INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

A COGNITIVE AND COMPUTATIONAL BASIS FOR DESIGNING

John S Gero[1] and Gregory J Smith[2]
[1]Krasnow Institute for Advanced Study and Volgenau School of Information Technology and Engineering, George Mason University, USA
[2]Key Centre of Design Computing and Cognition, University of Sydney, Australia

ABSTRACT
This paper presents a set of concepts that provides the cognitive and computational foundations for answering fundamental questions about designerly behaviour. A design agent should be dynamic and able to handle changes in external representations that describe its designs and changes in how it is guided by its past experiences. Such a view of agency is afforded by situated design computing and can represent and explain much designerly behaviour. It can model how a designer can commence designing before all the requirements have been specified, how two designers presented with the same specifications produce different designs, how the same designer later confronted with the same requirements produces a different design to the previous one, and how a designer can change their design trajectory during the activity of designing. This view is built upon three foundational concepts: knowledge grounded in interaction, constructive memory, situatedness. The concepts build on notions of memory that are often traced back to Dewey and Bartlett, although we use contemporary descriptions. Memory is not understood primarily as allowing for the retrieval of an object from a data store by knowing its physical location; it is guiding an experience in a fashion similar to how past experiences progressed, and recognising that this is so. The paper explicates experiences and situations and discusses implementation requirements.

Keywords: design agent, constructive memory, experience, situatedness

1 INTRODUCTION
How is it possible that a designer can commence designing before all the requirements have been specified? How is it that two designers presented with the same specifications produce different designs? How is it that the same designer later confronted with the same requirements produces a different design to the previous design? How is it that a designer can change their design trajectory during the activity of designing. Designing is often characterized by abstractness and an incomplete understanding of the problem and solution. Designers cope with this by exploring the space of requirements as they explore the space of possible conceptual designs. A design agent should therefore be dynamic and able to handle change, both in external representations that describe its designs and in how it is guided by its past experiences. Such a view of agency is afforded by situated design computing. This is built upon three foundational concepts: knowledge grounded in interaction, constructive memory and situatedness.
This paper presents a set of concepts that provide the cognitive and computational foundations for answering these fundamental questions. Their bases are experiential notions of constructive memory and situatedness. Memory in a computational system is usually taken to be a place filled with things called "memories". A place is indexed by either knowing its physical location or its content. For situated design agents, however, memory is a reflection of how the system has adapted to its environment. Recollection should be more than looking up records - it should be past experiences guiding active ones. These ideas are often traced back to Dewey and Bartlett, although we use contemporary descriptions. Dewey described the quality of an experience as having two aspects called continuity and interaction:

"The principle of continuity of experience means that every experience both takes up something from those which have gone before and modifies in some way the quality of those which come after" [1]

"Experience does not simply go on inside a person. It does go on there ... but this is not the whole of the story. Every genuine experience has an active side which changes in some degree the objective conditions under which experiences are had" [1]

2 WHAT ARE EXPERIENCES?

An experience is an interplay of continuity and interaction. If there are experiences there must be agents. In this work we take humans interacting with computational systems and artificial agents to also be agents. We denote agents as α_1, α_2,… and the environment as ξ. The environment need not only contain agents so we say that an environment is composed of entities, some of which are agent-entities (agents) and some of which are not. An entity cannot be an agent unless it is embodied in some environment, be it of our world or of a virtual world. As agents are embodied, α_1, α_2,… are a part of ξ, and α_1, α_2,… are distinct from each other and from ξ.

We say that the non-agent entities of the environment are thing-entities (things) that we denote γ_1, γ_2,… For things, γ_1, γ_2,… are a part of ξ, and γ_1, γ_2,… are distinct from each other and from ξ and α_1, α_2,…

As Dewey [2] noted, an experience is not of a disembodied agent (although Dewey would not have used the word "agent"). It is to do with interaction of the agent with an environment. An experience is not something static; it is dynamic and is of certain kinds of entities that are coupled to their environment. This "entities ... coupled to their environment" is like Dewey's experience changing "the objective conditions under which experiences are had" [1]. This is illustrated in Figure 1, and it shows two kinds of coupling. The "body" of agent a_i is an agent-thing, say an α_i that is a part of ξ. The "nervous system" of agent a_i are construct-entities (constructs) $\{\beta_i^1, \beta_i^2,\ldots\}$. The quotes are because the words "body" and "nervous system" are those used in [3] but which we avoid. Construct-entities are parts of agent-things, so each β_i^j is a part of α_i. The agent a_i is an agent-entity that is the composition of an agent-thing α_i and constructs $\{\beta_i^j\}$. That part of the environment that is not the agent is $\xi - a_i = \xi - (\alpha_i \cup \{\beta_i^j\})$.

