Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Surprise-Triggered Reformulation of Design Goals

Kazjon Grace and Mary Lou Maher
University of North Carolina at Charlotte
{k.grace,m.maher}@uncc.edu

Abstract

This paper presents a cognitive model of goal formulation in designing that is triggered by surprise. Cognitive system approaches to design synthesis focus on generating alternative designs in response to design goals or requirements. Few existing systems provide models for how goals change during designing, a hallmark of creative design in humans. In this paper we present models of surprise and reformulation as metacognitive processes that transform design goals in order to explore surprising regions of a design search space. The model provides a system with specific goals for exploratory behaviour, whereas previous systems have modelled exploration and novelty-seeking abstractly. We use observed designs to construct a probabilistic model that represents expectations about the design domain, and then reason about the unexpectedness of new designs with that model. We implement our model in the domain of culinary creativity, and demonstrate how the cognitive behaviors of surprise and problem reformulation can be incorporated into design reasoning.

Introduction

Designing presents unique challenges for cognitive systems: problems are ill-defined, solution spaces are very large, and there is often an unstated goal that new designs are expected to be creative. Creativity implies that a new design not just be useful and/or appropriate, but novel as well (Newell, Shaw, and Simon 1959; Boden 2003). Evaluating novelty in creativity requires more than a distance measure in an objective representation space. Novel designs induce surprise in observers given their past experiences with the design domain (Grace et al. 2015; 2014). Therefore evaluating creativity is dependent on dynamic and complex contextual factors, including the set of known past solutions to similar problems. Recognising the creativity of a design is increasingly a problem approachable by data science – by extracting knowledge from large volumes of product data and using it in the evaluation of novelty and value.

Cognitive models of creative designing that have inspired computational systems include: analogical reasoning, case-based reasoning, bio-inspired design, and evolutionary design (Gero 1994; Gero and Maher 1993; Goel and Pirolli 1992; Maher and Poon 1996; Maher et al. 1995; Goel et al. 2014; Helms and Goel 2012; Wiltgen, Goel, and Vattam 2011; Goel et al. 2012). These various models are primarily concerned with knowledge representation and generalized processes for design synthesis. While creating and pursuing design goals is a critical part of the synthesis of potentially creative design solutions, there remains a challenge of how to model the formation of new goals.

Cognitive studies of human creative design suggest that problem decomposition, goal formulation and solution search do not happen discretely and sequentially, but iteratively interact as designers re-interpret, re-formulate and solve problems (Schön 1983; Getzels and Csikszentmihalyi 1976). One documented trigger of this iterative reformulation is unexpected discovery – the observation of a whole or partial design of low prior likelihood (Suwa, Gero, and Purcell). The discovery of unexpected elements within a design has been shown to lead to the invention of new design requirements, which in turn lead to an increased chance of future unexpected discovery (Suwa, Gero, and Purcell 2000). In human designers, the ability to surprise oneself with intermediate external representations (e.g. sketches) is not only possible but highly desirable for creativity. We present a computational model in which observing surprising designs leads to reformulation of design goals. We do this by adopting a metacognitive perspective, in which designing occurs at the cognitive level and surprise-triggered reformulation at the metacognitive level.

Background

AI approaches to design have been inspired by cognitive models of human designers. Each of these models provides a representation of design knowledge that has the potential to generate creative designs, for example, by bringing knowledge from another domain (bio-inspired design, cross domain analogy) or by systematically combining components in a design space (i.e. the space of all possible designs) guided by a fitness function (evolutionary design). However, these models lack specificity in how to formulate new goals in response to the search in the solutions space that can lead to creative design solutions. In this section we review aspects of cognitive systems that are the basis for our more complete cognitive model that includes a metacognitive model of goal reformulation for creative design.

Metacognition in Design

Metacognition is a concept that proposes cognition about cognition, capturing that aspect of reasoning that is aware of its own processes. A hallmark of creative design is the metacognitive ability to reframe problems in ways that make surprising solutions accessible. This problem (re-)framing, referred to as reflection-in-action (Schön 1983), is an enduring challenge in the computational and cognitive modeling of design processes. Designers seek unintended consequences early in the design process by externalizing and re-perceiving their partially formed ideas and concepts (Schön and Wiggins 1992). Surprise occurs when a practitioner's grounded, experiential knowledge about performing design in this domain fails, forcing an in-the-moment process of task reformulation driven by the unexpected perception.

