DRAFT

Generate and Situated Transformation as a Paradigm for Models of Computational Creativity

Nick Kelly

Australian Digital Futures Institute, University of Southern Queensland, Brisbane, Australia; and Faculty of Science and Engineering, Queensland University of Technology, Brisbane, Australia

John S Gero

Department of Computer Science and School of Architecture, University of North Carolina at Charlotte, USA; and Krasnow Institute for Advanced Study and Department of Computational Social Science, George Mason University, USA

Corresponding author: Dr Nick Kelly
Email: nick.kelly@usq.edu.au
Phone: +61 409 652 047

Abstract: This paper describes 'generate and situated transformation' as a paradigm for models of computational creativity that can lead to novel questions and methods of enquiry. It describes the need for systems that can move towards a frame within which a design solution can be found. The term *situated transformation* is defined, with reference to Boden's notion of transformation and ideas from situated cognition, as a change of frame that is based upon experience. The paper then provides a demonstration of simple systems with generate and situated transformation that embody this idea. It then elaborates upon these models showing parallels within established systems in the literature. The paper concludes with a discussion of research questions and avenues of enquiry that are apparent within this paradigm.

Keywords: computational, creativity, situated, interpretation, exploration, analogy, search, transformation, situated cognition, design

1. Introduction

There have been many attempts to develop computational models that claim to be creative (Bentley & Corne, 2002; Boden, 1991, 2009; Colton, 2012; Cope, 2005; Langley, 1987; McCorduck, 1991; Simon, 1969). Such models typically make a contribution by demonstrating a system that can produce designs that fulfil the criteria of being both useful and novel (Runco, 1991; Runco & Jaeger, 2012; Stein, 1953). These two criteria are necessary but no longer considered sufficient for creativity since "surprising" as a third criterion is increasingly being added to describe the notion that an evocation of unexpectedness is not captured by "novel" (Grace, Maher, Fisher, & Brady, 2015; Itti & Baldi, 2006). This paper is concerned with the paradigms within which such models come to be developed. It is not concerned with describing any one computational model of creativity. In particular, the paper aims to articulate the paradigm of *generate and situated transformation* which, we claim, is useful for describing existing models of creativity and for the development of future models of creativity.

A *paradigm* is defined as "a set of assumptions, concepts, values and practices that constitutes a way of viewing reality for the community that shares them, especially in an intellectual discipline" (American Heritage Dictionary of the English Language, 2000). The term paradigm, with this meaning, was used by Thomas Kuhn to describe the way that scientific revolutions occurred through a series of paradigm changes (Kuhn, 1962). The development of theories and models occurs within a paradigm (e.g. Newtonian physics) until a subsequent paradigm replaces it (e.g. Einstein's relativity). Where others describe science as a progression towards an objective truth, Kuhn describes a sequence of paradigms each of which have implicit assumptions, concepts, values and practices that only allow some types of truths to be established. Pragmatically, the utility of a paradigm lies in its facility to create avenues of inquiry (what is studied and researched), to formulate questions (what gets asked) and to select methods with which to carry out enquiry (how results are interpreted) (Kuhn, 1962). This paper sets out to describe the paradigm of generate and situated transformation on the basis that it is useful for the development of models of computational creativity.

The core tenet of models that use generate and situated transformation is that they are able to change the space within which they are searching and that the way in which this is done matters. It is a well-established and demonstrated idea that computational models can both *search* spaces of design and *transform* the space being searched during design (Boden, 1991; Wiggins, 2006b). However within the literature this distinction between searching and moving to a different space is not yet well defined (Wiggins, 2006a): why, for example, can a movement from one space to another not simply be labelled as search within the larger space containing both of them? The distinction between search and transformation is the foundation for the paradigm of generate and situated transformation. This paradigm is then described through examples that demonstrate its utility. Many existing models of computational creativity can be understood from within this paradigm. The claim is not that the paradigm is new, but rather that articulating the paradigm makes a useful contribution to future research by opening up novel and useful avenues of inquiry, questions and methods.

2. Background

2.1 Reframing in design activity

Design can be considered to occur within a *frame* of understanding, where the word frame is used in the designerly sense[1] to refer to a cognitive construct that gives a perspective that allows interpretation and influences subsequent judgement (Dorst, 2015; Schön, 1984). In many studies of designers thinking aloud, the designers appear to change their conception of

---

[1] In contrast to the formal artificial intelligence or philosophical use of the term frame (McCarthy & Hayes, 1969)

their own activity during the process of designing in a process of *reframing* (Cross, 2004; Dorst & Cross, 2001; Schon & Wiggins, 1992; Schön, 1983; Seelig, 2012; Suwa, Gero, & Purcell, 2000). Reframing can occur at two levels – not only are designers able to reconceive the goal of their design, but they can also shift their worldview within which design goals are conceived. By framing a design problem, a designer has the potential to then radically change the solution space by consciously reframing it (Dorst, 2015), where reframing permits a designer to develop designs that were previously inconceivable (Schön, 1984). Reframing can also be described as occurring through a *co-evolution* of the problem space and the solution space (Dorst & Cross, 2001; Maher, Poon, & Boulanger, 1996; Yu, Gu, Ostwald, & Gero, 2015). By understanding the problem in a different way (perhaps as a result of something seen in a potential solution) the solution space may change, which can repeat in an iterative fashion.

2.2 Existing paradigms for computational creativity

At different times, different paradigms have been dominant within the field of computational creativity. Two of the dominant paradigms can be identified within the literature as *generate and test* and *analogical reasoning*. Each of these fits the definition of a paradigm in that it is a signifier for a set of assumptions, concepts and values about the world that have led to the production of useful models of computational creativity.

The paradigm of *generate and test* gained prominence with the realisation that computers can rapidly search a defined space of possible designs, through a repeated cycle of generating and evaluating the product of generation. The paradigm was described by Langley (1987), although by this time there were already many models that could be described through generate and test in many domains, from art and architecture to physics and mathematics (Ashby, 1960; Cohen & Feigenbaum, 1982; Pearl, 1984)

Implicit assumptions of generate and test are that useful spaces to search can be identified by the developers of models and that such spaces will be sufficiently constrained that they can feasibly be searched.

Core questions addressed through models that use generate and test are about representation (how to represent designs), generation (how to generate designs) and assessment of fitness (how to test to establish preference in candidate designs). Within this paradigm many techniques were developed that have become canonical such as shape grammars (Stiny, 1980), L-systems (Lindenmayer, 1968) and genetic algorithms (Holland, 1975). The paradigm was used to generate and test from formally represented mathematical heuristics in the BACON system (Langley, 1987). Many useful designs have been produced by systems that were (at times implicitly) within the paradigm of generate and test, such as the genetic programming system that led to patentable designs (Koza, 1992; Koza, Al-Sakran, & Jones, 2005).

A subsequent paradigm for models of computational creativity is that of *analogical reasoning* (Gentner, 1983; Gentner & Forbus, 2011; Goel, 1997; Holyoak, 1996). In retrospect, one way to portray this paradigm shift is that humans were seen to be capable of changing the space of design, yet existing models of computational creativity were not currently doing this (Boden, 1991). Analogical reasoning provides a way of developing models of creativity that can change their conceptual space, most commonly by expanding the conceptual space through structure mapping of variables (Gentner, 1983).

The core questions addressed through models using analogical reasoning are of retrieval (how to find useful representations for the current situation), mapping (how to use retrieved representations to apply them to the current situation) and representation (how to represent knowledge in the system) (Gentner & Forbus, 2011). The implicit assumption of analogical reasoning is that a system will have a store of useful representations and be able to retrieve them as needed.