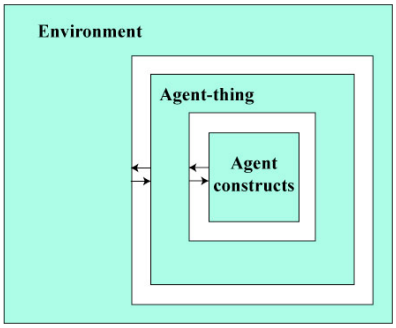

Figure 1. Agent coupled to the environment (figure derived from [3]).

A coupling between ξ and an agent-thing α_i is an exogenously generated experience (an e-experience) of a_i. An example is robot navigation experiences involving sonar sensual experiences and motion effectual experiences. A coupling between the agent-thing α_i and agent constructs $\{\beta_i^j\}$ is an autogenously generated experience (an a-experience) of a_i. An example is a human moving their arm, involving sensual experiences of proprioception and motor effectual experiences. Further, e-experiences and a-experiences may perturb each other directly or indirectly. An e-experience perturbs an a-experience if the agent interprets that e-experience. An e-experience fails to perturb an a-experience if the agent ignores it. An a-experience perturbs an e-experience when the agent acts on its environment - when that agent perturbs other agents or things.

We denote an experience of agent a_i as e_i^k. If an e-experience is able to perturb an a-experience and vice versa, an agent must be able to have multiple concurrent experiences $\{e_i^1, e_i^2,\ldots\}$. An e-experience involves entities perturbing each other, where one of the entities is an α_i and the other is either another agent α_j or a thing γ_m. An a-experience also involves entities perturbing each other,

where one of the entities is an α_i but where the others are constructs $\{\beta_i^j\}$. In order for one to perturb another there must be either

- some point at which they synchronize, such as one computational process synchronously passing a message to another, or
- some intermediary on which each synchronizes, such as one computational process asynchronously passing a message to another.

For one experience to perturb another requires two things: that experiences can have parts that are themselves experiences, and that some experiences can be parts of multiple experiences. We can describe this using mereological relations on processes [4]. These mereological relations are useful as a means to describing properties of experiences without the descriptions necessarily being reductionist.

An experience can be a part of another experience. An experience that is a part of, but not identical to, another experience is a proper part. An experience with no proper parts is atomic. If the experience is temporally atomic but spatially not, we call it an event. If the experience is spatially atomic we call it an entity. Two experiences with one or more common parts are said to overlap. Experiences that perturb each other must overlap and are disjoint if they never overlap. An experience e_i^x is emergent from experiences $\{e_i^y | y \neq x\}$ if e_i^x is a part of the sum of $\{e_i^y | y \neq x\}$ but no part of e_i^x (including itself) is a part of any e_i^y for $y \neq x$ (see [1] where this is defined more precisely). An experience starts when the agent activates an action that is qualitatively different from active experiences. The action may be that a person changes their visual focus of attention: agent constructs β_i^j sufficiently to perturb the agent-thing α_i to change what from its visual field it is looking at, triggering a new a-experience. The action may be that the robot moves in the world: agent constructs β_i^j sufficiently to perturb the agent-thing α_i to trigger a new a-experience, and the new a-experience perturbs the environment by shifting the robots location in it, triggering a new e-experience.

3 THE ROLE OF PAST EXPERIENCES

Let a_i be an agent in environment ξ. So α_i is a part of ξ and each of $\{\beta_i^j\}$ is a part of α_i. We denote the type of experiences of a_i as E_i such that $e_i^1, e_i^2 \ldots \in E_i$. e_i^x is an experience of a_i if it is of the agent-entity that is the composition of α_i and $\{\beta_i^j\}$. e_i^x is e-experience if it is perturbed by or perturbs one or more entities not part of a_i. e_i^x is an a-experience if it is perturbed by or perturbs one or more entities from a_i but is not perturbed by and does not perturb any entities not part of a_i. When one entity perturbs an experience of another we say that there is an effect on that entity. An effect by α_i on $(\xi - a_i)$ is via an effector of agent a_i and is of an e-experience. An effect by $(\xi - a_i)$ on α_i is via a sensor of agent a_i and is of an e-experience. Effects that are of a-experiences are internal to the agent and are via perceptors, conceptors and action activators.

The role of past experiences on active ones is central to what a constructive memory is about. Dewey again:

> "We have had experiences; these exist stored up, in some unexplained way, in the mind, and when some experience occurs which is like some one of these, or has been previously contiguous with it in time or space, it calls this other up, and that constitutes memory. This, at most, solves but one half the problem. The association of ideas only accounts for the presence of the object or event. The other half is the reference of its present image to some past reality. In memory we re-cognize its presence; i.e., we know that it has been a previous element of our experience." [5]

An agent should recognise in an active experience something of the trajectory, or history, or both of past experiences and use these to project forward. This is continuity. The trajectory of e_i^x at time t is how e_i^x came to be what it is at t. What perturbations of e_i^x there have been up until t is the history of e_i^x until t. Recognising the continuity of experiences is what we call memory. It is guiding an experience in a fashion similar to how past experiences progressed, and recognising that this is so. Memory is not retrieving an object from a data store. It is experiences being guided in familiar ways.