We build on the three-level notion of meta-reasoning from Cox & Raja (2011), in which a cognitive system can be divided into the object level, the reasoning level, and the meta-reasoning level. In designing, the object space is essentially the space of possible designs; the reasoning space is the processes associated with design synthesis; and the meta-reasoning is reasoning about synthesis.

Reasoning about Goals

Generally speaking, a goal is a desired state. In AI, goals are representational structures that are used to guide problem solving and to test if a specific process has been successful or completed. AI approaches to cognitive systems typically assume explicit representations of goals, and treat them as integral to problem solving and as a focus of attention. Adaptive goal reasoning (Aha 2015) is the study of models for generating, prioritising and selecting goals.

Dignum and Conte (1998) state that truly autonomous, intelligent agents must be capable of creating new goals as well as dropping goals as conditions change. They distinguish between abstract, high-level goals and concrete, achievable goals. They describe goal formation as a process of deriving concrete, achievable goals (e.g. 'driving at the speed limit') from high level, abstract goals (e.g. 'being good'). Abstract goals are difficult to formalise because of the challenge of representing objectives such as being good or being creative. Other approaches use models of motivation to take the place of abstract learning goals (Merrick and Maher 2009; Chentanez, Barto, and Singh 2004; Kaplan and Oudeyer 2003; Schmidhuber 1991). Computational models of motivation have also been proposed as an approach to embedding implicit motives in artificial agents to create agents with different preferences for certain kinds of activities (Merrick and Shafi 2011; 2013).

Vattam et al (2013) describe self-motivation and anomaly detection as two triggers for goal formulation, both of which we adopt in our surprise-triggered model. In our model surprises trigger specific exploratory behaviour rather than avoidance of the triggering anomaly. The drive for specific exploration occurs in human cognition as the distinction between specific and general curiosity (Berlyne 1960). In our context, a design goal is a function over a design (that is, over a state in the design search space) that yields a quantity to be maximised. The Synthesis section below gives examples, separating design goals into "novelty" and "value".

Computational Unexpectedness and Surprise

Operationalising surprise is a key step towards cognitively plausible models of novelty for evaluating creativity. Models of novelty based on distance in a representation space do not effectively capture the notion of novelty as it applies to design domains (Grace et al. 2014), and we adopt surprise as a measure of an observer's response to novel or unexpected stimuli. (Itti and Baldi 2004) define a Bayesian measure of surprise as as the degree of change in an observer's beliefs caused by an observation. The unit they give for this is the "wow", a two-fold variance in the prior probability of a model (in our case the marginal probability of a single feature in the expectation model) and its posterior probability given some observed data (in our case probability of that same feature given a certain context). This measure gives results in bits, but "wow" is used to distinguish its belief-centric nature from data-centric Shannon entropy.

For the purposes of surprise-triggered reformulation we must also know the cause of surprise, so as to influence future designing. Itti and Baldi's (2004) model-centric definition is insufficient for our purposes as it cannot easily be localised to particular input features. They define surprise as the K-L divergence between the expectation model's parameters before and after observing a new design, and in our model the mapping between hidden variables and design features may be complex and nonlinear. Baldi and Itti (2010) do provide a means of localising surprising stimuli by training models (and calculating surprise) for individual features separately, but we do not wish to impose such a per-feature training constraint upon our expectation models. We evaluate surprising combinations of features by the information that a partial observation of a design provides to another feature of that design. This allows us to denote a single feature as surprising to the model in the context of other features.

A metacognitive model of surprise-triggered reformulation

Grace and Maher (2015b) present a metacognitive framework for surprise-triggered reformulation in a design reasoning system (shown in Figure 1), adapted from the framework for metacognitive reasoning in Cox and Raja (2011). Typically, cognitive models of design include Synthesis as a goal directed process that perceives and acts on a design space, that is, a space of possible and existing designs. In our cognitive model of design we include a process we call Expectation, which learns from the existing designs in the design space, and builds models of expectation that explicitly guide the Synthesis process in a large search space of existing and possible designs. Surprise-triggered reformulation occurs as a metacognitive process of reflecting on expectations, reformulating goals that guiding synthesis towards the part of the design space that triggered surprise. As in humans the model's design process is iterative, with synthesis producing designs that affect future synthesis through expectation, surprise and reformulation. In this paper we present a computational model of this framework and provide proof-of-concept implementations of its core components.