2.3 Situated transformation: Prior definitions and cognitive basis

Boden (1991) distinguishes two types of computational creativity as: *exploratory*, in which an existing conceptual space is searched to find previously unexplored areas; and *transformational*, in which the space itself is altered. In subsequent literature there has been divergence in the use of terms to describe this difference between searching within a frame, and moving to a different frame, Table 1. The use of the terms exploration and transformation is further challenged by the suggestion of Wiggins (2006b), and subsequently Ritchie (2012), that transformation can be described as meta-level search: if transformation is the movement between frames then why can this not be seen as a type of exploration within the space containing all frames?

This paper adopts the term *situated transformation* to describe transformation in a way that allows it to be rigorously distinguished from search, exploration and transformation as Boden defines it.

Table 1 Distinctions in the literature relating to search, exploration and transformation

Reference	*Term*	*Definition*
(Boden, 1991, 2009)	*Exploration*	"The space is defined (and constrained) by a set of generative rules" (Boden, 1991, p. 25)
	Transformation	"the space or style itself is transformed by altering (or dropping) one or more of its defining dimensions" (p. 25)
(Gero, 1994)	*Search*	"…a process for locating values of variables in a defined state space…" (p. 315)
	Exploration	"…a process for producing state spaces" (p. 315) by creating new state spaces or modifying existing state spaces
(Gero, 1990)	*Routine design*	"All the necessary knowledge [to carry out design] is available" (p. 10)
	Innovative	"The context which constrains the available

	design	ranges of the values for the variables is jettisoned so that unexpected values become possible" (p. 10)
	Creative design	"Occurs when a new variable is introduced into the design" (p. 11)
(Wiggins, 2006a)	*Exploration*	With reference to search as an activity to "find a complete solution among a set of partial and complete solutions, expressed in a fixed, symbolic representation" (p. 210)
	Transformation	"…transformational creativity can be cast as exploratory creativity at the meta-level" (p. 212)
(Dorst & Cross, 2001)	*Creative design*	"…the creative event in design is not so much a 'creative leap' from problem to solution as the building of a 'bridge' between the problem space and the solution space by the identification of a key concept" (p. 435)

The basis for this definition is of design as a situated activity (Gero, 1998; Gero & Kannengiesser, 2004). A *situation* (analogous to a frame), a construct that emerges from experience with the co-ordination of concepts, is internal to the system, and is affected by interaction between system and environment (Clancey, 1997, 1999). Significant empirical evidence for the central ideas of situatedness, and their application to computation, has been uncovered through investigation into *grounded cognition* (Barsalou, 2007; Pezzulo et al., 2013).

Consider that a creative design system is carrying out design activity within one situation (say, sketching a design for an art gallery) and an observation of a stimulus within the environment leads to a change of the situation (say, observing a passing bird) that alters the design. In developing computational systems capable of computational creativity there is value in developing distinctions that permit discussion of these changes of situation (Gero, 1998): how and why do they happen? There is a reason to be interested in the way that a design system changes its situation and to consider what causes these changes of situation. By contrast, consider that a behaviourist understanding of a design system would arise from an observation of its inputs and outputs and subsequent discussion of these. A situated understanding would arise instead from observation of inputs and outputs as well as an understanding of its changing situations and interaction with the environment.

The term *situation* thus requires further definition if the changes of a system's situation are to become an object of study. It lies outside the scope of this paper to review the basis for adopting a situated perspective on cognition (e.g. Tschacher & Scheier, 1999). However, key notions from the literature can be reviewed. Firstly, a system can be considered to hold concepts that are an abstraction from patterns in the world and that have a perceptual basis (Barsalou, 1999, 2005). Secondly, concepts are utilised within an assemblage of active concepts that make up a "world view" of the system within the current moment, that can change even in the absence of learning – what we are referring to as a situation (Barsalou, 2009; Clancey, 1999). Thirdly, the use of a concept (e.g. a concept of a design solution) is changed by the situation within which it is used (Clancey, 1997), as evinced by such phenomena as priming and implicit memory (Schacter, 1987). Another way of conceiving of situated systems is that implicit within their state is a 'world-view', an understanding of the world. A situated system considers even its own concepts from within this world view, which world view is subject to change (Barsalou, 2009).

The notion of transformation is a contested area within the research literature. This section has argued for a situated understanding of what is meant by a frame of design; and that the movement of frames matters for understanding creativity. The term situated transformation has been adopted and in Section 3.1 is more rigorously defined.

2.4 Universe, known universe and conceptual space

Some symbols are introduced to represent these salient aspects of design as a situated activity. Following Kelly and Gero (2015b) the space of concepts that a creative system is capable of conceiving (i.e. without limits upon time, resources or experiences) will be referred to as its universe U. This construct of the universe is drawn from the nomenclature of Wiggins (2006a) and is useful for conceiving of the space of a system in the broadest sense; the limitations of what it is capable of representing given a defined internal language.

Most systems will only ever represent a small subset of the universe U. Through experiential knowledge (including reasoning about existing knowledge) a system can be said to have a space of concepts that it 'knows about', denoted as $U_!$. There are many ways in which a system can sense and develop some kind of internal representation of its world; $U_!$ refers to this knowledge as it develops, the *umwelt* of the system (Von Uexküll, 1957/1992; Ziemke & Sharkey, 2001). Through experience the system learns and $U_!$ comes from these experiences, the structure of a system and the way in which it represents its world. A system has, at any time t, a state. As time progresses and a system that is capable of learning learns, it can be seen as having a static U (i.e. what it is possible for the system to know as limited by its structure does not change) and a changing $U_!$.(i.e. what it knows about through experience in its world).

A further distinction, drawn from situated cognition, is to distinguish between what the system knows, what has been referred to as $U_!$, and 'what the system knows now' (Barsalou, 2009; Clancey, 1997, 1999; Kelly & Gero, 2015b). This idea of 'what the system knows now' is that within any particular state the system only has access to a subset of $U_!$. For example, a system may know about many concepts but, within a state, only be utilising some of them. This further reduced space of what the system both knows about and is utilising is the *situation* of the system, denoted by S, and this can be observed to change, independent of changes to $U_!$. Why would a creative system not utilise all of its knowledge all the time? One reason is that it may have operations that are more effective within smaller spaces, such as combinatorial operations, where an appropriate space must be found prior to the operation occurring.

The phenomenon of designers making *unexpected discoveries* during design activity can be used to give an example of these terms (Suwa et al., 2000; Suwa & Tversky, 1997). Consider that a designer, with a particular conception of the design task, is looking at a sketch that they have created. The designer has significant knowledge about the world from experience, but it matters that the designer is not attending to all of this knowledge during the moment in which the sketch is being observed – in perhaps the same way that one can know their mother's birthday but not be thinking about it whilst reading a research article. Those things that are being attended to by the designer – aesthetic judgements, axes of symmetry, similar design problems from past experiences, as well as perhaps a multitude of bodily concerns and unrelated trivia – all make up the current situation S within which the sketch is being interpreted. Designers in such a circumstance are observed to be able to make unexpected discoveries when they interpret something within their own generated design that was not intentionally put there. This can be described as interpreting from within a space S and making use of $U_!$ to change S – which change informs future design actions (Kelly & Gero, 2014).

The distinction of situated transformation in creative systems is based upon this foundational concept from situated cognition – that the situation S can change independent of changes to $U_!$. During creative activity a system can be thought of as navigating its own knowledge, through both search (within S) and situated transformation (within $U_!$ moving from one situation to another).

3 The paradigm of generate and situated transformation

The proposed paradigm of *generate and situated transformation* describes systems that generate designs within a state space and move between different state spaces in certain ways. This movement between spaces is a well-established notion, but the point of departure in this exposition is in describing knowledge as being situated, and from this suggesting that *the way in which this change of state space occurs matters*. Situated transformation, requires that the movement to a new space be based upon the past experiences of the system, and that the system be situated. Fundamental questions in models with generate and situated transformation are about how to represent knowledge in a situated way and how a system changes the space of design.

