraint	8.81	1.49	2.06	1.32	44%	32
Domain-independent, concrete constraint	11.77***	2.10	3.48***	1.61	71%**	31

p < .01 = **, p < .001 = ***

Table 7. The effects of abstractness of examples on design quality in Experiment 2.

workers were able to draw inspiration from these examples to generate creative conceptual designs for chairs that met these constraints. Their product designs were more creative (i.e., more practical and original) if the search for inspirations started with a problem representation that abstracted away the problem domain (e.g., kindergarten chairs), but kept the constraints concrete, compared to searches that started with a representation that mentioned the problem (e.g., kindergarten chairs) or that abstracted the nature of the constraints.

The contributions of this paper take a step towards moving distributed analogical idea generation in the direction of more complex, real-world design problems that often involve multiple constraints. However, an important open question is the extent to which more complex real-world problem-solving challenges, such as in developing a car or a mobile phone, would work with this approach. Furthermore, although we demonstrated this process with non-experts, there may be significant benefits in moving beyond the relatively naïve problem solvers (i.e., untrained Mechanical Turk workers) used here to trained problem solvers and domain experts (e.g., engineers). Some aspects of the current process may strongly benefit from employing more sophisticated workers at different stages. For example, it may be more useful to elicit constraints from domain experts (e.g., teachers, parents or children for the kindergarten chair challenge), as in traditional requirements analysis, to recruit experts in other fields for the search step to scour their domain archives for relevant mechanism, and to employ traditional product designers in the final, integration steps.

We consider our research only a first step. Additional research is needed at both the theoretical and practical levels. Progress in theory development requires crisper ways to decompose the nature of a problem. Our research differentiated the problem context (kindergarten chair versus other problems) from problem constraints (e.g., flexibility and safety), but we suspect this decomposition is too crude. For example, constraints themselves are likely to differ on multiple dimensions, such as their concreteness (which we examined in the current research), the degree to which they are integral to the overall problem or optional, their breadth of applicability and their complexity. These attributes are likely to influence the most effective way to communicate constraints to problem solvers and the degree to which analogical transfer will be a fruitful approach for finding mechanisms to satisfy them.

At the methodological level, we need to develop more robust ways to elicit constraints and requirements from analysts. Although requirements analysis is a key feature of HCI practice, we believe there is no single best way to elicit and present requirements in such as way to be useful to problem solvers.

At the application level, we believe the most interesting research involves both understanding and developing processes to help problem solvers to integrate multiple constraints. Having multiple constraints creates two challenges during integration: the large amount of information to process and the possible conflict between constraints. For example, a kindergarten chair may have dozens of constraints and people may be able to identify multiple analogs (examples of methods for satisfying each constraint), some of which will conflict with each other. For example, to make a chair easy to clean, workers may find analogs like a kitchen blender that uses a "disassembly" schema, so that parts from a bulky object can be removed and cleaned separately. However, applying this to a chair (e.g., by designing detachable legs) could undermine its stability and provide gaps that could catch fingers. In the research presented here, the people who searched for inspiration received each constrain separately, with integration deferred to people in the problem-solving stage who needed to integrate multiple mechanisms. However, this approach is not necessarily the best one.

This points to a fundamental tradeoff in distributed analogical processing between taking advantage of decomposing and distributing a problem across many people (which can improve the capacity and diversity of exploration) and dealing with the lack of context of other parts of the process that such decomposition incurs. Studying the conditions under which this tradeoff is optimized for design problems with multiple constraints would be fruitful follow-up research.

ACKOWLEDGEMENTS

This work was supported by NSF grants IIS-1526665, IIS-1149797, IIS-1217559, OCI-0943148, IIS-0968484, IIS-1111124, Bosch, Google, and Microsoft, Heinz College Center for the Future of Work, and Carnegie Mellon's Center for the Future of Work. We thank reviewers for useful feedback.

REFERENCES

1. Teresa Amabile. 1996. *Creativity in Context*. Westview Press.
2. Richard Buchanan. 1992. Wicked problems in design thinking. *Design issues*, 5-21.
3. Balakrishnan Chandrasekaran. 1990. Design problem solving: A task analysis. *AI magazine* 11, 4: 59.
4. Joel Chan, Susannah B.F. Paletz, and Christian D. Schunn. 2012. Analogy as a strategy for supporting complex problem solving under uncertainty. *Memory & cognition* 40, 8: 1352–1365.
5. Darren W. Dahl and Page Moreau. 2002. The Influence and Value of Analogical Thinking during New Product Ideation. *Journal of Marketing Research* 39, 1: 47–60.
6. Kevin Dunbar and Isabelle Blanchette. 2001. The in vivo/in vitro approach to cognition: The case of analogy. *Trends in cognitive sciences* 5, 8: 334–339.
7. Karl Duncker and Lynne S. Lees. 1945. On problem-solving. *Psychological monographs* 58, 5: i-113.