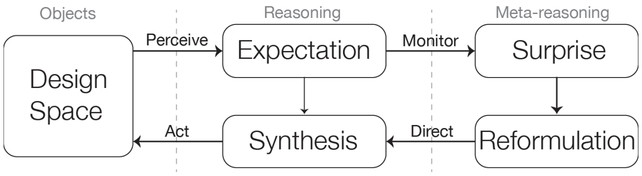

Figure 1: A cognitive model of surprise-triggered reformulation in designing (Grace and Maher 2015b), showing levels of reasoning as per Cox and Raja (2011).

Design Space

The design space demarcates the domain (i.e. the space of products that perform compatible functions – a necessarily subjective definition) in which the design reasoning system is operating, and comprises the object level of our computational model. The design space consists of a set of design features F, the powerset of which is the set of all possible designs $D = 2^F$. This formalism considers only discrete design features for notational simplicity, but could be extended to continuous features. A subset of possible designs $O \subset D$ have been observed, either created by the system through its synthesis process or created by others. Each possible design $d \in D$ contains a subset of all possible features: $d \subset F$.

Design space knowledge acquisition can be accomplished in several ways. Knowledge, which is captured in the expectation process's generative model, can be provided before use, or learnt from observed designs. The model could also support systems that explore the design space through synthesis alone, with O being initially empty and thereafter containing only the system's creations.

Expectation

The expectation process models the system's beliefs about the design space that can be used to synthesise and evaluate new designs. Expectation builds the system's beliefs about the design space as a probabilistic generative model over possible designs $P_\Theta(D)$. Our approach is compatible with any parameterisation, Θ, of this model, so long as it can be learnt from what is known about the design space (in our experiments we use a corpus of designs). The structure and parameters of this model, including any hidden variables, are implementation specific. We treat each feature $f \in F$ as a discrete random variable, and the relationships between features are modelled by the joint distribution of this model. This distribution is used in both the evaluation of surprise to recognise the unexpected features of designs and in the synthesis process to generate new designs.

Synthesis

We define synthesis to encompass all design reasoning from a set of design goals to generate one or more designs, a broad scope that allows our computational model to encapsulate a variety of approaches to design reasoning. Our model's purpose is to produce creative designs, for which systems possess a set of value goals G_v and a set of novelty goals G_n, both of which must be met for a design to be considered creative. For example, an automated recipe design system may have a value goal specific to cake baking that maximises the similarity between cake sweetness (evaluated by some flavour model) and a "target sweetness" in the knowledge base. In an online recipe system, a value goal may be to maximise the number of "likes" a recipe receives. In most design reasoning systems novelty goals are typically abstract, such as "maximise the distance between the new design and the nearest cluster of existing designs" or "minimise the likelihood of the design under a probabilistic model derived from domain knowledge" (Grace et al. 2014). These general novelty goals are supplemented in our system by goals formed as a result of surprise-triggered reformulation. Synthesis is a function from the current set of goals and the expectation model to C, a set of candidate new designs:

$$S(G_v, G_n, P_\Theta(D)) = C \subset D$$

This function, as with the expectation model, is provided as a "black box" compatible with many synthesis approaches, of which one example is given in this paper.

Surprise

Surprise occurs when a design, in whole or in part, is highly unexpected given the probabilistic model that comprises the expectation component of the system. This can result from evaluating a new design while searching the design space, but also from observing a design created by another (i.e. a point in the design space visited by another system and shared with this one). Under Itti and Baldi's (2004) surprise measure (modified as specified in the Background), a design feature that halved in a-priori likelihood when some other features were observed would have a surprise of 1 bit, or 1 wow. Here we extend that definition from the surprisingness of an individual feature to the surprisingness of a design. A surprising design, d, possesses one or more *surprising combinations*, $s(f, c)$: a single *surprising feature*, $f \in d$, and a *surprise context*, $c \in 2^d$, which consists of a set of features all observed in the design. The surprising feature is of significantly lower expected likelihood (under the expectation model) when the surprise context is present than if it were not. The evaluation of the surprise of a design is therefore the search for the most surprising $s(f, c)$ pairs:

$$s(d) = \arg\max_{f,c} s(f|c), \forall f \in d, \forall c \in 2^d$$

This involves a search of the powerset of the set of all features in a design, 2^d, for surprise contexts, testing each context against all other potentially surprising features. Exhaustive search is likely to only be computationally feasible in domains where the feature representation is sparse, such as recipes. See the Implementation section for a discussion of an approximate solution to the search component of evaluating the surprise of a recipe. With this we can evaluate an individual surprising combination $s(f|c)$ given $P_\Theta(f)$ and $P_\Theta(f|c)$ (from the expectation model) as:

$$s(f|c) = -log_2(P_\Theta(f)/P_\Theta(f|c))$$

assuming that $c \subset d$, $f \in d$, $f \notin c$, and $c \neq \emptyset$. In other words, $s(f|c)$ is the overall information content of f minus the information content of f when all features in c are also present, measured in bits. In the special case where $c = \emptyset$ we construct surprise differently:

$$s(f|\emptyset) = -log_2(P_\Theta(f)) - (\sum_{n=1}^{|d|} log_2(P_\Theta(f_n))/n$$

or the unconditional probability of f' minus the average unconditional probability of all other features. Surprise given an empty context can be considered the novelty of the feature f (Grace et al. 2014).

Reformulation

The purpose of surprise-triggered reformulation is to induce temporary novelty goals that will bias the search for designs towards those that are surprising in a similar way to the triggering stimulus. Our model defines reformulation as a function from a surprising combination to a new novelty goal: $R(s(f|c)) = G_{n'}$. This new goal consists of a weighting over variables in the expectation model that bias the original novelty goal G_n. This new weighted novelty function will rate the triggering combination at least as surprising as the default novelty goal G_n, and rate causes of surprise involving unrelated variables much less surprising. The specifics of this function from surprising combination to weighting over expectation model variables (both hidden and visible) are a matter of implementation, although various heuristics for surprise-triggered reformulation have been proposed (Grace and Maher 2015a; 2015b).

Implementation

As a proof-of-concept evaluation of our model for surprise-triggered goal reformulation we have implemented it in the context of recipe design, a popular domain for computational design and creativity (Varshney et al. 2013; Morris et al. 2012).

Design Space

We gathered approximately 130,000 recipes from `http://ffts.com/recipes`, an archive of public domain recipes designed to work with desktop recipe database software. We utilise a "bag of ingredients" approach to representing recipes, with each recipe consisting only of the set of ingredients that it contains, ignoring amounts, preparation, categories, etc. We do not present this representation as optimal for design reasoning in the culinary arts, but instead seek to establish a simple proof-of-concept of the surprise-triggered reformulation model. We removed ingredients present in less than 0.1% of the dataset, and then removed recipes with less than three ingredients, resulting in a final dataset of approximately 100,000 recipes containing 321 unique ingredients. There is insufficient data to infer the likely correlates of these rare ingredients, and therefore they cannot lead to confident expectations: they are novel, but not surprising (see Grace et al 2014 for a discussion of the difference).

Expectation Implementation

In our implementation we wrap the expectation process around a Variational Autoencoder (VAE) (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014), a deep neural network-based generative model that learns the hidden variables of a Bayes Net from nonlinear combinations of the inputs. Our implementation of expectation converts the set representation of each design to a binary vector of length $|F|$, capturing the presence or absence of each possible feature. The distribution over possible states of this vector becomes the parameterisation of our generative model $P_\Theta(D)$, and is learnt from the recipe corpus.

We found that the initialisation radius of the neurons in the encoder network needed to be increased significantly (to values around 0.1) to avoid the local minimum in which the latent variables z were always near 0 and the decoder network learnt only the individual probabilities of each feature. In this case the poor predictive performance of the network is offset by the K-L divergence term, which at this minimum is near 0. We hypothesise that this is likely to occur for any similar architecture trained on sparse feature vectors.