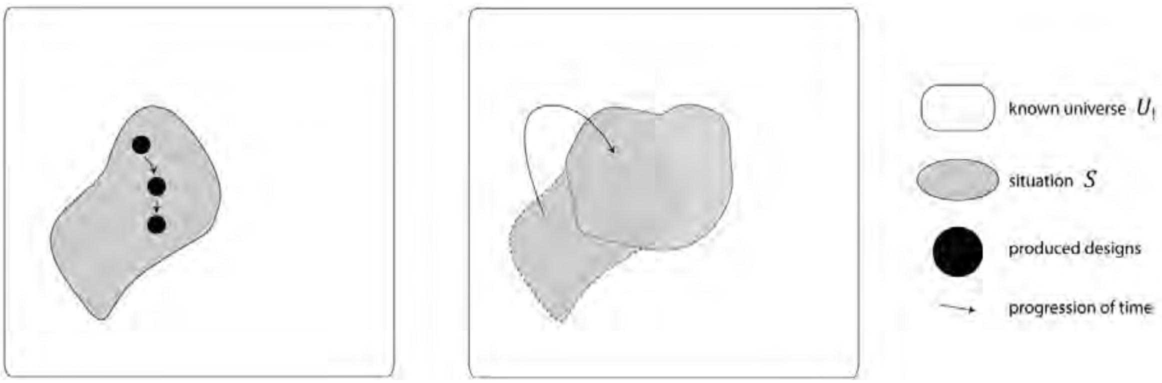
3.1 Distinguishing situated transformation

Situated transformation is defined here by building upon the definition given by Gero (1994) as (addition italicised): A process which creates new design state spaces or modifies existing design state spaces *based upon past experiences*. This introduction of the reference to past experiences tacitly implies by definition that situated transformation (unlike transformation as it is commonly conceived) is only possible within systems that have some form of *grounding* (Barsalou, 2010). Symbolically, the situation S implies a space of potential designs. Prior definitions referred to transformation as activity that changed S thus opening up a different space of potential designs. Under the definition proposed here, only changes to S that are caused by something within $U_!$ are considered to be situated transformation. Thus a mechanistic shift of conceptual space – such as a bit-flipping mutation in a genetic algorithm – is not considered to be situated transformation. An example of situated transformation is through interpretation – the process of interpreting something in the world causes the system to shift to a different situation (Kelly & Gero, 2014). Such an argument about the need for changes to S to be contingent upon $U_!$ can be considered in the context of Searle's Chinese Room argument: that something is different when a symbol manipulation system makes choices on the basis of its experience rather than instructions (Searle, 1980).

The motivations for adopting this definition are: (i) that situated transformation is rigorously distinct from search (and from exploration as defined by Boden); and (ii) that by taking this on as a core concept of the paradigm, it promotes the development of models that explore the relationship between situated/grounded cognition[2] and computational creativity.

Using the symbols defined in Section 2.3, situated transformation is about changes to S that are brought about by $U_!$. Search is defined as activity that is oriented towards the goals of the system within an unchanging situation S. Systems that solve optimization problems are commonly described as 'searching' for a solution. In search the conceptual space is static and bounded and search encompasses the activities (e.g. generating and testing) that are undertaken to move around within this space.

Situated transformation in contrast is defined as activity that changes the system to a new S where that movement is based upon $U_!$. The connection with past experience is relevant, as the difference between situated transformation and Boden's description of transformation lies in the distinction between S and $U_!$. Search and situated transformation are represented in Figure 1 (after Kelly & Gero, 2015b).



(a) (b) (Key)

Figure 1 (a) search as movement within a situation; and (b) situated transformation as leading to a changed situation

A distinction can be made between two types of operations within a system: (i) those that take place within S without access to $U_!$-outside-of-S; and (ii) those that take place within

2 Refer to Barsalou (2010) for a discussion of the overlap between these terms

S *with* access to $U_!$-outside-of-S. If S is defined as a set of concepts $\{c|c \in U_!\}$ then this space $U_!$-outside-of-S can be defined as its complement $T = \{c|c \in U_! \wedge c \notin S\} = U_! - S$.

It is the distinction between these two types of operations that makes it possible to define situated transformation such that it is clear how it differs from search. Situated transformation occurs when a system changes S with access to T, i.e. past experiences outside of S, described here as type (ii) operations. This is distinct from type (i) operations which reference only those experiences within S. Consider two descriptions of systems that using this definition can be seen not to be exhibiting situated transformation despite perhaps appearing similar to an external observer:

(1) A system where $S = U_!$ that has a multi-tiered approach to search involving tasks, sub-tasks, sub-sub-tasks, and so on.

(2) A system that searches within S and at a certain time interval changes the space of S according to a predefined algorithm.

In (1) the system does not change the situation and is thus an example of search rather than situated transformation, since for systems with generate and situated transformation it is necessary that $S \subset U_!$. This is distinct from systems that generate and test where it is possible (but not necessary) that $S = U_!$. Example (2) demonstrates the way in which changes to S occur. In (2) the system has a changing situation. However, the way in which the change of situation is occurring is not dependent on experience thus is not situated transformation.

This thesis as to why this connection to experience matters can be further elaborated. It can be argued by considering the obverse position, as seen in example (2) above. A system that has either an algorithmic or stochastic means of changing S falls into the same trap of being able to be considered as meta-level search. It is through the type (ii) operations, by definition, that a system has the potential to bring elements of $U_!$ into S. For example, during interpretation the current S, its relationship to $U_!$ and its relationship to the object of interpretation (either internal or external) form a basis for either bringing elements from $U_{!}$ into S or removing elements of S.

A formal representation of situated transformation is as a function Γ that produces a new situation from the current situation S_i, past experiences T_i, and some stimulus that is sensed, x, which may have its origins either internal or external to the system, Equation 1.

$$\Gamma(S_i, \mathrm{T}_i, x) \rightarrow S_{i+1} \qquad (1)$$

Operations that make up search are not functions of this broader space $U_{!}$. For example, generation G can be defined as a function of situation S that produces a design element α, Equation 2.

$$\mathrm{G}(S_i) \rightarrow \alpha \qquad (2)$$

In the paradigm of generate and situated transformation, systems generate within S and have some kind or combination of type (ii) operations that enable them to change S. The paradigm is most applicable for systems that have a wide range of experiences $U_{!}$ and hence have significant potential for type (ii) operations to be effective in producing a useful S.

3.2 What change to the situation looks like

The definition of situated transformation as distinct from search and from the classic definition of transformation (as Boden describes it) is central to the paradigm of generate and situated transformation. It is based upon the division between type (i) operations that occur within a situation; and type (ii) operations that draw upon experiences to change the situation. In order to give a picture of the implications of this distinction a more formal definition of a situation will be adopted in which a situation can be represented by a number of variables (Gärdenfors, 2000). For example, the representation used in some systems allows for the space of potential designs to be described by a number of design variables (Gero, 1990). This discussion at the level of variables allows for a description of the ways in which a conceptual space changes (Kelly & Gero, 2015a).

In this understanding, the potential variables of the system can be designated as the set V_U. The experiences of the system limit the variables that the system has knowledge of a subset of these, the set $V_{U_!}$. The current situation limits the variables to a subset of these, the set V_S. These relationships are expressed by Equation 3.

$$V_S \subseteq V_{U_!} \subseteq V_U \qquad (3)$$

Within the current situation S, a region where design is occurring can be identified by variables, the context of those variables (the understanding that the system has of what the variables signify) and the limits upon these variables of the type $v_i \in V_S\colon m < v_i < n$.

The ways in which changes to the situation may occur in this view can be listed as:

(i) changes to the limits upon variables (expansion or reduction of space within the same variables);

(ii) changes to the context in which the system conceives of variables; and

(iii) changes to the membership of V_S.

