Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2004), *18*, 163–198. Printed in the USA.
Copyright © 2004 Cambridge University Press 0890-0604/04 $16.00
DOI: 10.1017/S0890060404040120

Constructive memory for situated design agents

PAK-SAN LIEW AND JOHN S. GERO

Key Centre of Design Computing and Cognition, School of Architecture, Design Science and Planning, University of Sydney, Chippendale, NSW 2006, Australia

(RECEIVED June 3, 2003; ACCEPTED June 28, 2004)

Abstract

Design is situated. "Situatedness" in designing entails the explicit consideration of the state of the environment, the knowledge and experiences of the designer, and the interactions between the designer and the environment during designing. Central to the notion of situatedness is the notion of design situation and constructive memory. A design situation models a particular state of interaction between a design agent and the environment at a particular point in time. Memory construction occurs whenever a design agent uses past experiences and knowledge within the current design environment in a situated manner. This paper is concerned with the development of an agent-based computational design tool that takes into consideration the notion of situatedness in designing. A key element of this tool is a constructive memory system that supports the dynamic nature of designing. Memories of past experiences are constructed as required by the current situation, and past experiences are refined for future utility according to the current interactions between the agent and the environment. This latter case of knowledge improvement is illustrated through a series of experiments that demonstrates the effect of grounding on the operating modes and responses of a constructive memory system.

Keywords: Constructive Memory; Situated Design Agent

1. INTRODUCTION

Designing can be characterized as "a goal-oriented, constrained, decision-making, exploration and learning activity that operates within a context that depends on the designer's perception of the context" (Gero, 1990). These characteristics give rise to the unpredictability of designing (Goldschmidt, 1997), with the implication that it is not possible to encode all the required information into a design tool in a state suitable for their utility when the tool was first created. The dependency on the current state of the environment within which a tool operates, the current state of the tool, and the interactions between the tool and the environment during designing have to be considered during the creation of the tool. This dependency can be modeled explicitly by the notion of "situatedness" in designing through its emphasis on the interrelationship between knowledge (information that the tool has) and its application.

Situatedness in designing is concerned with the interactions between the designer and their environment so that the decisions that are taken are a function of both the situation and the way the situation is constructed and interpreted. This property of designing can be viewed through the notion of constructive memory. Constructive memory implies that memory is not a static imprint of an experience that is available for recall later through appropriate indexing (the computer filing-cabinet metaphor). Design experiences are stored, and the memories of them are constructed in response to a demand on these experiences so that it is possible to answer queries and effect actions that could not have been conceived of when that experience first occurred. The memory of an experience is a function of the situation in which the question, which provokes the construction of that memory, is asked.

A computational constructive memory system created according to the notion of situatedness is a dynamic, flexible, and open-ended learning system that has the ability to adjust and change itself to reflect its experience. Through the explicit modeling of the situatedness in designing via this memory system, the unpredictability of designing can be resolved to provide better support for designing.

Reprint requests to: Pak-San Liew, Key Centre of Design Computing and Cognition, School of Architecture, Design Science and Planning, University of Sydney, Room 290, Wilkinson Building (GO4), 148 City Road, Chippendale, NSW 2006, Australia. E-mail: paksan@arch.usyd.edu.au

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

Computer-aided design tools based on this memory system, modeled as situated design agents, are no longer seen as passive aids to designers. The usefulness of these tools increases through their applications (Gero, 1998*a*). Utility of the tool is seen as a means to change the tool so that its performance improves with usage through the acquiring or restructuring of knowledge. This knowledge is developed as and when required, and grounded according to the interactions between the tool and the environment. The quality of the knowledge is constantly improved through its use so that when the tool is reused in a similar situation it may have different knowledge than it had before. This new knowledge gives the tool an ability to produce better results, and its performance improves with usage and is not limited to what was initially encoded into the system.

2. THEORETICAL MODEL

The ideas presented in this paper are drawn from cognitive studies of the human behavior. In particular, the following models of human memory are adopted:

- modal model of the human memory system (Atkinson & Shiffrin, 1968),
- multiple-component model of long-term memory (LTM; Baddeley, 1999),
- multiple-component workspace model of working memory (Logie, 1995), and
- notion of constructive memory (Dewey, 1896; Rosenfield, 1988; Clancey, 1997).

Useful properties are extracted from these cognitive models to facilitate the creation of a computational memory system that exhibits some similar behaviors. The same terminology is used to describe the same memory phenomena in this paper. However, the aim of this paper is not to model the human memory system but to draw out useful properties from an analog of these models.

2.1. Model of constructive memory

Memory construction occurs whenever past experiences are utilized in the current situation. A memory is created in the construction process to give an agent the ability to operate in a new environment according to a similar previous experience. Through the construction process, all required information is created as and when demanded by the current situation. An agent that constructs memories of past experiences is actively restructuring what it knows through its interactions with the environment. These changes form the basis of adaptive learning that situates the agent within an environment.

Figure 1 illustrates a model of constructive memory that operates within the framework of a situated design agent (described later). Three memory construction processes are shown, and each carries out the following operations:

- cueing: the memory system is first cued by a demand from the current situation;
- activation and selection: a series of experiences is activated and one is selected;
- memory creation: memory is constructed according to the selected experience;
- incorporation: the constructed memory is incorporated into the system as a new experience available for subsequent use; and
- grounding: the activated experience is grounded if it is used as a basis for effecting actions into the current environment.

Memory construction commences with a demand for a memory. The memory system is cued according to the current situation. This situation is determined by the current state of the environment, the current state of the agent, and the interactions between the agent and the environment (not illustrated in Fig. 1). The current state of the environment is represented by what information is currently available from the environment. The state of the agent is determined by the knowledge and experiences processed by the memory system and what has been learned so far. Interactions between the agent and environment occur via sensations and actions (described later). Through cueing, associated experiences (Fig. 1) are activated for use during memory construction. One or more of these experiences is selected for each memory construction.

The first form of memory construction (Interpretation I) interprets the environment according to a past experience. When an experience has been selected, the knowledge associated with this experience is used to determine what gets extracted from the environment and the way in which the extracted information is structured. The knowledge and experiences form part of the situation and affect the kinds of memories that are constructed. The constructed memory forms an interpretation of the previous experience according to the current interactions between the agent and the environment. The experience provides the basis for interpreting the environment according to the current objective of the agent. Actions to be carried out are determined by this experience according to a similar situation in the past. The constructed memory is incorporated into the memory system as a new experience.

The experience that was selected and used during memory construction is grounded. In subsequent memory construction, this experience has more chance of being used due to its higher level of processing (described later) resulting from the current grounding. All experiences (existing and those created by incorporating constructed memories into the system) are available for use in the subsequent construction of memories and they are subjected to ground-