Surprise Implementation

In calculating surprise we estimate $P_\Theta(f|c)$ by repeatedly sampling from the VAE posterior conditioned on the surprise context c using the "missing data imputation" method described in Rezende et al. (2014). This method requires iteratively sampling from the VAE, making exhaustively evaluating $P_\Theta(f|c), \forall f \in d, \forall c \in 2^d, f \notin c$ computationally inefficient. We first applied a maximum context size in the range of 2-5 to limit search depth. Instituting a depth limit on the search is particularly apt in the case of recipes, where most interesting flavour combinations contain just a few ingredients, but may not be appropriate in other domains. We applied beam search, finding a beam width equal to $|d|$ multiplied by the depth limit to produce a good tradeoff between accuracy and speed. Beam search ignores features for which $P_\Theta(f|c) \approx P_\Theta(f)$, assuming that if f and c are independent, $f \cup c$ will not be a more insightful surprise context than c. In other words, the beam search avoids branches of the search tree that contain unrelated ingredients (i.e. those that do not affect each others' likelihoods).

For example, a recipe for chocolate bacon cupcakes might involve sugar, butter, flour, bacon, cocoa powder, salt, eggs and baking powder. Under the model, $P_\Theta(bacon|butter, eggs) > P_\Theta(bacon)$, and $P_\Theta(sugar|flour, eggs) > P_\Theta(sugar)$, making those highly unsurprising combinations. However, $P_\Theta(bacon|sugar, butter, cocoa) \ll P_\Theta(bacon)$, giving that combination 10.8 wows (see Background) of surprise and putting it in the 97^{th} percentile among the dataset.

Reformulation Implementation

We compare three approaches to reformulation, all based on the notion of finding other recipes that are similarly surprising to a trigger. This occurs by replacing the synthesis process's general novelty goal with a specific one. The first and simplest is *same combination* reformulation, which induces a goal for the presence of the exact surprising combination from the recipe that triggered reformulation. The second approach is *same feature* reformulation, which searches for recipes that find the same ingredient surprising, but in any context. The third is *same context* reformulation, which searches for recipes that find ingredient(s) surprising given the presence of some of the same ingredients that triggered the initial surprise.

Continuing the above cupcake example, the system would attempt to create new novelty goals from the surprising recipe, seeking other recipes that contained bacon, sugar, butter and cocoa (same combination), other recipes in which bacon was surprising (same feature), or other recipes in which something was surprising given sugar, butter and cocoa (same context).

Synthesis Implementation

The synthesis process of our implementation is trivial: we sample from the VAE (which is an approximate generative model of $P_\Theta(D)$, the joint probability of the design space) using the surprise (of the maximally surprising combination in the design) in wows as G_n. This is a minimal implementation for the purposes of demonstrating goal reformulation. Naïve sampling from a probability distribution learned from data is not presented as a replacement for design reasoning, merely as a generative mechanism by which to demonstrate the effects of surprise on designing. We do not measure the "value" component of creativity in this implementation, except in that it is indirectly captured by sampling from the expectation model that has been trained on archived recipes of presumably high quality. Recipe value could be estimated from user preferences in online recipe-sharing communities, or from a purpose-built flavour model, but our goal is only to demonstrate reformulation. Table 1 shows examples of recipes generated according to the novelty goals induced from the cupcake example above, along with an interpretation of the recipe as a dish by the experimenters.

In the rye bread example the original surprising combination was still the most unexpected. In the cheesy pasta dish bacon was highly surprising (14 wows) in the context of dill, pecan and vodka. The sweet potato casserole (which appears to be a variant on a dish commonly associated with Thanksgiving in the USA) both sweet potatoes and coconut were surprising ($>$9 wows) given the contextual target of the cocoa, butter and sugar. These results demonstrate how reformulation can influence search, as none of these combinations appear in the training dataset and are all highly unlikely to appear by random sampling alone. The interpretations are provided by the authors (who also selected each example from the top three most surprising of 100 generated designs), but a much more knowledge-rich system could infer such interpretations itself. It is possible for this process to return recipes found in the database, but this did not occur in our sample.