If an experience is to do with continuity and interaction, what is required of an agent to facilitate this?

Now for continuity something persists, and for interaction something changes. So experiences have a temporal aspect but they cannot be solely temporal. Suppose we want to look closely at an experience and see what this "something" of experiences is, so we fixate on an experience at a particular time

$t \in \tau$. This fixation is a function from e_i^x onto some space that we denote as N_i. As we have only fixed a time, the result is a subspace of reduced dimension: $\dim(t) \geq 1$ and $\dim(N_i) = \dim(e_i) - \dim(t)$. If e_i^k is from the space E_i, this $n \in N_i$ is from a hyperplane that is a subspace of E_i. A trajectory is these n changing over time. Each hyperplane will contain "somethings" that have meaning to the agent, so we call the subspace N_i the space of notions of a_i. We use the word "notion" to maintain independence from any particular kind of agent representation.
The concepts, percepts, acts, sense-data and effect-data of an agent are all subspaces of N_i. Any subset of N_i is a notion, including $\varnothing$ and N_i itself, as is the intersection of any two notions. As such, any given notion n will be a subset of N_i, or $n \in N_i$. Further, that n may itself contain other notions that are themselves both a subset of n and a subset of N_i.
Memory of an experience as "being guided in familiar ways" is temporally twofold. The first is projecting active experiences into the future:

> "experience in its vital form is experiental, an effort to change the given; it is characterized by projection, by reaching forward into the unknown" [2]

The second is recognising having previously had a similar experience:

> "reference of its present image to some past reality. In memory we re-cognize its presence; i.e., we know that it has been a previous element of our experience" [5]

Projections into the past and future are not against recordings of experiences, they are against reconstructions of experiences. An important idea of Bartlett is relevant here:

> "Remembering is not the re-excitation of innumerable fixed, lifeless and fragmentary traces. It is an imaginative reconstruction, or construction, built out of the relation of our attitude towards a whole active mass of organised past reactions or experience, and to a little outstanding detail which commonly appears in image or in language form." [6]

If a memory is an experience being guided in a familiar way, but the agent has adapted to other experiences since that familiar experience was active, then what the agent remembers at the later time may not be what was originally the case. Let the active experiences of agent a_i at a time t be $\hat{e} = \{e_i^k\}$. If the agent recollects these at some later time:

- one or more e from $\hat{e}$ may be different than it was at the original time
- the order of one or more e from se may be different than it was at the original time
- one or more e from $\hat{e}$ may not be recollected at the later time
- one or more e not from $\hat{e}$ may be recollected at the later time

So a perturbation may trigger the recall of an experience but that perturbation may be understood differently when recalled. One reason is that the agent adapts to subsequent experiences, hence recollections of what was experienced before may be different. Another is that memories of experiences are dynamic and interlinked, so a recollection is a reconstruction rather than a lookup. Another reason is that the agent adapts to subsequent experiences, hence some recollections of what was experienced before may be interpreted differently. Finally, one reason is the role of the current situation, which is elaborated in Section 4.
If we *want* an agent to be constructive and only constructive then we would need to force it to behave only in the way described. We *could* give it a Google-like memory that just records everything that happens but using such a memory would not be constructive. It may be satisfactory for an agent to sometimes be constructive and sometimes not. Perhaps sometimes it needs to be creative, other times it needs to recall facts. The distinction is between subjective and objective memories. Subjective memories are the kinds of reconstruction that we have been describing. They are of what it feels like to be this agent. Examples of objective memories are "a magnetic flux density of 1 tesla is 1 weber of magnetic flux per square metre", "an action is always opposed by an equal reaction" and "I owe Mungo $2.50 for coffee yesterday". It is for objective memories that an artificial agent with a constructive memory may sometimes also want non-constructive recall. But having objective and subjective memories does not necessitate an artificial agent having distinct or separate memory mechanisms.

4 WHAT DO SITUATIONS LOOK LIKE?

The mechanism of interaction is e-experiences coupling agent-things α_i and non-agent-things γ_m to other agent-things α_j. We say that such interactions are of agents situated in their environment, but what does this mean? A situation is a constructed characteristic of a system of interacting agents and things, like the patterns in the turbulent flow in Figure 2. These patterns are emergent or supervenient on the structure and behaviour of the river bed, the river bank, obstacles in the river, the wind, and the water. They may arise from interactions between non-agent things, between agents, between agents and things, between constructs in an agent, and between an agent-thing and its constructs.

