

Goal-Driven Conceptual Blending: A Computational Approach for Creativity

Boyang Li, Alexander Zook, Nicholas Davis and Mark O. Riedl

School of Interactive Computing, Georgia Institute of Technology, Atlanta GA, 30308 USA
{boyangli, a.zook, ndavis35, riedl}@gatech.edu

Abstract

Conceptual blending has been proposed as a creative cognitive process, but most theories focus on the analysis of existing blends rather than mechanisms for the efficient construction of novel blends. While conceptual blending is a powerful model for creativity, there are many challenges related to the computational application of blending. Inspired by recent theoretical research, we argue that contexts and context-induced goals provide insights into algorithm design for creative systems using conceptual blending. We present two case studies of creative systems that use goals and contexts to efficiently produce novel, creative artifacts in the domains of story generation and virtual characters engaged in pretend play respectively.

Introduction

Conceptual blending has been proposed as a fundamental cognitive process, responsible for the creation of a broad range of creative artifacts. (Fauconnier and Turner 1998, 2002; Grady 2000; Hutchins 2005). Fauconnier and Turner (1998, 2002) proposed that conceptual blending involves the merger of two or more input spaces into a blended space. A griffin, for example, may be considered as a blend of an eagle and a lion. Each of the input spaces contains a number of concepts and their inter-connections. These concepts are selectively merged and projected into the blend space. After that, additional structures may emerge in the blend by pattern completion and further elaboration.

Among artifacts created by conceptual blending, we distinguish two types of blends, namely *semiotic expressions* and *standalone concepts*. Semiotic expressions are used in communication to highlight certain aspects of or shed light on one of the input spaces. An often discussed semiotic expression is "this surgeon is a butcher" (cf. Grady, Oakley, and Coulson 1999; Brandt and Brandt 2002; Veale and O'Donoghue 2000). The input spaces of the expression are the conceptual spaces of the surgeon and the butcher respectively. The surgeon in the blend possesses the brutal attitude of the butcher, forming a criticism of the surgeon.

This paper focuses on the construction of the second type of blend, which we call standalone concepts. An

example is the lightsaber from *Star Wars*: a lightsaber blends together a sword and a laser emitter, but it is an independent concept that does not inform hearers about the properties of swords or laser emitters. During blend creation, contents are still projected from the input spaces into the blend, but the blend is not meant to convey information about the input spaces.

We share the belief that theories of creativity should be computable (Johnson-Laird 2002), yet most accounts of blending focus on analyses of existing blends and have not fully described how novel blends are constructed cognitively or algorithmically. In particular, three key procedures required for blending lack sufficient details necessary for efficient computation: (1) the selection of input spaces, (2) the selective projection of elements of input spaces into the blend, and (3) the stopping criteria for blend elaboration. Inefficiencies in these procedures can lead to significant difficulties in finding appropriate blends and elaborations. For example, a simplistic algorithm may produce all possible combinations of elements from all input spaces, resulting in a combinatorial explosion of possible blends.

We argue that these three main procedures must algorithmically make use of the *context* and *goals* of the blend being constructed. Brandt and Brandt (2002) proposed communication contexts and goals as the driving force behind the three key procedures, but their analysis is limited to semiotic expressions. We extend their theory to the construction of novel standalone concepts and provide computational justifications through two case studies of working computational systems in the domains of story generation and pretend play. Our systems construct blends in a goal-driven and context-driven manner. As integral aspects of the conceptual blending process, contexts and goals provide concrete computational benefits by pruning search spaces and improving average-case performance.

The Theories of Conceptual Blending

This section reviews the theories of conceptual blending described by Fauconnier and Turner (1998, 2002) and Brandt and Brandt (2002). We compare these two accounts side by side and identify some underspecified parts in the theories, which a working system must address.

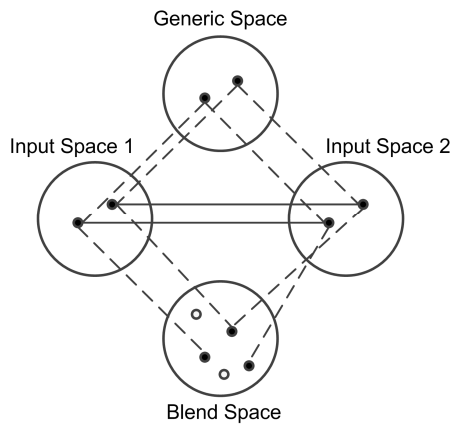


Figure 1. The four-space blending theory, adapted from Fauconnier & Turner (2002).