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

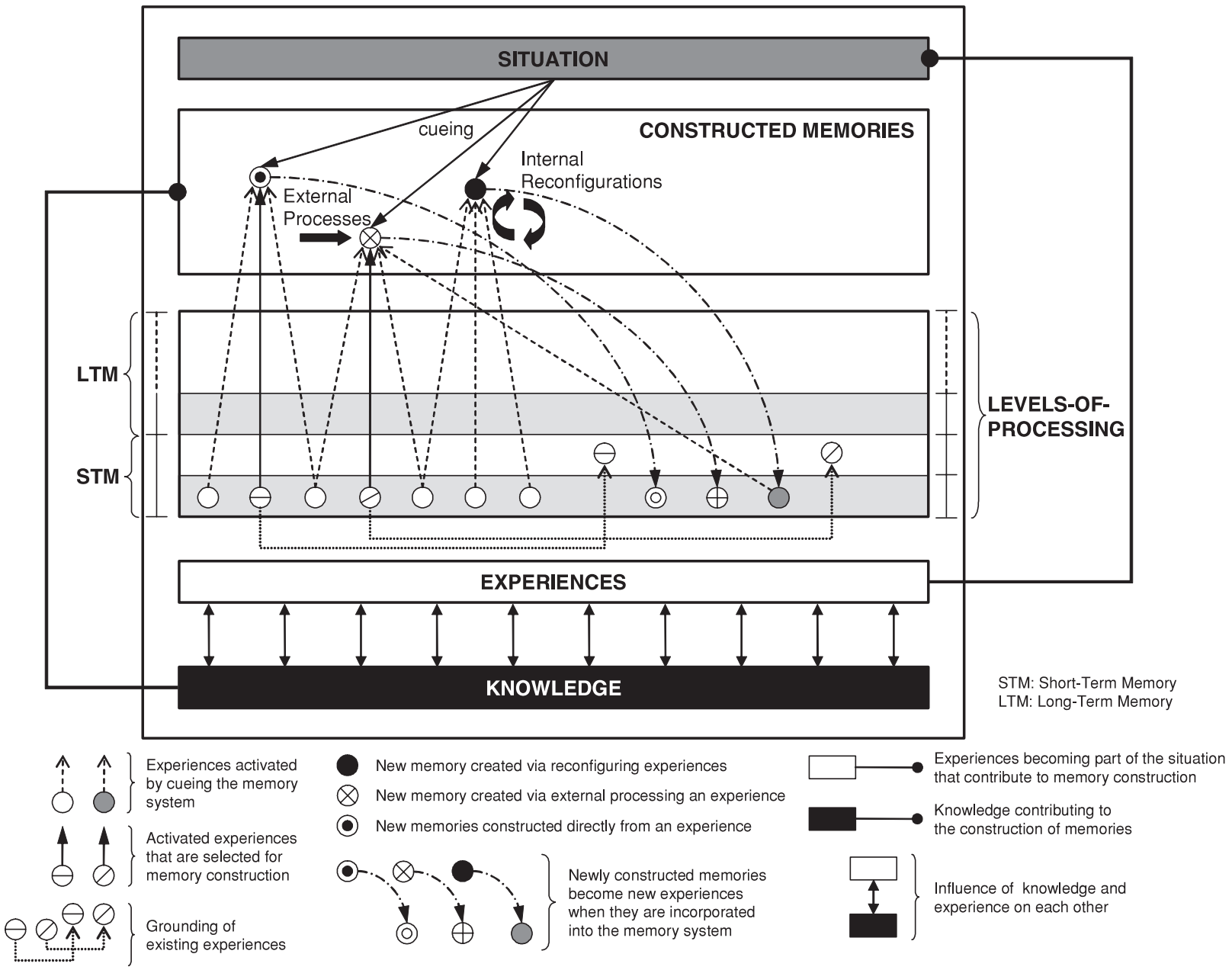


Fig. 1. The model of constructive memory (adapted from Gero, 1999*a*). The number of levels for LTM and STM is not limited to those shown.

ing when they form the basis for effecting actions into the environment.

The second form of memory construction (Reinterpretation II) creates memory through external processes such as analogy. An analogy is drawn between a related experience and the current situation during the construction of a memory. This memory is incorporated in the system as a new experience and the experience used as a source in the analogical process is grounded to give it greater emphasis in subsequent memory constructions.

The final form of memory construction (Reinterpretation I) is based on the modifications of the memory system's structure through internal configurations of its contents. In this case, a memory is constructed by modifying existing experiences with better performing structures from the environment. Once incorporated into the memory system, the constructed memory becomes a new experience that is available for subsequent use.

2.2. Learning

The experiences processed by an agent are not encoded *a priori*. The agent starts off with a basic set of capabilities and gains experiences as it carries out its tasks. Learning occurs when the agent modifies the way existing experiences are used and when the agent gains new experiences through its interactions with the environment. These two forms of learning are modeled by the notions of *long-term learning* and *constructive learning*, respectively.

2.2.1. Long-term learning

Long-term learning affects the way existing experiences are used during memory construction over a period of time. It deals with the adjustment of an experience's state that facilitates its future utility. This form of learning is effected through the *experiential grounding* of an experience that has been used in the current situation to effect actions onto the environment. An experience that has been grounded frequently is more likely to be used in subsequent memory constructions. The effects of grounding are modeled by the notion of levels of processing (Fig. 1).

The process of grounding entails the embedding of an artificial agent into its environment such that its behavior and the mechanisms, representations, etc., underlying this behavior are intrinsic and meaningful to the agent itself, rather than dependent on an external designer or observer (Ziemke, 1999). The meaningfulness of these mechanisms, representations, and so forth, is determined by their usefulness in assisting the agent in attaining any of its goals (Nehaniv, 1999). Although *symbolic groundings* deal with the provision of meanings for symbolic representations, *experiential grounding* deals with the provision of meaning for the experiences processed by the artificial agent. (Hereafter "grounding" refers to experiential grounding as described in this section.) For an experience to have meaning for an agent, the consequence of actions based on this experience must have an impact on the agent itself (Nehaniv & Dautenhahn, 1998). Underlying this meaningfulness of experience is the applicability of the experience through its usage as a basis for effecting actions. The interaction between the agent and the environment plays an important role in experiential-based grounding and has an effect on the use of existing experiences processed by the agent. An agent, with its current experiences, creates a particular view of the environment. If this view matches the other view created by agent through its interactions with the environment, it becomes useful in predicting the effects of a course of actions. The experience underlying this prediction is labeled useful and is grounded. This process of grounding models the effect of long-term learning.

Different *levels of processing* model different groundings of experiences based on their previous applications. A highly grounded experience has a high level of processing and is used more readily in the subsequent operation of the memory system. The different levels of processing are divided into two regions according to a predefined value of grounding to demarcate short-term memory (STM) and LTM (Fig. 1). An experience starts off in STM after its first creation by memory construction. This experience progresses toward LTM as it gains groundings through its applications in subsequent memory constructions. As its usage increases, the groundings of other experiences reduce, causing them to move toward STM. Experiences in STM are purged out of the system when required. These behaviors treat STM as a gateway for experience to get into LTM or out of the memory system.

Note that the notions of grounding and levels of processing determines which experience is given higher priority for use in the current memory construction. Experiences residing in both STM and LTM are available for use during memory construction. STM operating as a gateway is only applicable to the "movement" of experiences among the different levels of processing according to the groundings they received.

2.2.2. Constructive learning

The experiences processed by an agent affect the way in which it interacts with the environment (Lueg & Pfeifer, 1997). These experiences provide a basis for determining the actions to be effected in the current situation (Rylatt & Czarnecki, 1998). A commonly accepted notion about expertise is that experts differ from novices in terms of the number and quality of experiences they have in a particular domain. Useful experiences acquired over time form the basis for evaluating and refining complex knowledge organization (Goker, 1997; Seifert et al., 1997). This has the implication that the contents (knowledge and experiences) of a memory system are dynamic in nature and vary accordingly as the agent encounters new situations.

Constructive learning models these dynamics of a memory system based on the constructivism view of learning. This view of learning adopted here is derived from two fundamental principles of constructivism (Tait, 1997).

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

- Learning involves an active construction of a conceptual knowledge base of the learner (Bruner, 1966; Piaget, 1971; Jonassen, 1991).
- Learning is reflective in nature. It builds upon and develops the existing knowledge and experiences of the learner (Bruner, 1966; Ausubel, 1968).

Constructive learning uses similar experiences as bases to create new experiences during memory construction. Elements of a past experience are mapped into elements of the current situation to form a coherent memory structure for performing actions onto the environment or to the agent itself. The effect of constructive learning is a change in structure of the memory system through a reconfiguration of existing experiential components or the addition of new ones. Constructive learning occurs whenever a constructed memory is added into the memory system as a new experience.

2.3. Design experience

Design experiences contained within a constructive memory system play a critical role in designing, as they form the basis for the agent's behavior in the current situation related to similar situations it encountered in the past. (Hereafter "experience" refers to a design experience as described in this section.) The *design experience* used in this paper encapsulates ideas from the variables and processes operating within a situated design agent (Gero & Fujii, 2000), the situated function–behavior–structure framework of designing (Gero & Kannengiesser, 2000, 2002), and a design prototype (Gero, 1990).