Figure 2. Complex patterns in water flow

An agent may point a sensor at some location in an environment, the sensor receives a stream of data from the environment, and the agent interprets it. As interpretation requires perception, only a part of an experience of an entity will be of what was sensed. If perception is constructed then a memory of it will also be constructed, at least in part. Recollection involves a reconstruction of past experiences based on information in the current environment and on the way cognitive processing is currently accomplished. The idea is of the perceiver of an entity in the environment as being like the captain of a submarine [7]: they have knowledge of the medium in which they are submerged but they cannot experience it directly. That being so, it must be the agent that individuates what is in the environment and situations must be interpretations of an agent. The complexity evident in Figure 2 may or may not exist without an agent to interpret it but will not be what we call "situations" as situations will not exist without an agent.

A metaphor will illustrate this distinction. If I strike a tuning fork, waves of air pressure result. These waves are not sounds. Sound requires a sensor (an ear or microphone) and perception, and interpreting sequences of sounds as music requires further reasoning. Recognising that there is a situation is like recognising that there is music, and so requires an agent. The agent affects entities in the system, new sense-data get interpreted, resulting in changes to the situation. This is, to risk stretching another analogy too far, a little like Heisenberg's uncertainty principle: the act of observation changes the system and so the situation. The Copenhagen interpretation of quantum mechanics says that an electron does not exist until something registers its existence [8]. This can be taken in two ways. The first takes this non-existence at face value: an electron *really does not exist* until there is something that registers that existence. It is a metaphor for a radical form of constructivism that begins at an agent's sensors and ends at its effectors. A less radical interpretation is to consider anything not empirically testable as being beyond scientific theory, and so the electron does not exist *from the viewpoint of quantum mechanics as a scientific theory* until the electron is observed. This is a metaphor for a less radical form of constructivism: the environment beyond an agent may well have some independent, objective existence but it is not accessible to the agent except via sensors and effectors. We prefer the less radical metaphor.

If that which we call "the current situation" is a representation by an agent, what is it representing? It is a representation that influences how the world is viewed. An experience may be understood differently when recalled, and part of the reason for that is the changed situation. Notice that the situation is not "a view of the world"; it is a process that changes how those notions behave. Consider an e-experience $e_i^k(t)$ as an example. As it is an e-experience it involves the perturbation of a thing-entity or of another agent. The trajectory for $e_i^k(t)$ is illustrated in Figure 3 drawn over a time interval

(t^-,t^+) starting from a perturbation (a "query" on the experience) at t^- and ending at an equilibrium (a "result") at t^+. Shown are trajectories from two similar initially perturbed notions $n,n' \in N_i$ are shown.

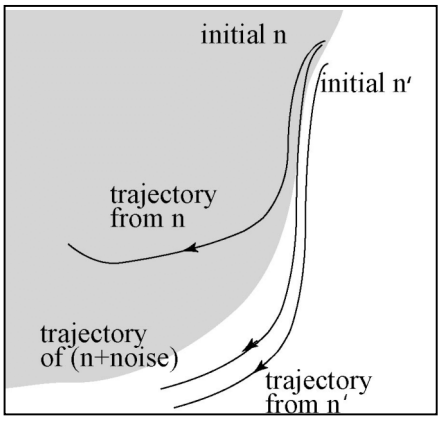

Figure 3. Trajectories of similar queries

Each perturbation upsets the equilibrium of the experience, and the behaviour of the memory is to try to re-settle to equilibrium. This experience will also perturb other experiences, and the equilibrium is settled with respect to the current situation. So a small change to the initial perturbation or to the current situation can result in a different eventual equilibrium, and hence a different interpretation. This idea is often described as different schemas. The trajectories shown in the example are through two similar schemas: one indicated by the grey background and one indicated by the white background. Each of these - the white and the grey - are more general notions and the trajectory arcs are through more specific notions, so the schemas are abstract interpretations of the perturbation. The first trajectory is from an initial excited notion n, settling into the grey schema. The second is from an initial n' and settles into the white schema. The third illustrates the idea that a small change to the starting point (in the example it is starting from a noisy n), a change to the situation before the query has settled, or a new perturbation before the former has settled can result in a large change in where it ends up, including being in a different schema. Such a change can alter the focus of attention of an agent, or even be a "Eureka" moment. We shall, however, avoid using the word "schema" as it has been applied so variously that we risk implying something unintended to some readers. Not reaching equilibrium within some computational bounds triggers the agent to adapt such that a similar perturbation in future will find equilibrium. This means that influences between notions may change, or that the space of notions known to the agent may change.