In the original blending theory (BT) by Fauconnier and Turner (1998, 2002), conceptual blending takes two or more mental spaces as inputs. Mental spaces are dynamically constructed during a discourse (e.g. conversation) to contain relevant concepts. The input space of the surgeon, for example, includes the surgeon and relevant entities, such as his scalpel, patient and so on. Elements in one input space are then mapped to their counterparts in another input space, using mapping rules such as identity or analogy. In the surgeon-as-butcher example, the cleaver of the butcher is mapped to the scalpel of the surgeon; the dead animal is mapped to the patient, and so forth. Elements from input spaces are selectively projected into a *blend space*. The *generic space* captures the structural similarities between input spaces. In addition to elements projected from input spaces, the blend space can also contain emergent structures created by pattern completion or elaboration. This four-space formulation is shown in Figure 1, where big circles denote mental spaces, black dots represent elements in the spaces, solid lines are the mappings between inputs, and dashed lines denote correspondences among the elements in the four spaces. Hollow dots denote emergent structures in the blend.

Fauconnier and Turner, however, did not specify how elements from these input spaces could be chosen during the selective projection. Although eight optimality principles—human scale, topology, pattern completion, integration, vital relations, unpacking, web, and relevance—were proposed as quality measures, they can only evaluate the quality of a complete blend *after* it is constructed. This suggests a computational approach where all possible blends have to be generated and tested individually, called a neo-Darwinian algorithm by Johnson-Laird (2002). A neo-Darwinian algorithm could lead to a combinational explosion of options and is infeasible for large input spaces. In contrast, what Johnson-Laird calls a neo-Lamarckian approach generates only valuable products by applying quality constraints on the search space. To do so in blending requires a mechanism

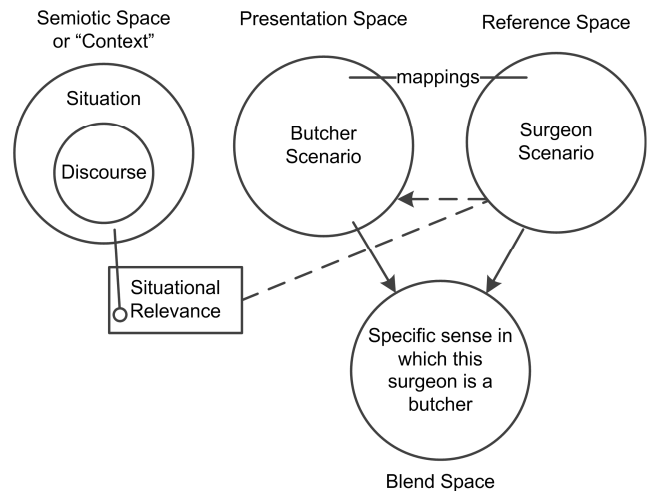


Figure 2. The context-dependent blending theory, adapted from Brandt and Brandt (2002).

that selects elements from input spaces effectively during the generation process. Note that projection occurs after inter-space mappings are built, so the complexity of analogy making should not be conflated with the complexity of projection.

Moreover, BT does not provide detailed procedures for the effective retrieval of input spaces, nor for the elaboration of blends. A full computational implementation of the blending theory should select input spaces by itself, rather than assume them as given. To create a powerful criticism of the surgeon, a system should decide to blend it with a butcher, rather than a driver or a school teacher. A creative system may possess a huge amount of knowledge, so an efficient selection procedure for input spaces is necessary for it to operate within reasonable time limits. The same argument goes for elaboration. Neither a human nor a computational system should elaborate a blend endlessly. We usually do not wish to simulate the entire world's reaction to an irresponsible surgeon, which will require excessive computational power or time.

In summary, the original blending theory left much ambiguity in three key procedures: (1) the selection of input spaces, (2) the selection of elements for projection, and (3) the stopping criterion for blend elaboration. However, a complete working implementation of BT must contain these procedures.

Context-Dependent Blending

Brandt and Brandt (2002) pointed out that blends used in communication do not have fixed meanings. Rather, they are real-world phenomena that can only be analyzed in the context of the discourse during which they were uttered. Under different circumstances, the same utterance “this surgeon is a butcher” can mean different things. For example, if a soldier referred to a battlefield medic as a butcher, he may be highlighting the fact that he has to perform an astonishing number of amputations. This

interpretation is vastly different than the ethical judgment about a surgeon's skills after a failed surgery.