An example can demonstrate these three changes. Consider a designer designing a chair. One variable amongst others within V_S is for the *number of legs* represented by n. Within their current conception of the design task (perhaps based upon past experiences) the designer is working within the limits such that $n \in \mathbb{N}$ and $1 \leq n \leq 4$, where $\mathbb{N}$ is the set of natural numbers. Within this example a narrative can be contrived to describe the three types of changes.

(1) **Changes to limits of variables.** The variable for *number of legs* can be extended such that $1 \leq n \leq 6$ or perhaps the boundary case $0 \leq n \leq \infty$. In this new conceptual space the chair may have 0 legs.

(2) **Changes to the context of variables.** The variable may be reconceived by the system such that $n \in \mathbb{N}$ becomes $n \in \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers. In this new space the chair may have 4.5 legs.

(3) **Changes to the membership of the set of variables.** Perhaps in response to (1) the system introduces a new variable for *number of strings connecting chair to ceiling* and ceases to attend to the variable for *number of legs*.

This list represents a catalogue of ways in which changes to the situation may occur, however it does not give examples of how they occur (computationally) nor begin to give an account for how they occur in designers (phenomenologically). Some anecdotal accounts of each of these types of change to the design space serve to flesh out the concept.

An example of changes to the limits of variables can be found in the case of pizza-maker, Johnny di Francesco, who won the pizza-making world championships by taking the traditional ‘four cheeses’ pizza and inventing a ‘99-cheese’ pizza (Leggatt, 2014). Whilst di Francesco was almost certainly not consciously extending a design variable, this is one way that this invention might be described.

An example of changes to the context of variables can be found in the Kekulé’s discovery of the structure of the Benzene molecule (after Boden, 2009). At the time of Kekulé’s work it was understood that chemical bonds formed strings, but the idea that they could form rings was not considered within this dominant conception of open strings. Within this space, a solution to the problem of describing the structure of Benzene was not possible; it had been an open problem for some time. Kekulé ‘discovered’ that Benzene formed a ring structure, and famously reported that this discovery came in the form of a dream of Ouroboros, a snake eating its own tail .As quoted by Boden (1991) Kekulé describes this dream: “But look! What was that? One of the snakes had seized hold of its own tail, and the form whirled mockingly before my eyes” (p. 63). One way that this discovery can be considered is as a move from a world-view in which chemical bonds form an open curve towards one in which they can form rings (Boden, 1991). The context of the variable for chemical bonds for Kekulé has been fundamentally changed from needing to bond with another atom to being able to bond with existing atoms.

An example of changes to the member of the set of variables can be found in the work of Qian and Gero (1996) who describe the way that the distant analogy system they developed can use analogical goal achievement mapping during design. In this example the system is considering the design of a door, in which the set of variables representing hinges is replaced by the set representing sliders, thus opening the possibility of sliding doors.

3.3 How change to the situation occurs

In systems with generate and situated transformation there is a difference between 'what the system knows', the experiences represented by $U_!$, and 'what the system knows now', in the smaller space represented by S. In some systems (e.g. the case of human cognition) $U_!$ is extremely large, making combinatorial explosion an issue for search by generation within $U_!$. The suggestion for models that use generate and situated transformation is that generation, and the combination of concepts, can occur within the more limited space of S, so long as the system is able to move towards a useful S. The question remains: how does this movement occur?

A proposed response is that movement of S occurs during interpretation. Interpretation can be defined as "a continuous, dynamic, constructive activity that attempts to construct an internal representation from a source, using expectations where it is possible and constructing an explanation from existing knowledge where it is not" (Kelly & Gero, 2014, p. 113). A system can interpret a source that is internal as well as a source that is external, where a source is defined as the object of interpretation (i.e., an internal representation or an external object). The 'expectations' in this definition refer to the knowledge from experience that the system has both in terms of the explicit expectations of S as well as the implicit expectations of $U_!$-in-the-context-of-S (Kelly & Gero, 2015b). Interpretation is a process, a type (ii) operation that produces an internal representation I from a source x and a set of expectations E, resulting in an internal representation being constructed.

$$x \overset{E}{\Longrightarrow} I \qquad (4)$$

For parsimony, all units of knowledge represented in the system will be referred to as concepts, whilst recognising that in many systems knowledge is represented as a hierarchy of abstractions (Barsalou, 2005). The situation S can be considered to be composed of a number of concepts. The significance of $U_!$-outside-of-S is that even when the system is performing this type (ii) operation and using $U_!$, the concepts that are not a part of S (that are outside of it) implicitly have a location in relation to S, shown in Figure 2. This relationship, between concepts outside of S and S itself, may be defined in a number of ways. For example, it could be defined by the Euclidean distance of a concept from S or it could be defined in terms of the number of connections away from S.

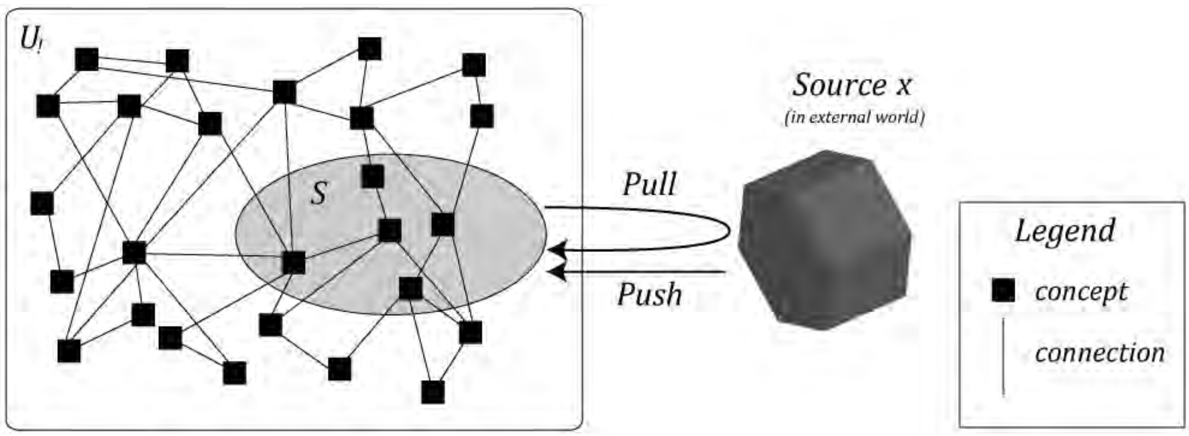


Figure 2 Concepts of the system in relation to the current situation S during interpretation of a source through pull and push (source: Kelly & Gero, 2015a)

The concepts outside of S thus have an implicit relationship with S and this relationship is made use of during interpretation. Interpretation involves both a ‘pull’ and a ‘push’ from expectations and data respectively (Gero & Kannengiesser, 2004). The expectations held by the system (explicitly within S and implicitly within $U_!$) make up the pull by which the system tries to construct an interpretation from x that looks like expectations. Where insufficient similarity to expectations is found a push from the data of x guides a change of expectations.

Kelly and Gero (2015b) describe how interpretation occurs within a system; however it is sufficient for describing the current paradigm that the process produces a movement of S. A source x becomes the trigger for the system to bring something from $U_!$-outside-of-S into S, changing the situation.

This language can be used to describe h-creative (historically creative; Boden, 1991) acts of creativity. For example, WiFi, or at least a major part of the technology for wireless networking, was invented by the Australian researcher John O’Sullivan (Sygall, 2007). O’Sullivan was involved in radio astronomy research to detect atom-sized black holes and developed novel ways to use the Fourier transform to avoid atmospheric distortion. In later years, when faced with the problem of reverberation in wireless networking, he was able to use this knowledge to come up with a viable solution that is still widely used in the world today. In the language used here, the work with black holes became a part of O’Sullivan’s experiences, a part of $U_!$. There was nothing in the problem statement when working on the challenge of wireless networking that specified any relationship to fast Fourier transforms or to black holes. There were many people working on the problem worldwide and for these other researchers the space of S did not have appropriately fit responses. O’Sullivan was able to interpret the problem and bring elements of $U_!$ into S to find the solution that was eventually patented. From this example it can be seen that there was a need to firstly have a rich enough $U_!$ from which useful experiences could be drawn; and secondly that the process of Γ was then able to bring these experiences into S.