2.3.1. Representation of design experience

A *design situation* models a particular state of interaction between a design agent and the environment at a particular point in time. It is a snapshot of all the variables that define the internal state of the agent and the external state of that part of the environment that the agent is interacting with. The expectation and interpretation of the agent about the environment may or may not be in agreement with each other within a design situation. This agreement is based on the comparison of the behaviors contained within the expectation and interpretation of the environment. A disagreement implies that there is a difference in terms of either behavioral attributes or attribute values and this forms the basis for actions.

A *design experience* models the transition of state from one design situation to another situation in which the expectation is anticipated to agree with the interpretation. The transition is represented by the initial and final state together with the action that facilitates the transition. Essentially, a design experience consists of two design situations and the action that transforms the initial situation into the final situation. Within the initial situation, the expectation and interpretation are not in agreement. In the final situation, the expectation agrees with the interpretation. Symbolically, a design experience (E) is represented by a collection of constituent parts:

$$E = \{V^{\mathrm{i}}, S^{\mathrm{i}}, M, Q^{\mathrm{i}}, H^{\mathrm{i}}, A, H^{\mathrm{f}}, Q^{\mathrm{f}}, V^{\mathrm{f}}, S^{\mathrm{f}}\},$$

where A is the action on the agent or on the environment; H^{i} is the initial focused concept; H^{f} is the final focused concept; M is the percept; Q^{i} is the initial expectation; Q^{f} is the final expectation; S^{i} is the initial sensory data; S^{f} is the final sensory data; V^{i} is the initial exogenous variable; V^{f} is the final exogenous variable; $Q^x = \{F, B, S\}$, with Q^x denoting the initial expectation if $x = i$, or the final expectation when $x = f$, F, B, and S denoting the function, behavior, and structure of the expectation, respectively; H^x denotes the initial focused concept if $x = i$ or the final focused concept, if $x = f$; S^x denotes the initial sensory data if $x = i$ or the final sensory data, if $x = f$; and V^x denotes the initial exogenous variable if $x = i$ or the final exogenous variable, if $x = f$.

Different aspects of designing are captured by the representation of a design experience. In this research, design information relevant to the situated function–behavior–structure view of designing are modeled. Other aspects of designing (such as design rationale) can be included in the representation of a design experience if they are to be processed explicitly by the memory system containing the experience. The use of a computational constructive memory system is limited by the representation of design experience adopted. If the required aspects of a design domain can be represented within a design experience, the proposed memory system can be used in that domain.

Focused concept. An H^x (Gero & Fujii, 2000) dictates the focus of the design agent based on a design goal. This focus determines the required functionality, and consequently, the required behaviors of the external structure according to the design goal. The function is used to bias sensation and perception (described later) while the behavior is used when the interpretation is evaluated against the expectation of the environment. A design prototype (Gero, 1990) is used to represent a focused concept. This prototype represents the design knowledge used within a design experience. A design prototype can either be retrieved from the memory system or derived from existing prototypes. External processes such as analogy, combination, mutation, and designing from first principles (Rosenman & Gero, 1993) and reformulation processes (Gero, 1998*b*) that deal with reparameterization and parameter expansions can be employed for this purpose. These operations are instantiations of the general conception process (described later). Each design prototype contains procedural knowledge that defines how the prototype can be used within a design experience to achieve a specific goal. This procedure is used to derive the required action to be effected into the environment.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

Action. An A is either a sequence of operations an agent performs internally on itself, or a sequence of operations performed externally on the environment (i.e., on the description of the design artifact). An internal action changes the behavior of the agent by altering variables representing its internal states. Long-term learning and constructive learning are instantiations of internal actions that alter the internal states of a design agent. An external action transforms the V^{i} to V^{f} according to the current design goal.

Exogenous variable (external structure). The variable V^x defines that part of the environment an agent is dealing with according to its design goal. It represents a reference to the *external structure* of a design artifact (Gero & Kannengiesser, 2000, 2002) to be manipulated by the agent. A design agent may or may not operate on the entire external structure. What is focused upon is a function of the current focused concept. The set of exogenous variables encountered by the design agent is different for unfamiliar environments.

Sensory data. The variable S^x models the raw data extracted from V^x based on the bias set by perception. The variables V^{i}, S^{i}, V^{f}, and S^{f} model the state of the artifact and its associated data sensed by the agent, before and after the agent has performed the required action according to its design goal.

Concept as expectation of the environment. The design prototype representing the current focused concept contains information pertaining to the function, behavior, and structure of a design artifact in terms of a dependency network (Gero et al., 1992). This part of the prototype is extracted as a concept through conception to form Q^x (Gero & Kannengiesser, 2000, 2002) of the agent about the environment according to the current design goal.

Percept (interpreted structure). The function, behavior, and structure information from the focused concept is used in perception to bias the sensation process and structure, S^x. The function information dictates which part of the environment will be focused upon and sensed by the agent. Only parts with the same function as that of the focused concept are extracted from the environment. The variable S^x is configured into a complete structure to form *percepts* (M), a form of *interpreted structure* (Gero & Kannengiesser, 2002) during perception.

Concept as interpretation of the environment. The *interpretation* of the environment (Gero & Kannengiesser, 2000, 2002) is created from the percept. This interpretation is represented by a dependency network that contains function, behavior, and structure information about the interpreted structure. The behavior and structure part of the interpretation are either computed from or extracted directly from an external repository of design prototypes. An interpretation is not stored within the representation of a design experience. It is computed as and when required.

2.3.2. Processes connecting constituents of a design experience

The constituents of a design experiences are connected by a series of interrelated processes. Figure 2 illustrates an example of a design experience (E_1) based on the domain of Design for Assembly (DFA).

Domain descriptions. DFA is an engineering design methodology that deals with the design of products for easy assembly to lower their production costs. Boothroyd and Dewhurst (1989) presented a series of Product DFA techniques that are concerned with the reduction of the cost of a product through the simplification of its design. It is proposed that the best way to achieve this cost reduction is first to reduce the number of individual parts that must be assembled and then to ensure that the remaining parts are easy to manufacture and assemble.

Figure 3 illustrates a series of mechanical assemblies used to illustrate the application of a DFA principle: reduce the number of parts (Boothroyd et al., 1982, 1994; Boothroyd & Dewhurst, 1989; Boothroyd, 1994). The main functionality of these assemblies is to fasten two plates together. From the perspective of the function, behavior, and structure of these assemblies, the common function of "fastening" is achieved through a series of behaviors, one of which (relevant to DFA) is "the number of parts." The components (bolt and nut, snap-fit, and integrated snap-fit) represent different structures that exhibit the required behavior.

In terms of DFA, the integrated snap-fit assembly gives the best performance because it requires the least number of parts. This assumes that the integrated snap-fit can be produced at a cost lower than that of a snap-fit. If a snap-fit is cheaper to produce, snap-fit assemblies are preferred. During the redesign of an external structure, all bolt and nut assemblies within the structure are replaced by snap-fit assemblies or integrated snap-fit assemblies to reduce production cost. The design experience that models the replacement of bolt and nut assemblies by snap-fit assemblies is illustrated in Figure 2.

Sensation. Sensation is an information extraction process that converts the external structure (V_1^{i}) into sensory data (S_9^{i}) according to the focused concept (H_1^{i}). Function, behavior, and structure information from the focused concept [FBS(1) refers to labels in Fig. 2] is used to bias perception. Perception based on the function information it now processes, biases the sensor to extract only structures that have the same required function as that of the focused concept (F). A repository of design prototypes (Szykman et al., 1998) provides the necessary function, behavior, and structure information [FBS(3)] for the external structure.