To account for this plurality of meanings, blend construction must not be based solely on the attributes of the input domains, but on the context. Hence, Brandt and Brandt abandoned the context-free generic space and proposed a context-driven blend construction process. A simplified version of Brandt and Brandt's theory is shown in Figure 2, where the dashed arrow indicates that the situational relevance prompts the retrieval of the two input spaces in order and solid arrows represent selective projection. First, the communicative situation (context) retrieves the input spaces in order. The situation implies a first input space, the *reference space* based in the actual real-world situation. For example, if we are talking about a particular surgery, the input space of the surgeon becomes available. Based on the goal of communication (e.g. conveying that the surgeon is irresponsible), a *presentation space* including the figurative entities (e.g. a butcher) is then retrieved and mapped to the reference space. Second, the goal of communication, captured through the situational relevance, determines what elements from the two spaces are projected into the blend. If we want to accuse the surgeon of being irresponsible, we should project the careless attitude of the butcher into the blend. Third, elaboration of blends also depends on the context and the goal, fleshing out a blended space until sufficient detail is achieved for the guiding goal. In summary, the sequential retrieval of input spaces, the selection of projected elements, and the stopping criteria are all driven by contexts and goals.

Blending Novel Concepts

In addition to metaphorical expressions, blending also yields novel concepts independent of the input spaces, such as the lightsaber. We believe the two cases differ in the relationship between the inputs spaces and the blend.

Blends used in communication are usually meant to underscore certain aspects of, or to attach new properties to an input space. In our example, the surgeon who possesses a butcher's attitude in the blend space becomes a criticism of the surgeon in the input. The linkage between the blend and the input spaces is essential for understanding.

In contrast, standalone concepts are blends that are not meant to convey meaning about their input spaces. For example, the birth stork is a blend of the metaphor birth-is-arrival, air travel, and the stork (Fauconnier and Turner 2002, ch. 14), but it does not tell us much about air travel or storks in general. As another example, the lightsaber in *Star Wars* is clearly a blend of a sword and a laser emitter, but it is not meant to tell us anything about swords or laser, even though understanding laser and swords can help us understand lightsabers. There are links going from the input spaces to the blend spaces, but not vice versa.

Standalone concepts created from blending are commonly seen in stories and story-related activities. Dragons, for example, possess features of snakes, large cats and birds of prey, and the instinctive fear of the three

is postulated to be its origin (Jones 2002). Many other mythical creatures are combinations of common animals. Novel concepts created from blending also appear in modern science fiction. The popular Japanese sci-fi manga *Doraemon* (Fujio 1974-1996) contains several gadgets made this way, such as a toy telephone that can transmit flu instead of voice.

The idea of an independent concept does not contradict the notion of contextualized meaning. The meaning of a novel concept, when used as a semiotic sign, can still change depending on the context. One may say "my new kitchen knife is a lightsaber" to emphasize its sharpness. In Brandt and Brandt's framework, this meaning is created from the blend of the kitchen knife and the lightsaber, which is a different blend than the lightsaber itself. The concept lightsaber, on the other hand, is as independent as the concept butcher. For another example, the birth stork is now often a humorous symbol for baby births rather than used to explain births. The contexts and meanings vary, but the concepts remain relatively constant.

We extend Brandt and Brandt's contextualized blending theory to the construction of novel concepts. Below we present two computational systems that generate novel concepts in fictions and pretend play. While Brandt and Brandt's (2002) theory initially was not meant to explain such blends, we show that a goal and context driven approach can account for their generation and bring computational benefits.

Previous Implementations

Most previous work on computational algorithms for conceptual blending are based on the theories of Fauconnier and Turner and thus do not fully account for input space selection, selective projection, or blend elaboration. It is common for computational blending algorithms to ignore one or more of these stages. The Alloy blending engine, as part of the Griot poem generator (Goguen and Harrell 2004), is the earliest implementation of BT that we are aware of. Input spaces are manually coded as symbolic expressions. Projection into the blend is based on a structural mapping between input spaces. Without the guidance of goals, any element from the input spaces may be projected. The authors found the number of possible blends is exponential to the number of relations in the input spaces, but did not discuss methods to prune these spaces. Hervás et al. (2006) proposed a process that enriches texts of stories which can be considered as conceptual blends. They proposed that readers' familiarity with input spaces, as part of the communication context, is important in selecting input spaces and elements for projection without specifying an algorithm.

Martinez et al. (2011) proposed a blending algorithm where input spaces are sets of axioms. Compatible axioms are selected for projection. The system does not consider goals to directly build blends that meet specific purposes. Rather, it "enumerates alternatives ranked by the complexity of the underlying mappings". Yamada et al. (2011) generate motion of dancing characters by

representing motion as wavelet equations and blending is a weighted summation of wavelet coefficients. The process does not involve any of the three key procedures mentioned earlier. Thagard and Steward (2011) proposed an implementation of blending at the neural level, which also does not consider goals.