The three types of change to S described in Section 3 can take place through changes to the concepts making up S. Concepts may be introduced or removed from the situation such that variables are extended, changed or introduced.

New concepts introduced to the situation and the removal of previous concepts may introduce changes to the space through extension of variables, changes to variables and new variables at the expense of others.

4 Demonstrations of situated transformation

A number of simple generative systems can be described to show, by demonstration, the notion of situated transformation through interpretation. The following examples all instantiate situated transformation by taking actions within a situation, and changing the situation during interpretation. The process of situated transformation through interpretation is described in more detail by Kelly and Gero (2014). A brief symbolic description here aids the examples that follow.

Following Section 3, processes within the system have been defined as being type (i), those that operate upon S, and type (ii), those that also have access to T, other knowledge held by the system outside of the situation. Situated transformation in all of the models here occurs through the process in Table 2. This is a specific type of situated transformation in which the only part of the environment that the system is interested in is the product of its own generation – a more general description of situated transformation can be given. The process that is referred to here as *mapping* during interpretation referred to here is described elsewhere as *constructive interpretation* (Kelly, 2011; Kelly & Gero, 2014, 2015b). The term α refers to representations of a design produced by the system.

Table 2 Pseudocode for situated transformation through interpretation

Pseudocode	*Description*
Input $U_!, S_1$	#A system with experience and a situation that uses a subset of these experiences
While termination condition not met	
$\mathrm{G}(S_1) \rightarrow \alpha$	#Generation within the situation produces designs
If $map(\alpha, T_1)$	
$\Gamma(S_1, \mathrm{T}_1, \alpha) \rightarrow S_2$	#Interpretation occurs and changes the situation
Else	
$\Gamma(S_1, \mathrm{T}_1, \alpha) \rightarrow S_1$	#Interpretation occurs but situation does not change

Distinguishing situated transformation from transformation allows for discussion of the different ways that a system might be able to explore and to measure their benefits. If a system is considered that has had a great deal of experience (large $U_!$) and that has finite resources for generating effectively based upon experiences (restrictions upon the size of S) then a comparison of methods of situated transformation can determine the useful ways of moving a limited S into different parts of $U_!$. For example, Kelly (2011) proposes a measure for the ability of a system to interpret a single stimulus in novel ways. The proposed measure is a function of both the similarity of the internally produced representations to their referent in the external world (the environment of the system) and the number of different interpretations that the system is able to produce. Such a measure is inspired by human divergence testing (McCrae, 1987; Runco, 1991).

4.1 A system generates

Consider a generative system that produces a series of numbers $\{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\}$. The situation of the system is defined by two variables: (i) r is a range within the set of integers and expressed as a minimum and maximum constraint; and (ii) ρ which designates an operator, which is always one of multiplication, division, addition or subtraction. The system has an initial state.

To demonstrate by example, consider that the system is initialised such that r is constrained by $5 < r < 95$ and that $\rho = +$, the operator for addition. The system *generates* by choosing two values for the variable r within constraints and performing the operation ρ on them to produce an artefact (the resulting number). For example, the system chooses first 6 and then 72 and adds them to produce an artefact of 78. The system as it has been described has a bounded space that it is 'searching' through the process of generation. Given enough time it would eventually 'discover' all possible artefacts in the space. In the example of system just initialised, this is the clearly bounded set $\{10,11,12,\ldots,189,190\}$.

The system has a clearly defined generator and performs search by generation that uses this generator within specified rules. It is routinely going about the search of a space.

4.2 A system generates and situationally transforms

Through adaptations the system in Section 4.1 can be given the ability to situationally transform. There are multiple ways this could be accomplished. Consider that after producing an artefact, the system interprets what it has produced such that:

1) If the current artefact α_n along with the previous two artefacts α_{n-1} and α_{n-2} make up an arithmetic or geometric progression, then change the operator to the corresponding operator; and
2) If the numeric value of the common difference ($x = \alpha_n - \alpha_{n-1} = \alpha_{n-1} - \alpha_{n-2}$) or the common ratio ($x = \alpha_n/\alpha_{n-1} = \alpha_{n-1}/\alpha_{n-2}$) is outside of the constraints of r then extend the bounds of the constraint by lowering the minimum or raising the maximum.

Within this system the situation S is defined by the variables r and ρ (as it was in the generative system in Section 4.1). Through situated transformation however the system is able to move the space S around within U. The system commences within one conceptual space but upon the basis of artefacts generated from within S it shifts to a different space. At time t the system produces α by generating using the concepts within S_t. Through Γ (in this model the process of interpretation) the system changes the situation to produce S_{t+1}.

Table 3 shows sample results from this system following implementation using a pseudo-random generator for selecting values[3]. In these results the system first moves around by generating within the situation as defined by the initial conditions. At time $t = 40$ the situation changes. The artefacts that the system has produced lead to a change of the conceptual space.

Table 3 Extract from results of a generate-and-situated-transformation system

T	*Description*	*Detail*
0	initial constraints	5>r>95
0	initial operator	addition
1	generated	42
…	…	…
36	generated	105
37	generated	118
38	generated	96
39	generated	79
40	generated	62
40	interpreted	AP of -17
40	changed operator	subtraction
40	changed constraints	minimum to-17
41	generated	1
42	generated	9
43	generated	-43
…	…	…
7868	generated	5
7869	generated	15
7870	generated	45
7870	interpreted	GP of *3
7870	changed operator	multiply
7871	generated	-54
7872	generated	-441
7873	generated	86
7874	generated	2368

4.3 Elaborating on the system

The system described in Section 4.2 is a simple example of generate and situated transformation. Part of its simplicity comes from the fact that is has an unchanging $U_!$ and that it has a trivial language for its knowledge. More complex systems that show generate and situated transformation have the characteristics that they: (i) are able to represent the world in

[3] Python code for the system is available at http://nickkellyresearch.com/script-generate-explore-demonstration/

a non-trivial way; (ii) are embedded in a non-trivial world; and (iii) have representations that change through experience (i.e. $U_{!}$ changes). These elements in more complex systems with generate and situated transformation can be elaborated with reference to the system described in Section 4.2.

The representation of the world in this exemplary system is limited to a single dimension, made up of the natural numbers. The representation of the system could become more complex through the inclusion of additional dimensions. For example, the addition of one more variable $m \in \mathbb{N}: 1 \leq m < \infty$ gives the system more representational power – for example, it can now represent an infinite set of regular shapes with side length α and number of sides m. In a similar way more complex systems result from the addition of yet more variables. The example could be revisited such that instead of producing numbers and interpreting for arithmetic and geometric progressions, the system is able to produce shapes and look for relationships between produced artefacts (e.g. size or emergent shapes) (Gero & Kazakov, 1998). In this way it could produce situated transformation by moving S around on the basis of experience.

Also, the system in Section 4.2 is not embedded within any kind of external world. It can only learn from its own actions, and its actions can be seen as the gradual exploration of its own capacity to create within its $U_{!}$. The addition of the more variables such as m merely changes the nature of the representations that the system produces. Consider however that the system has some input from a dynamic world. Perhaps it has a sensor in the physical world; perhaps it experiences a random image from the Google image database (which itself is constantly growing); or perhaps the representations that the system produces are perturbed by some external entity (e.g. edited by another creative system). Whilst the system still has the same restrictions of only being able to represent the world using the language it has for representation (in this case α and m) it now has the potential for its situated transformation to have its direction influenced by the external world.