Perception. Perception structures the sensory data S_9^{i} according to applicable function, behavior, and structure information of the focused concept [FBS(1)] to form an interpreted structure (percept) M_{10} of the environment.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

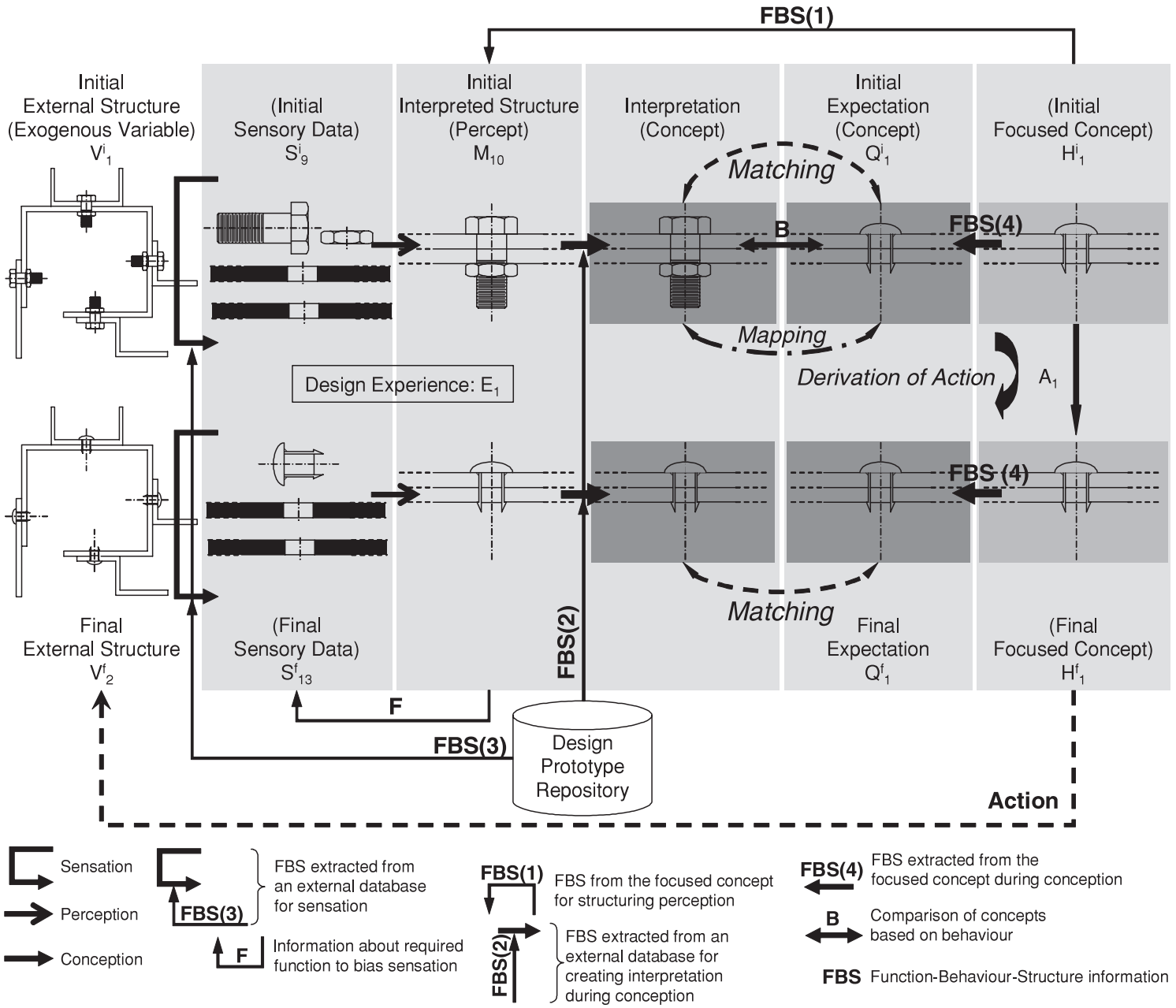


Fig. 2. Processes connecting different constituents of a design experience.

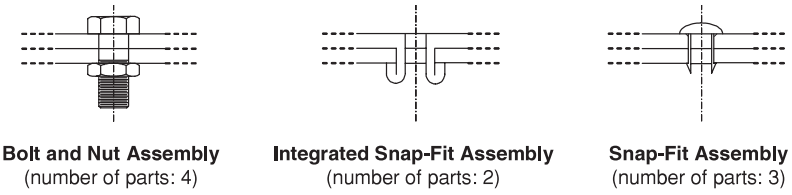


Fig. 3. A sample of mechanical assemblies.

Conception. An interpretation of the environment is formed by conception. This process uses function information from the focused concept, behavior information extracted and modified (if necessary) from an external repository of design prototypes [FBS(2)], and structure information from the environment to create a dependency network that represents the interpretation. The behavior and structure information of the interpretation is modified to a form suitable for processing during this conception process. This form is dictated by the function, behavior, and structure information from the focused concept H_1^i [FBS(1)]. The expectation of the environment (Q_1^i) is created from the focused concept (H_1^i) by extracting the dependency network from the design prototype representation [FBS(4)].

Matching, mapping, and comparison of behavior. The *mapping* process is a simplified version of the *analogy elaboration process* described by Qian (1994). Mapping is basically a transformation process used to identify structural correspondences between the expectation and interpretation. The basic aim is to find structure mapping (correspondence) through function, behavior, and structure matching (indexing). In general, matching implies that two (function, behavior, or structure) variables are identical at an abstract level. Mapping implies that two variables have the same associated functions or behaviors. Mapping is possible only after matching has been carried out successfully. When two functions match, the associated mapping for behavior and structure can be found. If two behaviors match, only structure mapping can be found. The interpretation and expectation are matched and subsequently mapped onto each other to obtain the correspondence between their structures before the required behavior is compared according to a design goal.

Derivation of action. When matching and mapping have been completed successfully to produce a structural correspondence between the dependency network of the expectation and interpretation, the procedural knowledge stored within the focused concept (H_1^i) is used as a basis for deriving the action (A_1) on the environment.

Completion of design experience. The action A_1 is effected onto the environment to create a new structure (V_2^f). The sensation, perception, conception, and matching process is repeated to form the final situation.

Initial and final situation. In the final form, the experience is made up of an initial design situation, an A_1, and a final design situation. The bolt and nut assemblies within the initial external structure (V_1^i) are replaced by snap-fit assemblies in the final external structure (V_2^f).

2.4. Complete process to construct a memory

Processes occurring within the model of constructive memory either provide the initial information required for memory construction (cueing, activation, selection, and previous groundings) or dictate the treatment of a constructed memory (incorporation and grounding). The processes that link different components of a design experience provide the base for memory construction after the initial information is made available (by cueing, activation, selection, and previous groundings). A complete process to construct a memory thus entails the following:

1. cueing;
2. activation;
3. selection;
4. memory creation;
   a. sensation;
   b. perception;
   c. conception;
   d. matching, mapping, and comparison of behavior;
   e. derivation of action;
   f. performance of action;
   g. completion of experience:
      i. sensation;
      ii. perception;
      iii. conception;
      iv. matching of final interpretation and expectation;
   h. incorporation;
   i. grounding.

Long-term learning is effected through the grounding of an experience. It is a process separate from the creation of a memory during memory construction. When a constructed memory has been created and incorporated into the memory system, constructive learning occurs. Both long-term

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

learning and constructive learning are consequences of memory construction.

3. COMPUTATIONAL IMPLEMENTATION

The theoretical model described in Section 2 provides the required specifications for the design and implementation of a computational constructive memory system. (Hereafter "memory system" refers to a computational constructive memory system as described in this section.) The requirement for interactions with the environment through sensation, perception, conception, and action leads to the agent construct (Russell & Norvig, 1995) that bases its operations on situatedness in designing (Gero & Fujii, 2000). A framework of a situated design agent presented provides the necessary structure for encapsulating a constructive memory system that is implemented through an extended Interactive Activation and Competition (IAC) neural network (McClelland, 1995).