The Divago system (Pereira 2007) is noteworthy in that it considers goals, but only uses them indirectly during construction of standalone concepts. Note that the input spaces are given to—rather than selected by—Divago. In Divago, goals do not directly participate in the selection of elements being projected. If an element a is mapped to b , one of 4 things can be projected into the blend: a , b , a/b , or nothing. This leads to a combinatorial explosion of possible blends regardless of the complexity of finding an inter-space analogical mapping. To effectively search an exponentially growing space, Divago utilizes a genetic algorithm (GA) that stochastically samples the space of possible blends. From a population of blends, Divago first selects those with high scores as computed by an evaluation function, consisting of the weighted sum of the eight optimality criteria introduced earlier. One of the criteria, relevance, is interpreted as goal satisfaction. After evaluation, highly ranked blends are randomly modified to create the next population. Imitating biological evolution, this process repeats in the hope of finding a near-optimal blend after sufficient number of iterations. Note however that a high score does not guarantee goal satisfaction because the evaluation function contains several, possibly competing criteria.

To elaborate a blend, Divago fires any production rules whose premises are true to add content into the blend. Divago also supplements details to the blend based on its similarity with other frames. For example, if a blend is similar enough to the bird frame, Divago will grant it the ability to fly. However, given enough rules and frames, rule firing and pattern matching can potentially go on indefinitely, as it does not specify any explicit stopping criteria for elaboration based on the notions of meaningfulness or necessity.

In the light of the above analysis, it is clear that goals and context can guide a computational conceptual blending process for the purpose of creating novel, standalone concepts. However, context and goals must be used directly to ensure successful blends and to focus search efficiently. In the next section, we study two creative systems that utilize contexts and goals to effectively realize the three procedures in conceptual blending.

Two Case Studies

This section presents two systems that implement blending theory in a goal and context driven manner. We describe these systems through the lens of the context-driven blending theory. The first system builds fictional gadgets in computer-generated stories. The second system constructs objects used in pretend play that combine features of a desired fantasy-world object with a real-world object at hand. The two systems address the three

mentioned problems—input space selection, selective projection, and elaboration—in a neo-Lamarckian manner by employing constraints introduced by the domain of application and the specific goal to be achieved. The first case study focuses on selective projection and elaboration. The second case study focuses on input space selection.

Generating Gadgets in Fictions

As a vibrant research field, artificial intelligence (AI) story generation aspires to create intelligent systems that can create and tell novel stories. Most current approaches to story generation are restricted to generating stories for static, hand-authored micro-worlds, manipulating given characters and objects to produce stories (e.g. Cavazza, Charles, and Mead 2002; Gervás et al. 2005; Riedl and Young 2010; Ontañón and Zhu 2010). These systems can be likened to jigsaw puzzle solvers who play only with given pieces and never dream of inventing a new piece. These story generation systems are not able to tell stories with novel objects or gadgets; they cannot tell *Star Wars* if the idea of lightsaber is not supplied ahead of time.

The aim of the gadget generation algorithm (Li and Riedl 2011a, 2011b) is to break out of these static world configurations and create new types of objects previously unknown to the system. Our approach was initially presented as a combination of partial-order planning (Weld 1994) and analogical reasoning. Here we point out its connection to conceptual blending. The algorithm blends existing concepts to generate novel standalone concepts as unforeseen gadgets in support of a goal derived from a story context.

To generate a gadget, the algorithm reasons about how the gadget should be used in the context of a story, including events that happen immediately before and after its usage. These events are captured as the behavior of the gadget, represented as a temporally ordered sequence of actions. Given a goal derived from a story (e.g., a character must become infected by a flu virus), the algorithm iteratively constructs the gadget's behavior by working backward from the goal using actions and entities from various input spaces. Goals are first-order logic predicates such as `infected-by(bob, virus)`. An action is an operator that requires certain predicates as preconditions and asserts some predicates as effects. The gadget generator works with a conventional story generator, which supplies goals considered appropriate for a gadget to achieve. The final behavior of the gadget must achieve these goals.

Goals are first used to identify the input spaces. The reference space includes objects in the goal predicates and relevant concepts. For example, the reference space implied by the goal `infected-by(bob, virus)` includes concepts such as flu viruses, a character named Bob, and actions such as coughing and curing. In fact, the reference space exists only conceptually and is not separated from the rest of the knowledge in the system. It highlights that knowledge structures closely related to the goal play more important roles in projection than the rest. The system

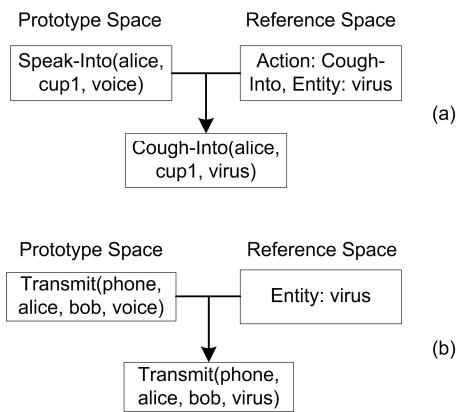


Figure 3. Two projections involved in the generation of the flu-transmitting gadget phone.

creates the second input space by retrieving an object from its knowledge base of many known objects that achieve predicates analogous to the specified goals. This object is called the prototype for the gadget. At this time, the list of known potential prototypes is small, and we go down the list from the best guess. In the long term, a robust mechanism to find the best prototype is needed (e.g. Wolverton 1994). The second input space, the prototype space, includes the prototype object, its behavior, as well as actions used and entities referred to in the behavior. For example, the behavior of a toy phone object refers to the entity *voice*, which becomes part of the space.