Perhaps most importantly however is the limitation that the system described in Section 4.2 and extended in this section is unable to change its representation. The system might be looking at the Sistine chapel; but it can still only represent what it sees as a regular shape with number of sides m and length α. In contrast to this, we observe in human creativity that new variables can be brought into a design problem and old variables discarded. Consider that the system could be extended to have the possibility of bringing variables from $U_!$-outside-of-S into S through the process of interpretation; and further, that it has the capacity to create novel variables based upon experience. Many implemented systems in the domain of analogy making (see Section 5.2) provide a demonstration of this capacity.

Finally, these descriptions only discuss systems that hold one representation at a time. More complex systems include relationships between multiple representations.

4.4 Types of situated transformation

The method of situated transformation through interpretation in Section 4.2 is an example of how situated transformation could be implemented. The focus is not so much the particular instantiation as the phenomena that the products of generation α are the trigger for changes to S. The system has within it means by which actions taken utilising the current concepts within S can trigger introduction of implicit concepts.

The system described in Section 4.2 is simply generating, as distinct from designing. As such it has no 'goal' and there is no basis upon which to compare one method of situated transformation with another. However, were the system to have a goal then it could be concluded that the implementation as described would be a more or less useful type of situated transformation than randomly shifting conceptual space. Making the notion of situated transformation explicit makes it possible to compare the ways in which a system is changing the space; different ways of carrying out situated transformation can be more or less useful for producing a space in which design solutions can be found.

5. Computational models of generate and situated transformation: A brief review

The paradigm of generate and situated transformation can be used to describe recent models of computational creativity. The core concepts of generate and situated transformation have been clarified as: (i) situated transformation occurs in models that are situated to some extent; and (ii) situated transformation happens by changing the situation in a way that is informed by past experience. The key assumption of this paradigm is that by conducting research into different methods by which situated transformation can occur, new contributions to the field of computational creativity could be made. In the models described here, the method by which situated transformation occurs is a based upon interpretation. The models make use of past experiences when interpreting either the outside world or their own work. The changes occurring during interpretation can cause changes to the situation S by drawing upon $U_{!}$-outside-of-S (Kelly & Gero, 2014).

5.1 Revisiting analogy-making systems

There is a long history of computational analogy-making systems (Gentner & Forbus, 2011; Goel, 1997; Hofstadter, 2008; Holyoak, 1996). Many of the ideas in these systems have their genesis in Gentner's structure-mapping, which describes the way that elements in a source come to be mapped onto a target (Gentner, 1983). For example, Kekule's discovery of the structure of Benzene can be described as a mapping from the source (a snake grabbing its own tail) to the target. This view of analogy within computational models of creativity has been persistent, in which individual concepts are analogically mapped onto other concepts. For example, Vattam, Helms, and Goel (2010) give an account of how designers use biological analogies to introduce new variables into the conceptual space (using an analogy with the small intestine in designing a water desalinator).

Analogy making can be described within the paradigm of generate and situated transformation. A system is within a situation represented by S and is considering an artefact. By drawing upon $U_{!}$-outside-of-S (what under the paradigm of analogical reasoning was described as retrieval and mapping), concepts can be introduced to S or altered, and thus causing the space S to change, and satisfying the description of situated transformation.

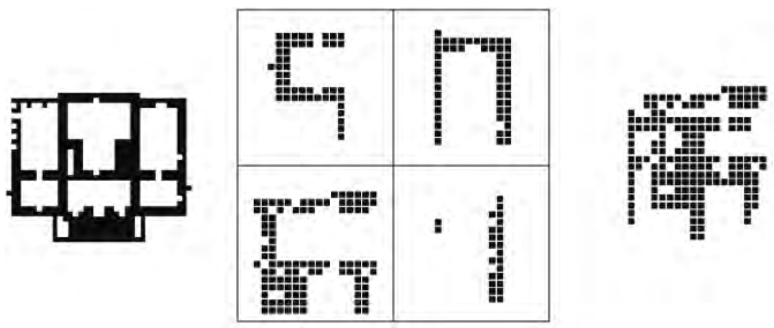
What is gained by this description is that it gives both a context and a reason for the effects of analogical reasoning. For the context, analogical reasoning is constrained/informed by the current situation S. For this reason, the effects of analogy are important not solely in being able to map a source onto a target but also for changing the situation within which future design actions will take place. Analogy, in this view of design, is one way how a system moves from a situation within which the solution cannot be found towards a situation within which a solution can be found.

Systems that make analogies demonstrate one way in which the situation can change. In the interpretation of a source, S_i may move to S_{i+1} through the introduction of outside concepts that come from the experienced universe of the system $U_{!}$. Typically this occurs through a partial mapping of a representation of the source onto a representation of the target. An outcome from this exercise of mapping is that concepts from the target can change the situation. The example from Vattam et al. (2010) could be described as a situation S that contains no concept of a "small intestine". During interpretation an implicit concept, from the experiences of the system but outside of the current situation, is attended to, changing the situation. This is an example of a type (ii) operation, bringing something from $U_{!}$-outside-of-S into S.

5.2 Experimenting with situated transformation

Within the paradigm of generate and situated transformation the focus is upon the way that a system makes changes to the situation S. Thus even a simple system that has an interesting process of utilising $U_!$ for situated transformation becomes of interest. Two different examples serve to demonstrate this.

A simple system with generate and situated transformation was implemented in the domain of floor plans (Kelly & Gero, 2011, 2014, 2015b). The system was trained on floor plans by three architects (Louis Khan, Andrea Palladio and Frank Lloyd Wright), such that it held knowledge in 16x16 pixel feature maps of floor plans, Figures 3(a) and (b).



(a) (b) (c)

Figure 3 (a) A representation of the original floor plan; (b) a set of four 16x16 feature maps as current S; and (c) the representation produced through $G(S)$ (source: Kelly, 2011)

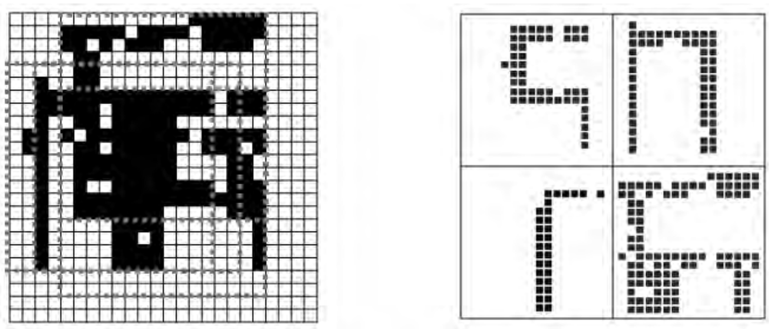
The system begins in a situation S that is made up of four of these feature maps that have a relationship, e.g. they are part of the one original floor plan. Figure 3(b) shows an example of the four concepts (feature maps) that constitute initial S. From this state, the system generates potential designs by utilising these concepts (16x16 pixel representations) that are within current S. It does this by randomly placing them within a 24x24 pixel 'canvas', Figure 3(c).

The interesting feature of the system is that it then interprets what it has produced, an implementation of Γ. This occurs through a saccade across the canvas, interpreting 16x16 sized perceptual 'chunks' and attending to the internal representations produced. This process of interpretation is an example of a type (ii) process in that it has access to $U_{!}$-outside-of-S. In most cases the internal representations produced correspond to the 4 concepts that were a part of S prior to drawing. In some cases, however, interpretation produces internal representations that come from $U_{!}$-outside-of-S.

Interpretation was implemented in the system as a situated process (Kelly, 2011). Each S has a different $U_{!}$-outside-of-S. Interpretation was implemented such that any access to $U_{!}$-outside-of-S is influenced by the current S through a measure of distance within conceptual space. Following Gärdenfors (2000), the idea of distance in conceptual space is that a concept held by a system can be represented by a region within a conceptual space, allowing for measures of distance between concepts for a quantitative measure of similarity. A concept that is in $U_{!}$-outside-of-S might be 'close' to the current S but far away in another S. This distance influences the likelihood of being brought into S during interpretation, as shown in Figure 2.