Comb.	Rye flour, flour, coffee, water, salt, caraway seeds, yeast, molasses, *cocoa, sugar, butter* and **bacon**. *Interpretation*: Rye Bread with coffee/bacon.
Feat.	Pasta, *pecans*, cheese, eggs, parmesan, butter, *dill, vodka*, white wine, salt, black pepper, and **bacon**. *Interpretation*: Rich and cheesy pecan-bacon pasta.
Ctxt.	**Sweet potatoes**, **coconut**, eggs, flour, vanilla, anise, cinnamon, *cocoa, butter and sugar*. *Interpretation*: Coconut-chocolate sweet potato casserole.

Table 1: Recipes generated under the effect of the three goal reformulation types as triggered by $s(bacon|sugar, butter, cocoa)$: Same combination (Comb.), same feature (Feat.) and same context (Ctxt). In each case the maximally surprising feature is bolded and its surprise context italicised.

Simulations

We conducted a series of simulations with our prototype surprise-triggered reformulation system, focussing on demonstrating its capacity to generate goals that can drive the design process in interesting (and potentially creative) directions. As an example, in the *same combination* recipe in Table 1, a new novelty goal is formulated to synthesise recipes containing bacon and sugar, butter, and cocoa. In our simulations the Expectation process builds a probabilistic model of the design space of recipes, the Surprise process triggers the reformulation of novelty goals based on highly surprising recipes from the dataset, and the Synthesis process searches for designs using the reformulated goal set.

We randomly sampled 1000 recipes from the database and ranked them in order of their maximally surprising combination. Table 2 shows two examples from the most surprising recipes thus evaluated, each with results generated by them acting as trigger for the three types of reformulation. All surprise evaluations (including those in the bacon cupcake example above) were performed with 10^5 samples, a scale necessary for reliably estimating scores of 13-15 wows, the maximum estimable from a database of 10^5 designs.

Discussion

We have developed, implemented and evaluated a cognitive model of how surprise affects design reasoning. Surprise triggers the reformulation of design goals to focus the synthesis process on exploring specific features of highly novel designs. We demonstrated that computational measures of surprise can be used to guide computational design reasoning towards creative designs. Our cognitive model includes metacognitive processes inspired by concepts such as specific curiosity (Berlyne 1960), in which exploratory behaviour can have specific, rather than diversive, goals. Our model extends the idea of an explore/exploit tradeoff to the design reasoning context, unifying it with the novelty/value approach to evaluating creativity. This forms an approach to design synthesis based on reasoning about spe-

Recipe:	*"Dilly zucchini ricotta muffins"*: *zucchini*, *ricotta*, sugar, baking powder, *margarine*, eggs, flour, zucchini, **dill**, salt, and milk. (14.5 wows)
Comb.	Eggplant, spinach, cottage cheese, **dill**, *margarine, ricotta* and *zucchini*. *Interpretation*: Spinach/ricotta stuffed eggplant. (14.5 wows)
Feat.	Broth, water, *sherry, lemon, cashews*, onions, **dill**, parsley, salt, black pepper. *Interpretation*: Lemon cashew soup. (8.5 wows)
Ctxt.	Pasta, *ricotta, zucchini*, tomatoes, potatoes, eggs, garlic, oregano, salt, basil, black pepper, **balsamic vinegar,** parmesan, and *margarine*. *Interpretation*: Vegetarian lasagne. (4.1 wows)
Recipe:	*"Lime velvet salad"*: Jell-o, water, mixed nuts, *cream cheese, whipped cream, cherries*, and **celery**. (13.4 wows)
Comb.	Cake mix, pineapple, eggs, pecans, **celery**, *whipped cream, cream cheese, cherries*, spices. *Interpretation*: Pineapple celery cake. (13.4 wows)
Feat.	Chicken, breadcrumbs, butter, apples, **celery**, *brandy*, parsnips. *Interpretation*: Fried chicken with apple/parsnip salad. (6.5 wows)
Ctxt.	Cream cheese, *whipped cream*, **crackers**, sugar, blueberries, lemon, oranges, *cherries*. *Interpretation*: Fruit cheesecake. (3.4 wows).

Table 2: Highly surprising recipes from the database shown with generated recipes after each of the three reformulation types. Format as per Table 1.

cific novelty goals formulated in response to surprising existing designs, in contrast to the more general goal of generating any design that is novel. This approach is enabled by a machine learning-driven expectation model that is data driven and capable of forming beliefs about the design domain over time. We use a knowledge-lean generative process as a proxy for design synthesis, and show that the surprise-triggered novelty goals were able to influence synthesis towards surprising-yet-plausible recipes.