The behavior of the gadget, or the blend space, is built by projecting actions from the two input spaces selectively. This projection is driven by goals in a backward-chaining, iterative process. Following the partial-order planning representation, actions have causal requirements—predicates that must be true in the micro-world for the action to be performed (also called *preconditions*). When an action is brought in the blend space to solve one goal, its own causal requirements are added to the set of goals. Actions continue to be projected to satisfy goals until all goals are satisfied or determined to be fundamental properties of the gadget itself. Note that when there are multiple possible ways to achieve a goal, the algorithm tries the best first and backtracks when mistakes are made. The selective projection process is thus completely goal-driven. Due to space constraints, we refer interested reader to (Li and Riedl 2011b) for details of the algorithm.

There are several methods to project actions from the input spaces into the blend space. First, the algorithm can project an action from the prototype space without any changes. Second, an action from either space can be projected with arguments from both spaces, constituting a form of blending. Figure 3(a) shows the action *Cough-Into* being projected with arguments from both input spaces. This action is mapped with the action *Speak-Into* in the prototype space because of identified analogical similarities. The analogical reasoning engine Sapper (Veale and O'Donoghue 2000) is used to establish analogical mapping across input spaces. Third, a special

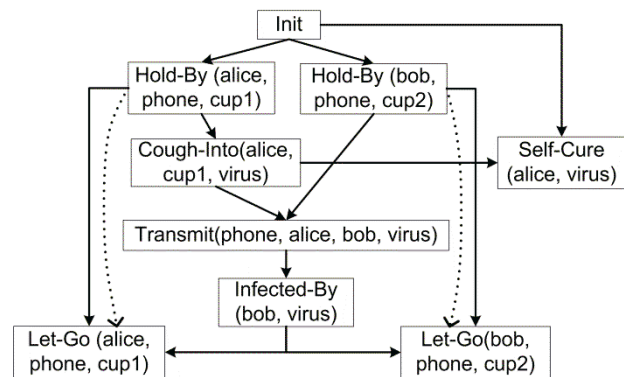


Figure 4. The behavior of the flu-transmitting gadget phone

case of blending occurs when an action from the prototype space is combined with illegal arguments from the reference space to create a new action previously not allowed. As illustrated in Figure 3(b), the *Transmit* action of a toy phone is now allowed to transmit *virus* instead of *voice*, which otherwise would not have been a legal assignment of parameters. The algorithm ignores the rules of the micro-world to achieve goals of an imaginary gadget phone that can transmit flu virus from one person to another. This allows a generated gadget to achieve the impossible, producing an object not conceived before.

Figure 4 shows the behavior of a gadget phone generated by the algorithm, which one can use to give her flu to someone else and free herself from it. The goal predicates it achieves are *infected-by (bob, virus)* and *not(infected-by (alice, virus))*. Each box represents one action. Solid arrows denote temporal precedence and dotted arrows denote closure actions. This gadget appeared in the Doraemon manga. Li and Riedl (2011a) also describes an example not from the manga.

Elaboration in the blend space is simulated with the use of closure actions. Closure actions are not necessary for the gadget's goal, but restore some goal-irrelevant aspects of the story world to an ordinary state. For example, in Figure 4, the actions where people put down the phone after use (the *Let-Gos*) are closure actions. Closure actions are manually labeled in prototype behaviors and projected into the blend space. Their use is motivated by the desire to restore an ordinary state after using the gadget, which creates a sense of denouement in the story. Although the elaboration is not directly driven by goals, it is motivated by the domain of storytelling.

Finally, the gadget is verified by incorporating the gadget behaviors into the story from which the goals were originally derived. If any necessary conditions of the gadget's behavior cannot be achieved in the story, the gadget does not work and has to be modified further by going through additional rounds of blending. When this happens, a second prototype is retrieved as yet another input space and blended with the current gadget to make it even more powerful. The algorithm cannot create two gadgets simultaneously.