When the system brings new concepts into S through this process of interpreting, it is an example of situated transformation. The criteria that the system has both changed S and it has come about through reference to past experiences are satisfied. Figure 4(a) shows the output from a saccade across the design generated in Figure 3(c), where the four grey dashed squares indicate features that have triggered concepts (16x16 feature maps) within the system. Figure 4(b) shows the internal concepts that are the basis for these features being found interesting by the system. Three of these concepts (top left, top right and lower right) are seen to be present in Figure 3(b), indicating that they were expected because they were used in drawing the canvas. However one of these concepts, the lower left, shows a concept that has come from $U_{!}$-outside-of-S. This has occurred firstly because the feature observed has some similarity to the concept; and secondly because the concept is 'close' to the current situation (as measured within the conceptual space of the system). The example is described in further detail by Kelly and Gero (2014).

Using the formulation from Section 3 each situation S_i that the system occupies implies a limited range of potential designs with variables V_{S_i} given the method of generation G. When, through interpretation, the system changes to S_{i+1} the limits to these variables can change.



(a) (b)

Figure 4 Interpretation within the system occurs through a saccade from top left to bottom right: (a) the four areas used for construction during interpretation marked by dashed grey lines; (b) the concepts constructed through interpretation, where the lower left comes from expectations (source: Kelly, 2011).

6 Discussion

This paper has described the paradigm of generate and situated transformation as appropriate for developing models of computational creativity. Situated transformation has been identified as the movement of a system between different situations on the basis of experience. Interpretation has been identified as one way in which situated transformation can be computationally implemented. It has been claimed that this paradigm of generate and situated transformation is useful because it opens up avenues of inquiry and novel questions, and leads to methods with which to carry out enquiry. This discussion section addresses these in turn.

6.1 Avenues of inquiry and questions

How can we move towards systems that can find a useful situation within which to carry out design activity? Humans utilise their capacity for situated transformation during design activity with impressive results. For example, humans are capable of having *moments of insight* when "a solver breaks free of unwarranted assumptions, or forms novel, task-related connections between existing concepts or skills" (Bowden, Jung-Beeman, Fleck, & Kounios, 2005, p. 322). There is still much that is not understood about how to create systems that are able to move towards a useful frame within which to carry out search. The paradigm of generate and situated transformation is appropriate for developing systems with this capacity. The paradigm is most apt for systems that have a great deal of knowledge from experience.

How can situated transformation be further developed within creative systems? Some systems within the literature can already be described as carrying out generate and situated transformation. By articulating the paradigm, the aim is that future systems can explicitly describe the way in which the systems are carrying out situated transformation, allowing for transfer and progression to more developed forms of situated transformation.

Further, there is much about human creativity (and cognition in general) that is not yet understood. Whilst the phenomenon of designers carrying out situated transformation during design has been well documented (with other names), it has not yet been modelled to the point where it can be said to be adequately understood. There is potential for models with generate and situated transformation to contribute to our understanding of human creativity.

6.2 Methods with which to carry out enquiry

The paradigm of generate and situated transformation also posits new methods with which to carry out enquiry. Making the phenomenon of situated transformation explicit and measurable allows for the possibility of comparison of systems on the basis of situated transformation. Consider an experiment of the type where two systems have identical large knowledge bases and implementations of generation but different implementations of situated transformation. Replications of such experiments can allow for incremental development of useful techniques for situated transformation within specific domains.

In an example of such measures, Kelly (2011) suggests two complementary measures for systems that interpret, the often used measure of *similarity* (a measure of similarity between the source and the interpreted representation) combined with *reinterpretation frequency* (a measure of how many different interpretations the system can produce of the same source over a certain time period). This second measure in combination with the first, whilst limited, introduces a metric for the situated construction of interpretations within a system.

There is also potential for further cognitive studies of the way that human explore and using these results to inform construction of models of creativity. The cited study of unexpected discovery in humans provides a model of this type of study (Suwa et al., 2000).

6.3 Conclusion

This paper has suggested that models of computational creativity arise within a particular paradigm, whether or not their creators are explicitly aware of that paradigm. As described by Kuhn, scientific paradigms both constrain and focus research. Two historical paradigms, those of generate and test and analogical reasoning, have been discussed. Each of these paradigms has core concepts and assumptions and can be used for re-interpreting historical models of creativity or to inform current and future models of creativity.

This paper has described generate and situated transformation as a paradigm that presents novel research questions and methods with which to carry out enquiry. Any paradigm has core assumptions and concepts that define it. The key assumption of generate and situated transformation is that the way in which a model changes the space that it is searching matters. Specifically, the focus is placed upon *how* a model moves from one situation to another through reference to its own experiences.

The paper has contributed a definition of situated transformation that allows it to be rigorously distinguished from search, where situated transformation is a process that brings about new state spaces or changes existing state spaces on the basis of past experiences. There are many ways that situated transformation can be implemented in creative systems. This paper has described situated transformation that occurs during the process of interpretation. It has identified the need for two types of process in models of creativity: those that occur within the current situation, and those that make reference to experiences outside the current situation that may lead to changes of the situation. Interpretation is an example of this second type of process, where some focus (a source or target, be it internal or external) forms the basis for drawing upon experiences. This allows for a balance between the convergent (search, in a restricted space) and the divergent (situated transformation).

Models with generate and situated transformation change their conceptual space as a result of processes that make use of existing knowledge – as distinct from mechanistic processes that can transform a state space without reference to the system's knowledge. This paper has provided examples of how this looks through models that use interpretation as the process through which the situation changes. Interpretation of a source, such as the developing design itself, becomes the trigger for a system to bring new concepts into a conceptual space or else change the space in another way. A human phenomenon identified in the introduction is the way in which designers are able to reframe the problem that they are working on, changing their conception of what they are doing. In some cases it is this reframing that makes it possible to find creative solutions to design problems. Models with generate and situated transformation provide a way of enquiring into this phenomenon, in the expectation of eventually understanding it.

Acknowledgement

This work is supported in part by the US National Science Foundation under Grant Nos CMMI-1161715 and CMMI-1400466. The work is also supported through the Australian Government's Collaborative Research Networks (CRN) program. 'Digital Futures' is the CRN research theme for the University of Southern Queensland.

References

American Heritage Dictionary of the English Language. (Ed.) (2000) American Heritage Dictionary of the English Language (4th ed.). Boston, MA: Houghton Mifflin Company.

Ashby, W. R. (1960). *Design for a Brain*: Springer Science & Business Media.

Barsalou, L. W. (1999). Perceptions of perceptual symbols. *Behavioral and brain sciences, 22*(04), 637-660.

Barsalou, L. W. (2005). Abstraction as dynamic interpretation in perceptual symbol systems. *Building object categories*, 389-431.

Barsalou, L. W. (2007). Grounded Cognition. *Annual Review of Psychology, 59*(1), 617-645. doi:10.1146/annurev.psych.59.103006.093639

Barsalou, L. W. (2009). Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B: Biological Sciences, 364*(1521), 1281-1289.

Barsalou, L. W. (2010). Grounded cognition: past, present, and future. *Topics in Cognitive Science, 2*(4), 716-724.

Bentley, P., & Corne, D. (2002). *Creative evolutionary systems*: Morgan Kaufmann.

Boden, M. A. (1991). *The creative mind: Myths & mechanisms*. London: Wiedenfeld and Nicholson.

Boden, M. A. (2009). Computer models of creativity. *AI Magazine, 30*(3), 23.

Bowden, E. M., Jung-Beeman, M., Fleck, J., & Kounios, J. (2005). New approaches to demystifying insight. *Trends in cognitive sciences, 9*(7), 322-328.

Clancey, W. J. (1997). *Situated cognition: On human knowledge and computer representations*: Cambridge University Press.