The agent may have multiple concurrent experiences. These experiences are not hidden off in isolated compartments. They are all couplings of the same agent-thing, the same agent-constructs, and the same part of the environment. They may be modular but they still need to influence each other. So if the situation is experiences influencing each other, changing how the world is viewed, then situations are processes.

This means that we need a way to describe agent processes that lets each relevant experience influence a target experience. Let a target experience be that experience if the role of the situation is ignored, and call it $e_i^{k\varnothing}$ (the un-situated experience). The experience e_i^k is the target experience after the influence of the situation with respect to one or more other experiences are included. That is, an experience is situated unless it is explicitly denoted otherwise such as in $e_i^{k\varnothing}$. The type of situations is Ψ such that $\psi^1,\psi^2,\ldots \in \Psi_i$. The type of experiences of a_i is E_i, so situations are functions Ψ_i: $N_i \rightarrow E_i \rightarrow E_i$. The current situation as seen by an experience $e_i^{k\varnothing}$ is one or more functions Ψ_i that each use another experience to influence this one. The idea is of the representation of the current situation arising from expectations, so those "other experiences" are of more abstract notions. Given this, the idea of situations influencing experiences can be described as follows:

- An experience $e_i^{k\varnothing}$ will be computed by one or more constructs $\hat{c}$
- At the current time t, $\hat{c}$ will involve notions n from N_i that vary in generality or abstractness
- If we partition n into one or more layers of similar generality, some layers will precede one or more other layers

- Each layer constructs a situation of type Ψ_i that applies to layers that precede it
- Situations that apply to $e_i^{k\varnothing}$ at time t are applied to give e_i^k over temporal extent (t, t+δt).

5 MOVING TOWARD AN IMPLEMENTATION

Employing an agent development framework that implements a multi-agent systems standard (which complies with FIPA agent specifications) would seem a natural choice for an implementation. Such a framework would certainly encourage the use of our implementation. Unfortunately, while it may provide the communications infrastructure we need, it would not help with the constructive, situated, experiential memory that is our central concern. So we will assume here the use of such a framework and will instead focus on what is required when implementing the memory.

Firstly, an implementation must be scalable beyond toy demonstration examples to cover, in the long term, larger scale designing problems. To that end we are building the implementation on Numenta's NuPIC [18]. This is a distributed network of nodes employing Pearl's message passing technique for Bayesian belief propagation. A node will implement a set of notions that are computed similarly from similar inputs, and so constitute a layer. The idea is that nodes receive messages with probability changes, such as new evidence from a sensor update, from nodes upon which they depend. Upon receiving a message a node updates its state and then forwards on its own messages.

One implementation issue regards situations. Two principles are key here:

- An agent and/or its memory may behave differently in different situations or after different experiences, and
- A situation is not "a view of the world" but instead is a process that changes how notions behave.

If *S* is "that which determines situations" and *M* is "that which determines notions", there are four possibilities:

[S1] That which determines situations is identical to that which determines notions. From the "inside" of the memory there is nothing to distinguish parts of situations (of *S*) from parts of notions (of *M*). They and their algorithms are identical. They are not distinguished except by we omnipotent memory builders that can look at what we have built and label some of them as being of situations and others as not.

[S2] That which determines situations is like that which determines notions. Whatever computes *S* is distinguished from whatever computes *M* but they are of the same kind. For example, for perception there may be a network for perception, a network for situations, and these would be interlinked.

[S3] That which determines situations changes whatever that which determines notions produces. *M* produces something *Y*, but *S* changes that *Y* before it can be seen by the agent or by the world.

[S4] That which determines situations changes how that which determines notions does so. *S* changes how *M* operates.

Both [S1] and [S2] are valid approaches to building memories for agents that many would call situated but neither is likely to result in an agent that is situated in the manner described above. Our implementation is of [S4], with a fallback of [S3].

Another implementation issue regards the nature of experiences. The key principle is of memory guiding an experience in a fashion similar to how past experiences progressed, and recognising that this is so. Explicit guidance is other experiences changing the trajectory of an experience at the time of recall; implicit guidance is other experiences causing adaption such that the trajectory is different than it would have been otherwise. Explicit representation is where we can point to a concrete implementation artefact that represents what an experience is and directs how it should behave.

Implicit representation is where we third person omnipotent observers can look at the behaviour of the memory and describe that behaviour using the idea of experiences. So again there are four possibilities:

[E1] Implicit representation and implicit guidance. An experience is only a trace through a space of notions, and past experiences cause adaption in that space.

[E2] Implicit representation and explicit guidance. An experience is a trace through a space of notions except that the traces can influence each other. This is like [E1] except driven directly, where "the explicit guidance".

[E3] Explicit representation and implicit guidance. Experiences are threads through a shared address space and this address space is of notions, but the influence of threads on each other is indirect and

implicit by threads somehow warping the underlying space. Think of relativity, where notions are like masses that warp the space that threads use.