By using means-ends search guided by goals, the search space of possible actions to be added to the behavior of the gadget is pruned, resulting in improved average-case efficiency of the algorithm. Each iteration of the algorithm tries to satisfy a goal in the blend space, and will thus only consider actions that can achieve that goal. While the total number of actions in both input spaces may be large, usually only a small fraction of them can be projected to achieve any given goal. In contrast, a naïve selective projection algorithm will attempt to project all actions in any input space using all possible projection methods. Although the goal-driven best-first search in the worst case has to consider all possibilities in the search space, in an average case it only considers a small portion of the total possibilities (Weld 1994). A goal-driven blending algorithm also has the added benefit of ensuring that any result of the algorithm is guaranteed to meet all of the acceptability requirements.

In summary, the gadget generation system implements a goal-driven model of blending, including relatively efficient selective projection (due to pruning of the search space) and elaboration procedures. These procedures are driven by the gadget's goal, the particular story that serves as the context, and the general domain of storytelling. However, this system has not fully investigated the selection of input spaces, which will be illustrated in the next case study.

A Virtual Character for Pretend Play

Our second case study illustrates the use of goals in blending with a specific focus on the selection of appropriate input spaces. In this case, a real-world object is selected to represent an object from a fantasy world, as required in children's pretend play. A goal specifies means to appropriately prune potential input spaces and select one option based on contextual constraints of similarity. Below we describe how our pretend play system can be viewed through the lens of conceptual blending; full details on the system are presented in (Zook, Riedl, and Magerko 2011).

In *pretend play*, children construct and enact story scripts and roles with real-life objects (Nourot 1998). Examples of pretend play include lightsaber duels with cardboard tubes, holding pretend tea parties with stuffed animal guests and acting as a group of pirates sailing on a couch. When enacting these scripts, pretenders have goals of using particular objects from a fictional world, but are limited to using the real-world objects that are ready at hand. Pretend players imaginatively overlay the fictional object onto the real-world object to create a blend, i.e. a pretend object existing in both the fictional world and the real world. Pretend play research has found children project traits of the fictional object on the real object. As an example, children engaged in a lightsaber duel with cardboard tubes may make buzzing noises when they swing the tube. In general, the pretending process involves identifying real-world objects as stand-ins for the fictional object, and selectively projecting traits of the fictional object onto the real object. The objects are input spaces to

the blend. The construction process must account for how input spaces relate to a larger context of a target pretend play activity, pruning the options considered.

Building computational systems that can engage in pretend requires the capacity to construct the objects used in these scripts (Zook, Riedl, and Magerko 2011). To formalize the problem, the play activity (lightsaber duel) provides a structuring situation such that a pretender—human or agent—selects a real world object (cardboard tube) as a presentation space for a given reference space of a fictional object (lightsaber). That is, the goal is to find a presentation input space that most closely matches the reference input space. Once the input spaces are selected, the blending process takes the most relevant aspects of the fictional object for the activity (buzzing), which are imposed onto the real object in the blend space for use in play (swinging a cardboard tube while buzzing). This process starts with a situation in the fictional world and a specific fictional object (the lightsaber I am using in the duel), and seeks a presentation object in the real world to effectively manifest the fictional object.

To reason about numerous objects in the fictional world and the real world, we need a computational representation of objects and their attributes. Lakoff and Johnson (1980) proposed that the salient perceptual, motor-activity, and purposive features of objects affect how humans interact with them. We model objects in both the fictional and real domains using selected attributes in these categories. Following prototype theory (Rosch 1978), these attributes are assigned fuzzy values to represent a real-valued ($[0, 1]$) range of degree of membership (DOM). As an example, a lightsaber may have a 0.8 DOM value for the perceptual feature of being blue (very blue, except for the handle), 0.9 DOM value for the motor-activity feature of ease of handling (very easy to hold and swing), and 0.1 DOM value for the purpose of supporting weights (unsuitable for propping up heavy objects).

Iconic attributes are salient attributes of an object that distinguish it from similar objects within the same category. These attributes help to resolve the potential ambiguity of which fictional object is being represented by a given real world object. For example, if a pretender grabs an object and begins making buzzing noises, it may be unclear if they are signaling that they are holding a buzzing lightsaber or shooting a laser pistol. An iconic posture of handling, however, makes this difference clear. Iconic attributes help participants in pretend play interpret other players' behaviors and intentions.

The computational play system algorithm has three steps for context and goal driven blending: (1) select a real-world object based on the pretending goal and context; (2) select the set of fictional object attributes to project; and (3) project these attribute values into the blend. The first step is the selection of the input space—the real-world object—to be blended with the fictional object.