3.1. Framework of a situated design agent

The memory system developed in this paper operates within a cognitive-based situated design agent. (Hereafter "agent" refers to a situated design agent as described in this section.) A situated design agent is essentially an autonomous computer construct modeled around the ideas of agency and situatedness in designing. This agent provides the necessary infrastructure for memory construction. Figure 4 illustrates a high-level view of the functional components that make up the situated design agent. The functionalities of these components have been described in terms of the operations of the state-of-affair and course-of-event memory systems in Gero and Fujii (2000). Essentially, the sensor, perceptor, and conceptor provide functionalities for sensation, perception, and conception. An effector provides the means to perform actions on the agent or on the environment.

3.2. Architecture of a general computational memory system

Figure 4 also illustrates the architecture of a general computational memory system for a situated design agent. The paths in Figure 4 are shared between processes to avoid overcrowding the diagram. They can be taken as directed associations between two entities, that is, information or instructions flow in the indicated direction. The constructive memory system proposed in this paper is embedded within the general computational memory system. Only paths and components that are relevant to this constructive memory system for memory constructions are described.

Memory construction occurs within the working memory according to experiences that are activated and selected from either STM or LTM. The working memory is the working environment where executive control functions are performed. These processes selectively alter and/or add information to the working memory based on the contents of LTM (14) or STM (12). (The numbers in parentheses refer to the paths in Fig. 4.) Information within the working memory is combined with the stored knowledge and experiences, manipulated, interpreted, and recombined to develop new knowledge, assist learning, form goals, and support interaction with the environment. The reasoning that occurs within the working memory is based on three main components: the executive control, content from STM or LTM, and the cache (11). The executive control is where the agent control processes are located. These processes are involved in the organization, coordination, manipulation, movement, and comparison of information and knowledge for design-related reasoning. Experiences and their associated knowledge are stored in STM or LTM. The cache provides the buffering for the reasoning processes. Inputs to working memory arrive from the conceptor (8) and perceptor (9) during conception, the perceptor (9) during perception, and from LTM (14) or STM (12) when they play host to activated experience for memory construction.

During long-term learning, an experience is transferred to/from LTM from/to STM without any structural change being made to its constituents. STM serves as a gateway into LTM and out of the memory system. The path into LTM (17) is used when an experience is transferred from STM to LTM as it gains groundings through increased applications. STM acts as a gateway out of the memory system when experiences in LTM that are not grounded frequently are transferred into STM (16) and possibly later, out of the memory system. These two actions model the phenomenon of "use it or lose it."

STM and LTM are modeled as two separate constructs to illustrate the different contents of the general computational memory system and the treatment of different experiences based on their levels of processing. The multiple-component illustration of LTM emphasizes the fact that memory construction involves experiences stored within the episodic memory and their associated knowledge stored within semantic memory. Each experience has its associated knowledge reference by a link to the semantic memory (20). The slots in STM emphasize the notion of levels of processing. This concept is significant in STM as the purging of experiences from the system occurs in STM, and this removal of experience has a great impact on subsequent operations of the memory system. The same levels of processing concept is also applicable to LTM. The differences between experiences residing in STM and LTM lie only in the fact that experiences in STM are not as readily recalled in the memory construction process as those in LTM and they are subjected to removal from the memory system.

3.3. Implementation of constructive memory

Memory construction can be viewed as a constraint satisfaction process in which a memory of a past experience is

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

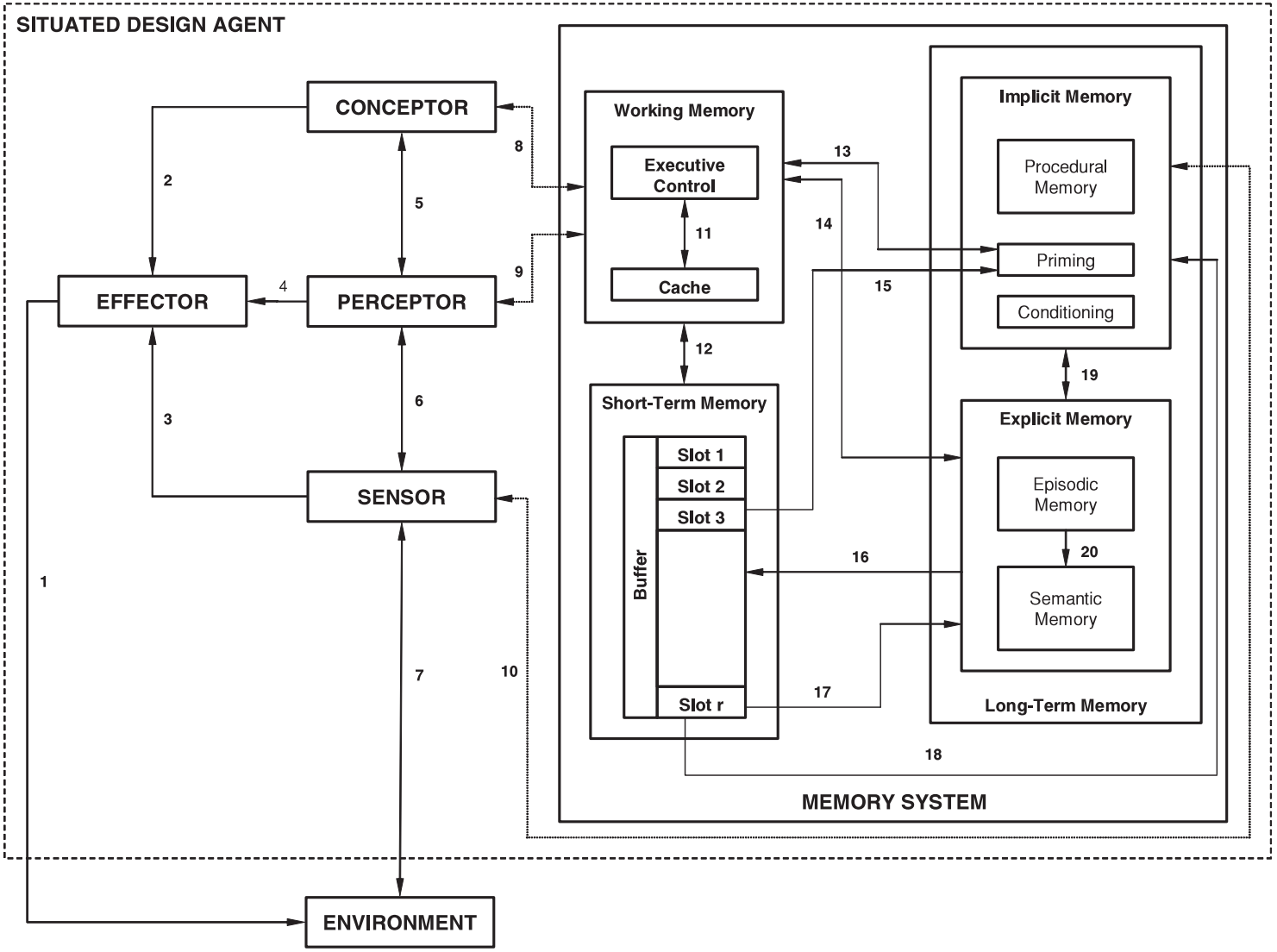


Fig. 4. The conceptual architecture of a general computational memory system within a situated design agent; numbered paths are indicated as (#) in the text, where (#) is the number as shown in the diagram.

created under the constraints of the contents of the experience in question, the available knowledge and the influences of the current situation (McClelland, 1995). This constructive process has been implemented through an IAC network (McClelland, 1981) by McClelland (1995), and forms the foundation for the implementation of the constructive memory proposed in this paper.