[E4] Explicit representation and explicit guidance. Experiences are threads through a shared address space and these threads explicitly influence each other. Think of a control system but with a past experience instead of a set-point and with feedback causing adaption, or signal processing formalised as data flow computation.

The answer to whether there is required to be an explicitly implemented thing called "an experience" is no. The answer to the question "are experiences just the way we describe memory behaviours" is yes. To this end we need not implement [E3] or [E4] as described; they are best used as intuitions and formalisms for describing certain kinds of observed memory behaviour. Our implementation is of [E2], with a fallback of [E1].

Our approach is to build an initial implementation using supplied NuPIC nodes, replacing some of these with our nodes implemented as described. Each node will be a vector of belief variables that are treated similarly: inputs from same other node(s), outputs to same other node(s), same algorithms within a node. Nodes may be referred to as layers, numbered 1..L, but this need not imply a requirement for a strict hierarchy. Each node has a state that is a vector of random variables Θ^l. Each node accepts a input variables X^l from lower layers; this may be the concatenation of outputs from a number of lower layers. The output of a node is Y^l. Each node may accept hypotheses variables H^l from higher layers. The hypotheses at top-most layers are set by the agent externally to "the memory" such as to change goals. Belief propagation is triggered by a change at the bottom (such as from new sense data) or at the top (such as a changed most general hypothesis). This corresponds to a data push (here is a new input – update your beliefs) and a data pull (here is what I am looking for – make your beliefs look like this). Bayes' Rule is central here, whether the belief propagation uses Pearl's message passing or a particle filter algorithm such as in [19]. If Θ^l is learned so as to do a bottom up push, driving this backwards via hypotheses using Bayes' Rule is abductive. For example, suppose we learn a model p($e|X$) of how likely we are to observe certain evidence given that an object is in a certain state, and we drive this with new evidence. Computing how likely a particular object state is given particular evidence is driving this backwards as p($X|e$). Driving it backwards is abduction, and Bayes' Rule says how to do it. Regarding learning Θ^l for a hierarchy, we could attempt to learn one layer at a time as the NuPIC nodes HTM do, or by using expectation maximisation one layer at a time. Kruschke [20] also shows how to propagate class labels down through a hierarchy during learning.

6 SITUATED DESIGNING

How do these ideas apply to agents and design, and how do they help with the questions posed at the start of this paper? We posit that a design agent should individuate entities as being in the environment only when the situation is such that the agent ought to distinguish those entities from others. That is, an agent has a partial view of the environment, and that view depends on expectations which are past experiences guiding current ones in the current situation. This is Dewey's continuity. The importance of Dewey's interaction to a situated agent becomes especially evident when considering perception and action on external representations. A few examples will illustrate, consider the sketches in Figure7.

Why is the leftmost sketch in Figure 4 a sketch of a church, and not just a set of scribbles? Why is the centre sketch a conceptual drawing of a house among trees, and not just a set of scribbles? Why is the rightmost image a sketch of a site layout or an interior layout? Maybe it really is just a set of scribbles? In each case our perception depends just as much on background knowledge and expectations as on sense-data. All of these may be considered to be external representations, and all are clearly abstractions. For designers such abstractions maintain the interactive nature of external representations while still being conceptually flexible and "unrestrictive to thinking processes" [12] so as to encourage reinterpretation.

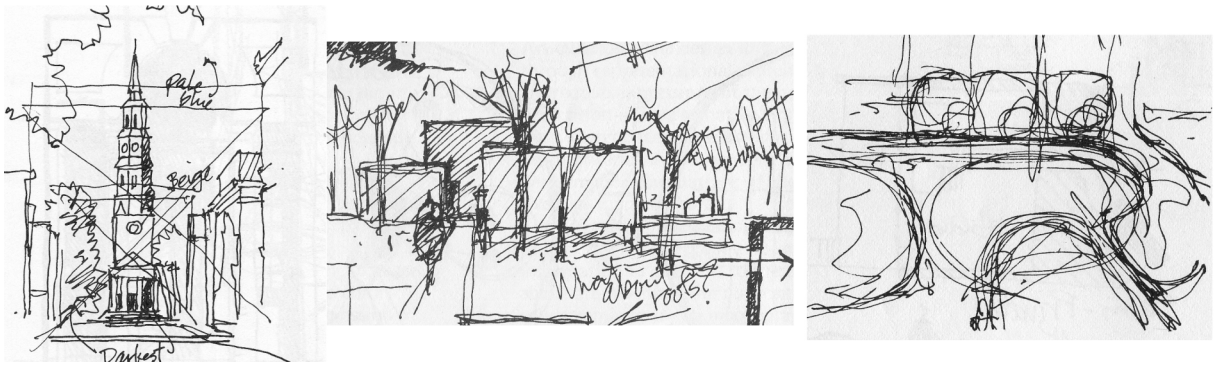

Figure 4. Examples of design interpretation. The left sketch is from [11], the others are from[12].