Selecting an input space uses the pretending goal to first prune impossible input spaces and then search for the optimal input space among those that remain. Conceptually

this process is similar to the surface-level filtering process used in the MAC/FAC system (Forbus, Gentner and Law 1995) with the modification of using fuzzy attributes rather than predicates. The filtering quickly removes from consideration all real-world objects that differ too significantly from the desired fictional object on a single attribute. The goal specifies attributes that are important to the pretend play object and the extent to which they must be preserved. Thus, when seeking a lightsaber, all real objects that are too difficult to handle would be ignored, as they cannot serve as useful lightsabers during the play. Computationally, all real objects are compared to the desired fictional object on the set of relevant attributes, and those that differ by a specified threshold are pruned. For example, when considering ease of handling, a cardboard tube would be kept as a candidate real object for a lightsaber, while a wooden log would be discarded.

After filtering, the remaining real-world objects are searched for the single real-world object with minimum difference from the desired fictional object. Computationally, the pretend play system exhaustively searches all available real-world objects and calculates the Euclidean distances between the attributes of each candidate real object and the fictional object. The real-world object with the minimal distance from the fictional object is then selected. In this domain, pruning appears to sufficiently constrain the set of potential input spaces to enable subsequent exhaustive search, although alternative search techniques are likely applicable.

Once the input spaces are selected, the second step is to bridge the remaining distance between the real-world object and the fictional object by determining the set of iconic attributes that capture characteristic attributes of the fictional object. These will then be mapped back to the real-world object so that the agent can play with the real-world object as a placeholder for the fictional object. Iconic attributes of an object are those attributes that are most different from other objects under consideration; they capture which features are relevant to the pretend goal. The level of *iconicity* of an attribute for the desired pretend object is calculated as the sum of Euclidean distances between that attribute and the same attribute of all other objects in the fictional world. Iconicity values are normalized within categories of objects and an attribute is considered iconic for an object if it falls in the proximity of the maximum value.

In the third stage, blending occurs by projecting iconic values of the desired fictional object onto the selected real object. By default, the blend space contains all attributes of the real object. All iconic attributes of the pretend object are projected into the real object, replacing the original values. This process captures the notion that most action and reasoning should treat the real world as the base with the pretend domain layered onto this base.

In the pretend play algorithm, context and goals are utilized to filter the set of possible input spaces to only those that are most crucial for the use of a real-world object for pretend play. By pruning the set of possible

objects according to their relevance to a goal, the process avoids the naïve consideration of all possible combinations of spaces to use for blending. Selective projection of attributes is achieved by searching for the most iconic attributes of the presentation input space (the fictional object) to be blended with attributes of the reference input space (the real-world object). While this search is performed in a brute-force manner, the number of attributes that cross the acceptability threshold for iconicity are relatively limited.

Discussion and Conclusions

As a powerful mechanism for creativity, conceptual blending is capable of synthesizing known concepts into new concepts. Much existing theoretical work focuses on the blending phenomenon and identifying the input spaces and the blend without a mechanism for blending. This paper presents our first efforts at building a complete computational outline for conceptual blending systems. We identify three major procedures in conceptual blending: (1) the selection of input spaces, (2) the mechanism for selective projection of input space attributes into the blend space, and (3) the sufficiency condition for pattern completion and elaboration. These components play vital roles in conceptual blending and have significant implications for the efficiency of computation. In our analysis of computational implementations of blending theory, we found few systems fully account for all three processes.

Brandt and Brandt (2002) argued that the construction of semiotic expressions as blends are cued by communication contexts and guided by the specific communicative goal. We argue that context and goals can provide the basis for rigorous and efficient computational algorithms for the three main processes described above. We present two computational systems that utilize goals and context to guide generation of standalone conceptual blends. The gadget generation algorithm mainly demonstrates procedures (2) and (3) by utilizing goals to select concepts from the input spaces and elaborating them. The pretend play work likewise uses context to determine which input spaces to select and which concepts from the input spaces to project into the blend, illustrating procedures (1) and (2).

These two case studies suggest the three main procedures can be implemented efficiently by employing constraints introduced by their respective domain of application, the contexts of the solutions, and the specific goals the solutions must achieve. Our analysis shows that goals can be used to prune the search space and improve average-case performance. Although our implementations are deterministic, we believe determinism and goals are not a bundled package. A goal-driven procedure may not be completely deterministic or even optimal. Future work is needed to reduce the effort required to author the knowledge representations used by our systems (e.g. with crowdsourcing (Li et al. 2012)).

Boden (2004) raised the questions of whether computers can appear to be creative, and whether computational

systems can help us understand creativity. By implementing theories of creativity, we are forced to consider procedural details which theories sometimes do not cover. We believe a computational approach can help expose underspecified components or flaws in existing theories, hint at their solution, or even lead to their remedy. A computational approach to creativity will strengthen our confidence in answering yes to both of Boden's questions.