An IAC network is a neural network that uses an activation and competition mechanism to operate on a collection of nodes. The knowledge of an individual experience is captured in a network of nodes. An *instance node* and a set of corresponding *property nodes* are defined for each of the experiences the system knows. The system's knowledge of a design experience is represented as an instance node linked to a set of property nodes through excitatory connections. The connected property nodes represent the constituents of the design experience (Fig. 5).

The property nodes and instance nodes are arranged in groups of mutually exclusive inhibitory connections. The property nodes are localized representations of concepts. These representations are activated via a spreading activation mechanism where activated experiences (instance nodes) activate the representation of their properties. Mutually exclusive property values compete in such as way that allows

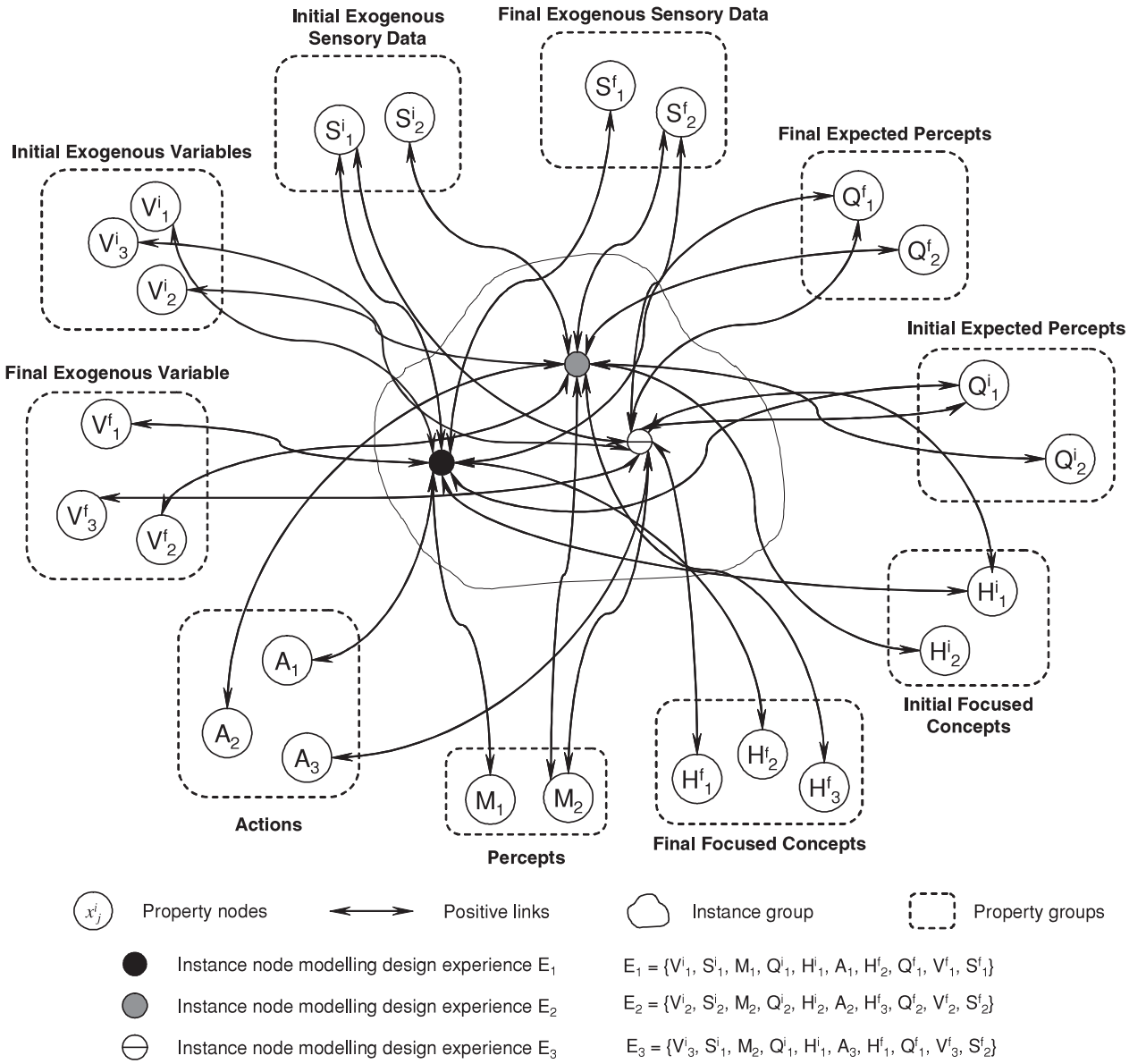


Fig. 5. Representations of design experiences within an IAC network; negative links connecting property nodes within the same property group are not shown.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

properties that are supported by a large subset of the active instance of the category to be reinforced and become strongly active while suppressing those that are not.

3.3.1. IAC formulation

The standard formulation of the IAC network (McClelland, 1988) is used as a base upon which learning mechanisms are incorporated. In an IAC network the total input for node $j(t_j)$ is

$$t_j = \alpha e_i + \sum w_{ji} a_i,$$

where α represents the external input scaling, e_i represents the external input, w_{ji} represents the weight to node j from node i, and a_i represents the activation of node i. During cueing, if the probe contains the node j, e_j is set to 1.0. α is set to a global value of 4.0 for the entire network.

After the total input has been calculated for all the nodes, activations are updated according to the following equations (O'Reilly & Munakata, 2000):

$$(t_j > 0): \quad \Delta a_j = s\{(a_{\max} - a_j)t_j - d(a_j - r)\},$$

$$(t_j < 0): \quad \Delta a_j = s\{(a_j - a_{\min})t_j - d(a_j - r)\},$$

where Δa_j denotes the amount by which to change a_j.

This formulation differs from the original IAC with the additional parameter step (s) for controlling the rate at which the node activations are changed: s is set to a global value of 0.1 for the entire network; $a_{\max}$ is set to 1.0, $a_{\min}$ to -0.2, rest (r) to -0.1, and decay (d) to 1.0. The network starts off with all nodes having an initial resting activation value of -0.1. Positive links are set to 1.0 and negative weights are set to -1.0. These settings are used for the initial network before any learning occurs. The same parameter values are also used for all newly created experiences.

During memory construction, a probe is created within the IAC network as a cue for memory construction. The contents of this probe consist of a new property node that models the external structure from the environment and an existing property node that is reused for modeling the current goal of the agent for memory construction. Once the probe has been created, its property nodes are clamped (given an external input) and the network is cycled for a predefined number of times. For each cycle, the activation value for each node is updated according to the above equations. When all the activation values have settled into an equilibrium state, the resulting instance nodes and their associated property nodes that are activated with positive values, highlight similar design experiences that can be used in the current situation. Each activated experience has an associated activation value based on its usefulness in previous design sessions. A reference to the activation value of an activated experience refers to the positive activation value of the instance node representing that experience within the extended IAC network.

3.3.2. Learning mechanism

The original IAC network (McClelland, 1981, 1988; Rumelhart & McClelland, 1986) does not provide any learning mechanism. Learning can occur through:

- constructive learning via changing the structure of the network; or
- long-term learning via grounding through the adjustments of weights between existing nodes of a design experience.

Constructive learning via system restructuring. Computer implementations of constructive learning are exemplified in the modification of neural network structures (Mandic & Chambers, 2001). Constructive learning in neural networks deals with the changing of architecture or interconnections within the network. There are two basic classes of constructive learning: network growing and network pruning. In the network growing approach (Hoehfeld & Fahlman, 1992), the network starts small and grows as it learns. Network pruning starts with a large network and reduces its size until the desired ratio between accuracy and network size is reached (Reed, 1993). The idea behind these learning mechanisms is similar to the concepts of ontogenic networks (Fiesler & Beale, 1997) where the topology of the network is changed during learning.

Constructive learning within the IAC via network growth can occur with the addition of new instance nodes that model new experiences (constructed memory) created via memory construction (Fig. 6). New property nodes are created and added into the system according to new external structures from different environments (Interpretation I, Reinterpretation I, and Reinterpretation II) and according to the reconfiguration of existing property nodes (Reinterpretation I). Constructive learning via network pruning occurs through the forgetting process based on the levels of processing of different experiences. In this processes, instance nodes and property nodes that are not shared among other instance nodes are removed from the IAC network.