8. Ronald A. Finke, Thomas B. Ward, and Steven M. Smith. 1992. *Creative cognition: Theory, research, and applications*. The MIT Press.
9. R.A. Fisher. 1954. *Statistical methods for research workers.* Oliver and Boyd.
10. Dedre Gentner. 2002. Analogy in scientific discovery: The case of Johannes Kepler. In *Model-Based Reasoning*. Springer, 21–39.
11. Dedre Gentner, Sarah Brem, Ron Ferguson, Philip Wolff, Arthur B. Markman, and K. D. Forbus. 1997. Analogy and creativity in the works of Johannes Kepler. *Creative thought: An investigation of conceptual structures and processes*, 403–459.
12. Mary L. Gick and Keith J. Holyoak. 1980. Analogical problem solving. *Cognitive psychology* 12, 3: 306–355.
13. Mary L. Gick and Keith J. Holyoak. 1983. Schema induction and analogical transfer. *Cognitive psychology* 15, 1: 1–38.
14. Ashok K. Goel and Sambasiva R. Bhatta. 2004. Use of design patterns in analogy-based design. *Advanced Engineering Informatics* 18, 2: 85–94.
15. Andrew Hargadon and Robert I. Sutton. 1997. Technology brokering and innovation in a product development firm. *Administrative science quarterly*, 716–749.
16. Mary B. Hesse. 1966. *Models and analogies in science*. University of Notre Dame Press, Notre Dame, IN.
17. J. S. Linsey, A. B. Markman, and K. L. Wood. 2012. Design by analogy: A study of the WordTree method for problem re-representation. *Journal of Mechanical Design* 134, 4: 041009.
18. Winter Mason and Siddharth Suri. 2012. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior research methods* 44, 1, 1–23.
19. Scarlett R Miller, Brian P Bailey. 2014. Searching for inspiration: an in-depth look at designers' example finding practices. *ASME 2014 International Design Engineering Technical Conference and Computers and Information in Engineering Conference*.
20. Meg Monk. (2013, December). BYU engineers use origami to make more space in space. The Digital Universe, Retrieved from http://universe.byu.edu/2013/12/12/byu-engineers-use-origami-to-make-more-space-in-space/
21. Donald A. Norman. 2002. *The design of everyday things*. Basic books.
22. Tony Poze. 1983. Analogical connections—The essence of creativity. *The Journal of creative behavior* 17, 4, 240–258.
23. Tom Ritchey. 2006. Problem Structuring Using Computer-Aided Morphological Analysis. *The Journal of the Operational Research Society* 57, 7, 792–801.
24. David Serrano. 1987. Constraint management in conceptual design. Retrieved January 13, 2015 from http://dspace.mit.edu/handle/1721.1/14689
25. Jami J. Shah, Steve M. Smith, and Noe Vargas-Hernande. Metrics for measuring ideation effectiveness. *Design studies 24*, 2 (2003), 111–134.
26. Patrick C. Shih, Gina Venolia, and Gary M. Olson. 2011. Brainstorming under constraints: why software developers brainstorm in groups. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction*, British Computer Society, 74–83. Retrieved January 13, 2015 from http://dl.acm.org/citation.cfm?id=2305331
27. Robert J. Sternberg. 1998. *Handbook of creativity*. Cambridge University Press.
28. Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2014. Searching for analogical ideas with crowds. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (CHI'14), ACM, 1225–1234. http://dl.acm.org/citation.cfm?id=2557378
29. Lixiu Yu, Robert E. Kraut, and Aniket Kittur. 2014. Distributed analogical idea generation: innovating with crowds. In *Proceedings of the ACM conference on human factors in computing systems* (CHI'14), ACM, 1245-1254. http://dx.doi.org/10.1145/2556288.25573781
30. Lixiu Yu. 2012. *Crowd Idea Generation*. Ph.D Dissertation. Stevens Institute of Technology, Hoboken, NJ.