Clancey, W. J. (1999). *Conceptual Coordination: How the Mind Orders Experience in Time*: LEA, Inc.

Cohen, P., & Feigenbaum, E. (1982). *The Handbook of Artificial Intelligence*: Addison Wesley.

Colton, S. (2012). The painting fool: Stories from building an automated painter *Computers and creativity* (pp. 3-38): Springer.

Cope, D. (2005). *Computer models of musical creativity*: MIT Press Cambridge.

Cross, N. (2004). Expertise in design: an overview. *Design studies, 25*(5), 427-441.

Dorst, K. (2015). *Frame Innovation*: MIT Press.
Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem–solution. *Design studies, 22*(5), 425-437.
Gärdenfors, P. (2000). *Conceptual spaces: The geometry of thought*: Cambridge, MA, US: The MIT Press.
Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy*. *Cognitive science, 7*(2), 155-170.
Gentner, D., & Forbus, K. D. (2011). Computational models of analogy. *Wiley Interdisciplinary Reviews: Cognitive Science, 2*(3), 266-276.
Gero, J. S. (1990). Design prototypes: A knowledge representation schema for design. *AI Magazine, 11*(4), 26-36.
Gero, J. S. (1994). *Towards a model of exploration in computer-aided design.* Paper presented at the Formal design methods for CAD, Amsterdam.
Gero, J. S. (1998). Conceptual designing as a sequence of situated acts. *Artificial intelligence in structural engineering*, 165-177.
Gero, J. S., & Kannengiesser, U. (2004). The situated function–behaviour–structure framework. *Design studies, 25*(4), 373-391.
Goel, A. K. (1997). Design, analogy, and creativity. *IEEE expert, 12*(3), 62-70.
Grace, K., Maher, M. L., Fisher, D., & Brady, K. (2015). Modeling expectation for evaluating surprise in design creativity *Design Computing and Cognition'14* (pp. 189-206): Springer.
Hofstadter, D. R. (2008). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*: Basic books.
Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
Holyoak, K. J. (1996). *Mental leaps: Analogy in creative thought*: MIT press.
Itti, L., & Baldi, P. (2006). Advances in neural information processing systems. *Bayesian surprise attracts human attention*, 547-554.
Kelly, N. (2011). *Constructive interpretation in design thinking.* (PhD), The University of Sydney, Sydney.
Kelly, N., & Gero, J. S. (2011). *Constructive interpretation in design thinking.* Paper presented at the Computation: The New Realm of Architectural Design - eCAADe 2011, Turkey.
Kelly, N., & Gero, J. S. (2014). Interpretation in design: modelling how the situation changes during design activity. *Research in Engineering Design*, 1-16. doi:10.1007/s00163-013-0168-y
Kelly, N., & Gero, J. S. (2015a). Creative systems that generate and explore. In A. Chakrabarti, T. Taura, & Y. Nagai (Eds.), *DS79: Proceedings of The Third International Conference on Design Creativity* (pp. 96-103). Bangalore: Design Society.
Kelly, N., & Gero, J. S. (2015b). Situated interpretation in computational creativity. *Knowledge-Based Systems, 80*, 48-57. doi:http://dx.doi.org/10.1016/j.knosys.2014.12.005
Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* (Vol. 1): MIT press.
Koza, J. R., Al-Sakran, S. H., & Jones, L. W. (2005). *Automated re-invention of six patented optical lens systems using genetic programming.* Paper presented at the Proceedings of the 2005 conference on Genetic and evolutionary computation.
Kuhn, T. S. (1962). *The structure of scientific revolutions*: University of Chicago press.

Langley, P. (1987). *Scientific discovery: Computational explorations of the creative processes*: MIT press.
Leggatt, J. (2014, 15/12/2014). The cheesiest pizza in the world. *The Guardian*. Retrieved from http://www.theguardian.com/lifeandstyle/australia-food-blog/2014/dec/15/the-cheesiest-pizza-in-the-world-guardian-australia-reviews
Lindenmayer, A. (1968). Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of theoretical biology, 18*(3), 280-299.
Maher, M. L., Poon, J., & Boulanger, S. (1996). Formalising design exploration as co-evolution *Advances in formal design methods for CAD* (pp. 3-30): Springer.
McCarthy, J., & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Readings in artificial intelligence*, 431-450.
McCorduck, P. (1991). *Aaron's code: meta-art, artificial intelligence, and the work of Harold Cohen*: Macmillan.
McCrae, R. R. (1987). Creativity, divergent thinking, and openness to experience. *Journal of personality and social psychology, 52*(6), 1258.
Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*: Addison-Wesley.
Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., McRae, K., & Spivey, M. J. (2013). Computational grounded cognition: a new alliance between grounded cognition and computational modeling. *Frontiers in psychology, 3*.
Qian, L., & Gero, J. S. (1996). Function–behavior–structure paths and their role in analogy-based design. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, 10*(04), 289-312.
Ritchie, G. (2012). *A closer look at creativity as search.* Paper presented at the International Conference on Computational Creativity, Dublin.
Runco, M. A. (1991). *Divergent thinking*: Ablex Publishing.
Runco, M. A., & Jaeger, G. J. (2012). The standard definition of creativity. *Creativity Research Journal, 24*(1), 92-96.
Schacter, D. (1987). Implicit memory: History and current status. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*(3), 501-518.
Schon, D., & Wiggins, G. (1992). Kinds of seeing and their functions in designing. *Design studies, 13*(2), 135-156. doi:citeulike-article-id:8497732
Schön, D. A. (1983). *The reflective practitioner: How professionals think in action* (Vol. 5126): Basic books.
Schön, D. A. (1984). Problems, frames and perspectives on designing. *Design Studies, 5*(3), 132-136.
Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences, 3*(03), 417-424.
Seelig, T. L. (2012). *inGenius: A crash course on creativity*: Hay House, Inc.
Simon, H. A. (1969). *The Sciences of the Artificial* (Vol. 136): MIT press.
Stein, M. I. (1953). Creativity and culture. *The journal of psychology, 36*(2), 311-322.
Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and planning B, 7*(3), 343-351.
Suwa, M., Gero, J., & Purcell, T. (2000). Unexpected discoveries and S-invention of design requirements: important vehicles for a design process. *Design studies, 21*(6), 539-567. doi:10.1016/s0142-694x(99)00034-4
Suwa, M., & Tversky, B. (1997). What do architects and students perceive in their design sketches? *Design studies, 18*(4), 385-403.

Sygall, D. (2007). How Australia's top scientist earned millions from Wi-Fi. *The Sydney Morning Herald*. Retrieved from http://www.smh.com.au/technology/sci-tech/how-australias-top-scientist-earned-millions-from-wifi-20091207-kep4.html

Tschacher, W., & Scheier, C. (1999). The perspective of situated and self-organizing cognition in cognitive psychology. *The nature of concepts: Evolution, structure and representation*, 68-90.

Vattam, S., Helms, M. E., & Goel, A. K. (2010). A content account of creative analogies in biologically inspired design. *AI EDAM, 24*(4), 467-481.

Von Uexküll, J. (1957/1992). A stroll through the worlds of animals and men: A picture book of invisible worlds. *Semiotica, 89*(4), 319-391.

Wiggins, G. A. (2006a). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems, 19*(7), 449-458.

Wiggins, G. A. (2006b). Searching for computational creativity. *New Generation Computing, 24*(3), 209-222.

Yu, R., Gu, N., Ostwald, M., & Gero, J. S. (2015). Empirical support for problem–solution coevolution in a parametric design environment. *AI EDAM, 29*(01), 33-44. doi:doi:10.1017/S0890060414000316

Ziemke, T., & Sharkey, N. E. (2001). A stroll through the worlds of robots and animals: Applying Jakob von Uexkull's theory of meaning to adaptive robots and artificial life. *Semiotica, 134*(1/4), 701-746.