Long-term learning via weight modification. Computer implementations of long-term learning are exemplified in the modification of weights in neural networks (Haykin, 1998; Medler, 1998). Long-term learning in the IAC networks deals with the changing of interconnection weights within the network of nodes and links through a grounding process. Grounding causes the positive links of a grounded experience to increase in value while reducing the positive link values of all other experiences through a decay process. Only the weights of excitatory (positive) links are allowed to change. Negative links are not changed locally as their global settings are used to control blending.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

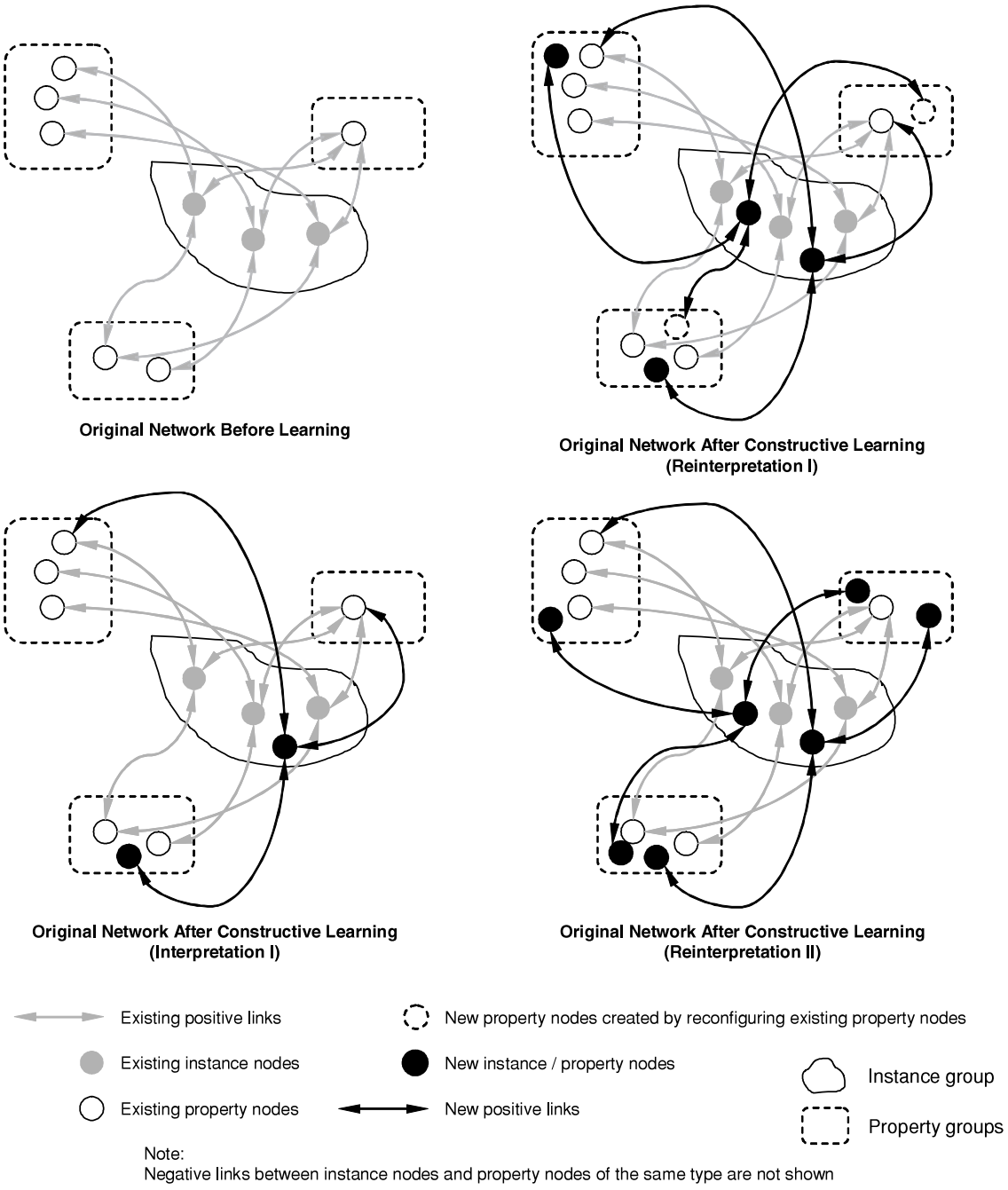


Fig. 6. The IAC network before and after various forms of constructive learning.

The learning mechanism is formulated as (Medler, 1998)

$$\Delta w_{ij} = \beta\{(w_{\max} - w_{ij})a_i a_j - \delta w_{ij}\},$$

where Δw_{ij} denotes the amount by which to change the link between node i and node j, β denotes the learning rate, δ denotes the weight decay factor, and $w_{\max}$ denotes the maximum weight value. This formulation of learning is similar to that for computing activation values of nodes. The learning process occurs as a separate process only after memory construction has taken place.

The learning processes based on weight adjustments in the IAC network is carried out by clamping the property nodes of the experience to be grounded and cycling the network according the formulation for learning above. This emphasis of a particular experience strengthens the interconnections between the nodes involved. Whereas it is possible to overtrain other neural networks, it is not possible to overtrain the extended IAC network via grounding of expe-

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

riences based on their applications. The activated experiences are always evaluated according to the current situation through sensation, perception, conception, matching and mapping, and comparison of behavior.

3.3.3. Implementation of STM and LTM

STM and LTM are implemented as a multilayered IAC network (Fig. 7). The separation of experiences into different layers is based on their levels of processing. The common interconnection weight between the instance node and its corresponding property nodes of an experience provides the basis for these levels. Experiences with high interconnection weights are placed close to the left of Figure 7 to indicate that they are more durable within the memory system. Experiences with low interconnection weights are placed toward the right of Figure 7. A threshold value for interconnection weight is set up to delineate LTM and STM. This threshold can be visualized as a cutting plane that separates LTM from STM among the different layers of the network.

New experiences start off in STM (right side, Fig. 7). As the grounded usage of an experience increases, it is repeatedly presented to the memory system to strengthen its interconnection weights and moves toward LTM (left side, Fig. 7). All experiences in STM and LTM are subject to a decay process. If the subsequent memory construction process does not reinforce an experience, its interconnection weights will decay and its will be moved into the STM region. When experiences are removed from the system by a forgetting process to increase computational performance, the experiences within STM are purged from the system first. The separation of experiences into STM and LTM is only critical during this removal process.

3.4. Role of the extended IAC network in memory construction

The extended IAC network is involved in the following memory construction subprocesses:

- cueing: where external inputs are supplied to the relevant property nodes of the network;
- activation: where the network is cycled according to the external inputs to activate related nodes with positive values;
- selection: where an activated instance node is chosen;
- creation of memory: where the focused concept (property node) of the selected experience (instance node) is used as a basis for sensation, perception, conception facilitated by the agent's infrastructure;
- matching and mapping: where the selected focused concept provides the required functional, behavioral, and structural information for linking up different structures within the interpretation and expectation of the environment for behavioral comparison; and
- derivation and performance of action: where the selected focused concept provides the required procedural information.

The extended IAC network plays the role of a repository of design experiences. After a memory has been constructed, it is incorporated into the network as a new experience through constructive learning. The instance and property nodes modeling this new experience are merged into the existing structure of the network.

4. ILLUSTRATIVE EXPERIMENTS

Situatedness in designing provides a different perspective on the role of machine learning in designing. Learning is seen as a means to change a tool so that its performance improves with usage through the acquisition and refinement of knowledge. The quality of the knowledge is constantly improved through its use, and as a consequence, when the tool is reused in a similar situation later, it will have different knowledge than it had before and thus potentially produces better results (Gero, 1998*a*). Two forms of learning that fulfill the above requirements are long-term learning through grounding and constructive learning. Constructive learning deals with the acquisition of knowledge. Long-term learning via grounding improves existing knowledge through its application so that the performance of the system containing this knowledge improves with use.