Our contribution is a model for creating specific goals for exploratory behaviour in design reasoning. Our model for reasoning about goals differs from Baranes and Oudeyer's (2010) "maturationally-constrained self-adaptive goal generation" approach: their model progressively relaxes constraints on learning in order to continuously promote exploration, while our model formulates new goals. In their model, general relaxation occurs when progress towards goals slows, in contrast to our model where specific goals are formulated as a contextual response to surprising observations. Maher et al. (2015) propose that curiosity is a kind of reasoning in the absence of goals, while we consider curiosity a metacognitive process that can lead to new goals through surprise-triggered reformulation. Our formulation of specific goals is as a focus for the synthesis process, similar to Foner and Maes' (1994) work on agent models of attention. Foner and Maes (1994) distinguish between goal-driven and world-driven focus, where the former is defined before problem solving starts, and the latter allows reasoning to respond to information from the world. Our model uses information about the world (that is, observed designs in the design space) to trigger the formulation of new goals and thus further goal-driven behaviour.

In summary, we contribute a cognitive model for goal reasoning in a design system that addresses the challenge of formulating specific goals that lead to creative designs. We distinguish between reasoning about design synthesis and meta-reasoning about goal formulation. Our meta-cognitive processes of surprise and reformulation reason about expectations: a probabilistic model of existing and possible designs constructed by our expectation process. We have demonstrated a proof-of-concept of our model's Synthesis, Expectation, Surprise and Reformulation processes in the design space of recipes. This implementation provides evidence that goal formulation can guide a synthesis process towards creative designs. The broader implications of this approach include: (1) the ability to build design systems that can be surprised by their own synthesis processes as well as by observation of new designs produced by others, and (2) the ability to distinguish between goal-directed reasoning about design synthesis and meta-reasoning that can recognise when those goals should change.

Acknowledgements

We are grateful for the assistance of Stephanie Grace and Marlene Shankar in performing this work, the latter of whom was supported by the NSF (#1461166).

References

Aha, D. W. 2015. *Goal reasoning: Papers from the ACS workshop (Technical Report GT-IRIM-CR-2015-001)*. Georgia Institute of Technology.

Baldi, P., and Itti, L. 2010. Of bits and wows: a bayesian theory of surprise with applications to attention. *Neural Networks* 23(5):649–666.

Baranes, A., and Oudeyer, P.-Y. 2010. Maturationally-constrained competence-based intrinsically motivated learning. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, 197–203. IEEE.

Berlyne, D. E. 1960. *Conflict, arousal, and curiosity.* McGraw-Hill Book Company.

Boden, M. A. 2003. *The creative mind: Myths and mechanisms.* Routledge.

Chentanez, N.; Barto, A. G.; and Singh, S. P. 2004. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, 1281–1288.

Cox, M. T., and Raja, A. 2011. *Metareasoning: Thinking about thinking.* MIT Press.

Dignum, F., and Conte, R. 1998. Intentional agents and goal formation. In *Intelligent Agents IV Agent Theories, Architectures, and Languages*. Springer. 231–243.

Foner, L. N., and Maes, P. 1994. Paying attention to what's important: Using focus of attention to improve unsupervised learning. Master's thesis, Citeseer.

Gero, J. S., and Maher, M. L. 1993. Preface. In *Modeling Creativity and Knowledge-Based Creative Design*. Lawrence Erlbaum Associates, Inc.

Gero, J. S. 1994. Towards a model of exploration in computer-aided design. In *Formal design methods for CAD*, 315–336.

Getzels, J. W., and Csikszentmihalyi, M. 1976. *The creative vision: A longitudinal study of problem finding in art*. Wiley New York.

Goel, V., and Pirolli, P. 1992. The structure of design problem spaces. *Cognitive science* 16(3):395–429.

Goel, A. K.; Vattam, S.; Wiltgen, B.; and Helms, M. 2012. Cognitive, collaborative, conceptual and creative – four characteristics of the next generation of knowledge-based cad systems: a study in biologically inspired design. *Computer-Aided Design* 44(10):879–900.