The leftmost image in Figure 5 shows a cup that is a thing located in the environment.

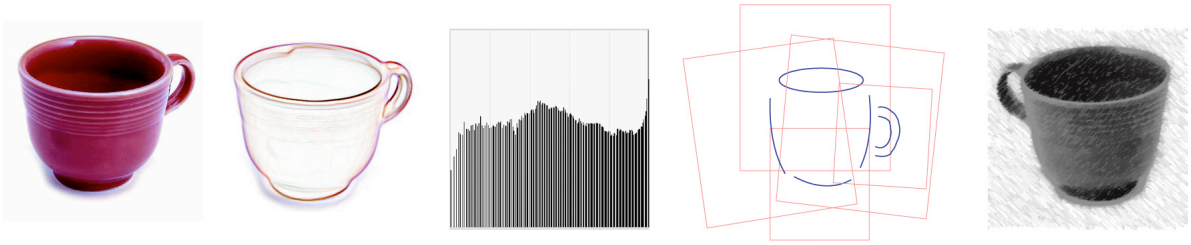

Figure 5. Perception of a cup. Leftmost is a cup in view of a designer. Next are illustrations of edge detection and colour sense-data from the cup. Second rightmost are expectations of "cup-ness" (this image is after [13]). Rightmost is a sketch of cup.

For the example assume that the agent is designing new cups. The next two leftmost images are of a sensor that does edge detection and a sensor that does colour detection. Perception will have expectations that are characteristic aspects of "cup-ness", shown in the second rightmost image in the manner of Nelson and Selinger's [13] cubist metaphor. That is, that interpreting the category of things that are cups can be approached not as constructing a single, homogeneous object but as a small number of key properties with local context that are assembled in a loose global context. These sense-data drive perception bottom-up but at the same time their interpretation relies on biases and expectations.

As another example of applying constructive memories, consider analogies and metaphors. Metaphors are analogies that require inference on what the properties may represent rather than a simpler analogical matching. The cliché "money is the root of all evil" involves concepts reconstructed from "money", "root" and "evil", but having these constructs still has us a long way from understanding the metaphor. Understanding it requires inference.

In the leftmost image of Figure 9 the annotation indicates one way that the two depicted entities may be similar: that the tap root of the tree is similar to the support of the building.

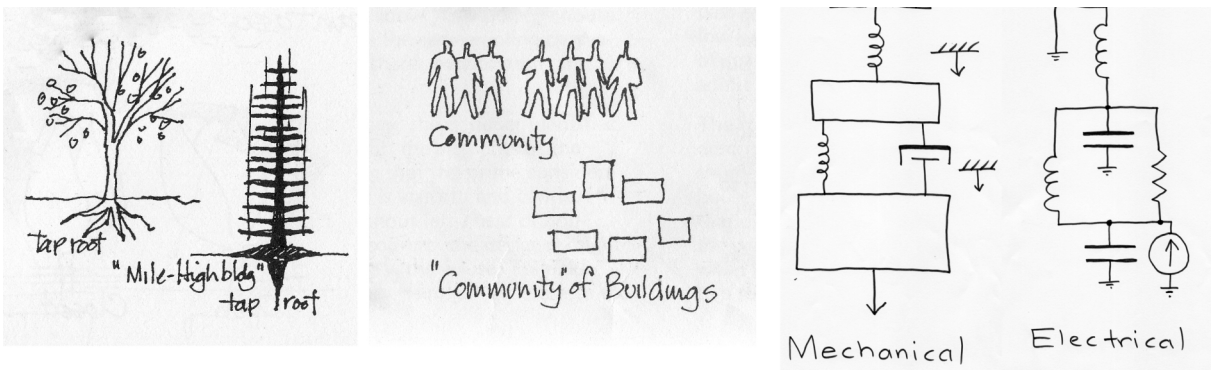

Figure 6. Analogies and metaphors. The two leftmost images are from [12].

This could be in terms of structure (descends underground below the entity) or behaviour (supports the entity to stop it from falling over). Looking at its left side results in a perceptual interpretation, and hence concepts, of what a tap root structure and behaviour are. This memory results in properties appropriate to that sketched tap root. But some of these properties apply analogically to buildings also, so we have an analogy from a perceptual interpretation to another perceptual analogy. Also, having found the analogy may cause others to be recollected such as between the tree trunk and the structures around the elevator shaft. The centre image shows an analogy from a concept (that of community) to drawing actions that cluster buildings and hence to expectations for perceptual interpretation. The rightmost image visualises a conceptual analogy from one representation of a mechanical artefact (as a mass-spring-damper system) to an equivalent electrical representation (because the behaviours of the two have the same differential equations and their components have analogous behaviours).