Acknowledgement

This work was supported by the National Science Foundation under Grant No. IIS-1002748. We thank Per Aage Brandt and the anonymous reviewers for valuable inputs.

References

- Boden, M. A. 2004. *The Creative Mind: Myths and mechanisms*. 2nd ed: Routledge.
- Brandt, L., and Brandt, P. A. 2002. Making Sense of a blend. *Apparatur* 4:62-71.
- Cavazza, M., Charles, F., and Mead, S. J. 2002. Planning Characters' Behavior in Interactive Storytelling. *Journal of Visualization and Computer Animation* 13 (2):121 - 131.
- Fauconnier, G., and Turner, M. 1998. Conceptual Integration Networks. *Cognitive Science* 22 (2):133-187.
- Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books.
- Forbus, K.D., Gentner, D., and Law, K. 1995. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science* 19(2).
- Fujio, F. F. 1974-1996. *Doraemon*. Vol. 1-45. Tokyo: Shogakukan.
- Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R. 2005. Story Plot Generation based on CBR. *Knowledge Based Systems*. 18.
- Goguen, J., and Harrell, F. 2004. Foundations for Active Multimedia Narrative: Semiotic spaces and structural blending. *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*.
- Grady, J. 2000. Cognitive mechanisms of conceptual integration. *Cognitive Linguistics* 11 (3/4):335-345.
- Grady, J., Oakley, T., and Coulson, S. 1999. Conceptual Blending and Metaphor. In *Metaphor in Cognitive Linguistics*, edited by R. Gibbs and G. Steen. Amsterdam: John Benjamins.
- Hervás, R., Pereira F. C., Gervás, P., Cardoso A. 2006. Cross-Domain Analogy in Automated Text Generation. In *3rd Joint Workshop on Computational Creativity at ECAI*.
- Hutchins, E. 2005. Material Anchors for Conceptual Blends. *Journal of Pragmatics* 37:1555-1577.
- Johnson-Laird, P. N. 2002. How Jazz Musicians Improvise. *Music Perception* 19 (3):415-442.
- Jones, D.E. 2002. *An Instinct for Dragons*. Routledge.
- Lakoff, G., and Johnson, M. 1980. *Metaphors We Live By*. Chicago: The University of Chicago Press.
- Li, B., and Riedl, M. O. 2011a. Creative Gadget Design in Fictions: Generalized Planning in Analogical Spaces. In *8th ACM Conference of Cognition and Creativity*.
- Li, B., and Riedl, M. O. 2011b. A Phone That Cures Your Flu: Generating Imaginary Gadgets in Fictions with Planning and Analogies. In *4th Workshop of Intelligent Narrative Technologies*. Palo Alto, CA: AAAI.
- Li, B., Lee-Urban, S., Appling, D. S., Riedl, M. 2012. Automatically Learning to Tell Stories about Social Situations from the Crowd. In *the LREC 2012 Workshop on Computational Models of Narrative*.
- Martinez, M., Besold, T., Abdel-Fattah, A., Kuehnberger, K.-U., Gust, H., Schmidt, M., and Krumnack, U. 2011. Towards a Domain-Independent Computational Framework for Theory Blending. In *The ACM Fall 2011 Symposium on Advances in Cognitive Systems*.
- Nourot, P.M. 1998. Sociodramatic play: Pretending together. In *Play from Birth to Twelve and Beyond: Contexts, perspectives, and meanings*.
- Ontañón, S., and Zhu, J. 2010. Story and Text Generation through Computational Analogy in the Riu System. In *7th AI and Interactive Digital Entertainment Conference*.
- Pereira, F. C. 2007. *Creativity and AI: A Conceptual Blending Approach*. Berlin: Mouton de Gruyter.
- Riedl, M. O., and Young, R. M. 2010. Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research* 39:217-268.
- Rosch, E. 1978. Principles of Categorization. In *Cognition and Categorization*. Lawrence Erlbaum.
- Thagard, P., and Stewart, T. C. 2011. The AHA! Experience: Creativity Through Emergent Binding in Neural Networks. *Cognitive Science* 35 (1):1-33.
- Veale, T., and O'Donoghue, D. 2000. Computation and Blending. *Cognitive Linguistics* 11:253-281.
- Weld, D. 1994. An Introduction to Least Commitment Planning. *AI Magazine* 15 (4):27-61.
- Wolverton, M. 1994. Retrieving Semantically Distant Analogies. Ph.D. Dissertation, Stanford University.
- Yamada, K., Taura, T., and Nagai, T. 2011. Design and Evaluation of Creative and Emotional Motion. In *8th ACM Conference on Creativity and Cognition*.
- Zook, A. E., Riedl, M. O., and Magerko, B. S. 2011. Understanding Human Creativity for Computational Play. In *8th ACM Conference on Creativity and Cognition*.