Goel, A. K.; Vattam, S.; Wiltgen, B.; and Helms, M. 2014. Information-processing theories of biologically inspired design. In *Biologically Inspired Design*. Springer. 127–152.

Grace, K., and Maher, M. L. 2015a. Specific curiosity as a cause and consequence of transformational creativity. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 260.

Grace, K., and Maher, M. L. 2015b. Surprise and reformulation as meta-cognitive processes in creative design. In *Proceedings of the Third Annual Conference on Advances in Cognitive Systems ACS*.

Grace, K.; Maher, M. L.; Fisher, D.; and Brady, K. 2014. Data-intensive evaluation of design creativity using novelty, value, and surprise. *International Journal of Design Creativity and Innovation* 3(3–4):125–147.

Grace, K.; Maher, M. L.; Fisher, D.; and Brady, K. 2015. Modeling expectation for evaluating surprise in design creativity. In Gero, J., ed., *Design Computing and Cognition'14*. Springer. 189–206.

Helms, M., and Goel, A. 2012. Analogical problem evolution in biologically inspired design. In Gero, J., ed., *Design Computing and Cognition 2012*. 3–19.

Itti, L., and Baldi, P. 2004. A surprising theory of attention. In *IEEE Workshop on Applied Imagery and Pattern Recognition*, 45–54.

Kaplan, F., and Oudeyer, P.-Y. 2003. Motivational principles for visual know-how development. In *Proceedings of the 3rd International workshop in Epigenetic Robotics*, 73–80. Lund University Cognitive Studies.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Maher, M. L., and Poon, J. 1996. Modeling design exploration as co-evolution. *Computer-Aided Civil and Infrastructure Engineering* 11(3):195–209.

Maher, M.; Boulanger, S.; Poon, J.; and Gómez de Silva Garza, A. 1995. Assessing computational methods with a framework for creative design processes. In Gero, J.; Maher, M.; and Sudweeks, F., eds., *Proceedings of Computational Models of Creative Design*. 233–266.

Maher, M. L.; Merrick, K.; and Wang, B. 2015. Reasoning and making sense of data in the absence of goals. In *Goal Reasoning: Papers from the ACS Workshop*, 105–119.

Merrick, K. E., and Maher, M. L. 2009. *Motivated reinforcement learning: curious characters for multiuser games*. Springer Science & Business Media.

Merrick, K. E., and Shafi, K. 2011. Achievement, affiliation, and power: Motive profiles for artificial agents. *Adaptive Behavior* 19(1):40–62.

Merrick, K. E., and Shafi, K. 2013. A game theoretic framework for incentive-based models of intrinsic motivation in artificial systems. *Frontiers in psychology* 4.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity*, 119–125.

Newell, A.; Shaw, J.; and Simon, H. A. 1959. *The processes of creative thinking*. Rand Corporation.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Schmidhuber, J. 1991. A possibility for lmplementing curiosity and boredom in model-building neural controllers. In *Proceedings of the International Conference on Simulation of Adaptive Behaviour*. MIT Press. 222–227.

Schön, D. A., and Wiggins, G. 1992. Kinds of seeing and their functions in designing. *Design studies* 13(2):135–156.

Schön, D. A. 1983. *The reflective practitioner: How professionals think in action*. Basic books.

Suwa, M.; Gero, J.; and Purcell, T. Unexpected discoveries and s-inventions of design requirements: A key to creative designs. In Gero, J., and Maher, M., eds., *Computational Models of Creative Design IV*. University of Sydney, Sydney, Australia. 297–320.

Suwa, M.; Gero, J.; and Purcell, T. 2000. Unexpected discoveries and s-invention of design requirements: important vehicles for a design process. *Design Studies* 21(6):539–567.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schoergendorfer, A.; and Chee, Y.-M. 2013. A big data approach to computational creativity. *arXiv preprint arXiv:1311.1213*.

Vattam, S.; Klenk, M.; Molineaux, M.; and Aha, D. W. 2013. Breadth of approaches to goal reasoning: A research survey. Technical report, Naval Research Lab.

Wiltgen, B.; Goel, A. K.; and Vattam, S. 2011. Representation, indexing, and retrieval of biological cases for biologically inspired design. In *Case-Based Reasoning Research and Development*. Springer. 334–347.