Analogy is important to designing [14] and to creativity [15], and one approach to creative problem solving is to deliberately use analogy as a route to lateral thinking. But constructive memory means that this need not always be deliberate. Two things that constructive memory models of humans must account for are false recognition and intrusions [16]. False recognition is where a person claims that a novel word or event is familiar, and an intrusion is the production of novel information. That is, false recognition is a false memory on the perception side of the agent and intrusion is a false memory on the action side. But to creative designers these effects may not be disadvantageous. Perhaps some part of lateral thinking is the reconstruction of memories "incorrectly"?

7 CONCLUSION

Designing as an activity is not well understood. As a consequence our ability to develop tools to support designing is very limited. This work aims to create the foundations of models of designing based on both cognitive and computational studies. In particular we draw concepts from the constructivist view of cognition and develop an approach that has as its foundation the twin concepts of constructive memory and situatedness embodied in computational agents. Such a view of design agency can represent and explain much designerly behaviour. It can model how a designer can commence designing before all the requirements have been specified, how two designers presented with the same specifications produce different designs, how the same designer confronted with the same requirements at a later time produces a different design to the previous one, and how a designer can change their design trajectory during the activity of designing. These are all characteristics that are essentially designerly and not part of problem solving alone.

ACKNOWLEDGEMENTS

This research is funded by the Australian Research Council, grant number DP0559885.

REFERENCES

[1] Dewey J. *Experience and Education*, 1938 (Collier, reprinted in 1963).

[2] Dewey J. The need for a recovery of philosophy, In *Creative Intelligence: Essays in the Pragmatic Attitude,* 1917, pp. 3-69 (Henry Holt and Company). Reprinted in John Dewey: The Middle Works, 1899-1924, Vol. 10: 1916-1917, 1985, pp. 3-49 (Southern Illinois University).

[3] Beer R. D. The dynamics of active categorical perception in an evolved model agent, *Adaptive Behavior*, 2003, 11,(4), 209-243.

[4] Seibt J. Free process theory: Towards a typology of occurrings, *Axiomathes*, 2004, 14, 23-55.

[5] Dewey J. *Psychology*, 1887 (Harpers & Brothers). Reprinted in John Dewey: The Early, 1882-1898, Vol. 2: 1887, 1969, pp. 3-21 (Southern Illinois University).

[6] Bartlett F. C. *Remembering: A Study In Experimental And Social Psychology*, 1932 (Cambridge University Press).

[7] Gordon I. E. *Theories Of Visual Perception*, 1997 (John Wiley and Sons).

[8] Wheeler J. A. *At Home In The Universe*, 1994 (American Institute of Physics).

[9] Schön D. A. and Wiggins D. Kinds of seeing and their functions in designing, *Design Studies*, 1992, 13(2), 135-156.

[10] Smith G. J. and Gero J. S. What does an artificial design agent mean by being `situated'?, *Design Studies*, 2005, 26, 535-561.

[11] Crowe N. and Laseau P. *Visual Notes for Architects and Designers*, 1984 (John Wiley and Sons).
[12] Laseau P. *Graphical Thinking for Architects and Designers*, 1980 (Van Nostrand).
[13] Nelson R. C. and Selinger A. A cubist approach to object recognition, *International Conference on Computer Vision (ICCV98)*, 1998, pp. 614-621.
[14] Qian L. and Gero J. S. Function-behaviour-structure and their roles in analogy-based design, *AIEDAM*, 1996, 10, 289-312.
[15] Runco M. A. and Pritzker S. *Encyclopedia of Creativity*, 1999 (Academic Press).
[16] Riegler A. Constructive memory, *Kybernetes*, 2005, 34, 89-104.
[17] Bristow-Johnson R. *Wavetable Synthesis 101: A Fundamental Perspective*. 1996, http://www.musicdsp.org.
[18] Numenta. *Numenta platform for intelligent computing: Programmer's guide*, Version 1.0.1, March 2007, http://www.numenta.com.
[19] Lee T. and Mumford D. : 2003, Hierarchical Bayesian inference in the visual cortex, *Journal of the Optical Society of America A: Optics, Image Science, and Vision,* 2003, 20, 1434-1448.
[20] Kruschke J. K: 2006, Locally Bayesian learning with application to retrospective revaluation and highlighting, *Psychological Review*, 2006, 113(4), 677-699.

Contact: John S. Gero
Krasnow Institute for Advanced Study and
Volgenau School of Information Technology and Engineering,
George Mason University,
VA 22030 USA
john@johngero
http://mason.gmu.edu/~jgero/