Long-term learning through grounding is the focus of the following experiments. These experiments deal with the redesign of mechanical assemblies for ease of assembly. Redesign is taken as the modification of an existing design according to the current goal of a design agent. The current goal is modeled through a series of design guidelines, principles or rules that fulfil the various objectives of the goal. An agent performing redesign for the purpose of DFA will modify any designs it encounters to allow their assembly operations to be carried out more easily and economically (the goal of DFA) according to a list of DFA guidelines. The focused concept of a design experience (represented by a design prototype) acts as an index to the design guidelines. During redesigning, a list of design prototypes is processed sequentially to emulate the serial processing of design guidelines. For each DFA guideline to be processed, the associated focused concepts and the exogenous variable representing the external structure are used in the cue for memory construction.

Two sets of experiments are conducted to demonstrate the different *operating modes* and *responses* of separate memory systems[1] (residing in different situated design agents) due to grounding. Different operating modes are

[1]The term "separate memory systems" refers to separate constructive memory systems that start off with the same set of experiences (Fig. 8) in the same states. These systems are the same in all aspects initially.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

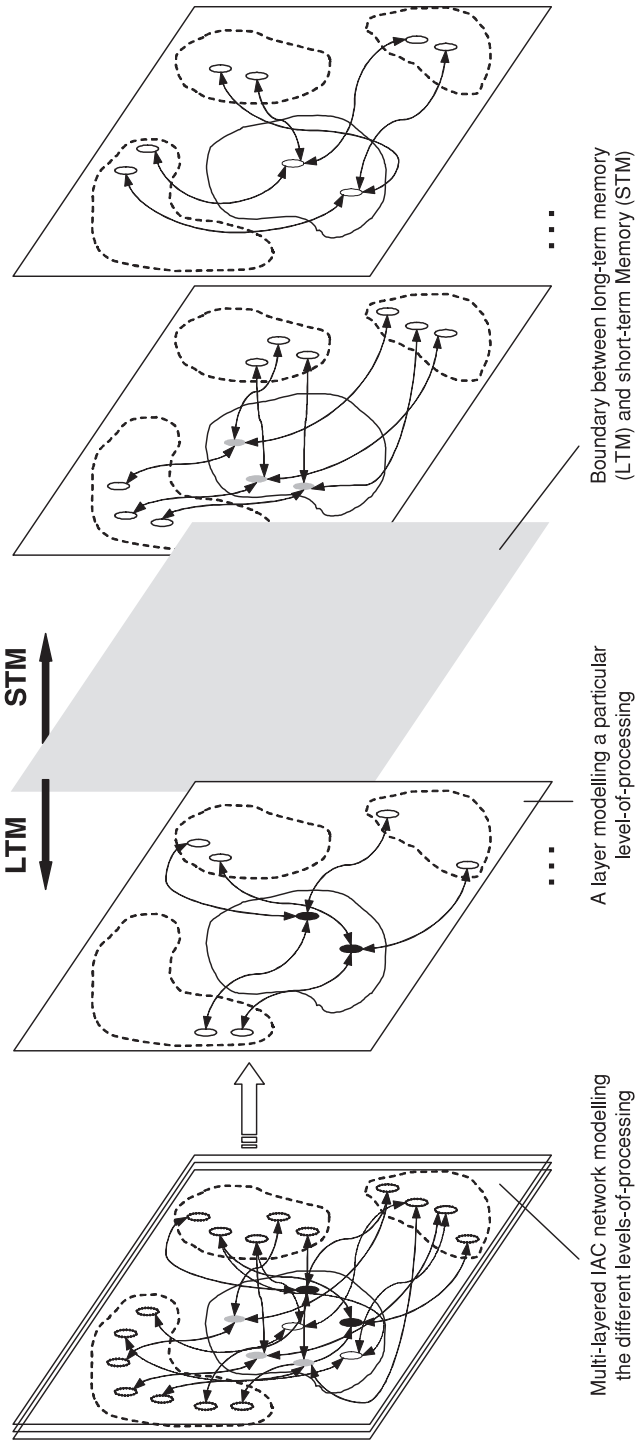


Fig. 7. The demarcation of STM and LTM. The number of levels for STM and LTM is not limited to those shown.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

indicated by different activation values of the same experience within separate memory systems. This difference in activation values highlights the fact that these systems operate in different ways to arrive at their respective response. The response of a memory system is obtained by extracting the activated experience with the maximum activation value within one memory construction. Two separate memory systems produce the same response when the same experience is activated with the maximum values within these systems. If the magnitudes of these activation values are different, it implies that these values are computed through different operating modes of the respective memory systems.

The basic aim of these experiments is to illustrate the situated behaviors of design agents based on constructive memory. If two agents (based on two separate memory systems), starting off with the same set of experiences (Fig. 8), are exposed to different situations, when they encounter the same situation later, their respective memory systems should operate or respond differently to demonstrate the effect of situatedness. These differences within the memory systems form the base for the different behaviors of the two agents.

The operating modes and responses of two memory systems are illustrated through the two runs associated with each experiment. Each run can be taken as a record of an agent's memory constructions history. To compare the operating modes and responses of these two memory systems, each system starts off with the same set of experiences in the same states and ends with the same situation. The intermediate groundings within each memory system are varied to illustrate their effects on the different operating modes or responses of the respective memory system in the same situation.

It should be noted that these experiments are used purely as a vehicle to illustrate the concepts and operations of the constructive memory system based on grounding. They are not here to justify the use of the memory system within their domains. The structures illustrated in these experiments can be redesigned in many other ways, some of which may employ mechanisms that are simpler than those proposed here. The memory system offers possibilities that are not directly derivable from these methods.

4.1. A simplified memory construction process

The following simplified memory construction steps are used in each experiment:

- creation of partial design experience as a cue to the memory system;
- cueing of memory system;
- cycling of memory system to obtain activations of related experiences;
- selection of a candidate experience;
- construction of new experience by completing the partial design experience according to the selected experience; and
- grounding of the relevant experience.

A partial design experience is created according to the current situation. This partial experience is made up of the current focused concepts (representing the design guideline to be used in the current redesign session) and the representation of the external structure. These parameters are embedded into the extended IAC network and clamped before the system is cycled to activate the relevant experiences. One activated experience is selected for the memory construction by completing the partial design experience used initially as a cue. The focused concept of the selected experience is used as a basis for sensation, perception, conception, matching and mapping, comparison of behaviors (based on DFA), derivation, and performance of actions.[2]

For all experiments, the following apply:

- experience E_1 has focused concepts H_1^{i} and H_1^{f}, each modeled by a design prototype of a snap-fit assembly;
- experience E_{801} has focused concepts H_{800}^{i} and H_{800}^{f}, each modeled by a design prototype of an integrated snap-fit assembly;
- experience E_{901} (used as a dummy variable in this paper to represent a totally unrelated experience) has focused concepts H_{901}^{i} and H_{901}^{f}, each modeled by a design prototype of a rocker-arm assembly; and
- relevant experiences are selected for grounding according to the specific aim of each experiment.

A particular memory construction is labeled P_x, where x represents a situation based on a particular set of initial focused concept, final focused concept, and exogenous variable. A common value of x for different memory constructions implies that these processes operate within the same situation. Grounding of an experience is performed separately from memory construction and is labeled G_z, where z represents the experience that was grounded. For example a record of memory constructions represented by $P_1 \rightarrow G_{E_{200}} \rightarrow P_2 \rightarrow G_{E_{200}} \rightarrow P_1$ denotes the following sequence of processes:

- P_1: cycle the memory system according to the cue: $\{V_{200}^{\mathrm{i}}, H_{200}^{\mathrm{i}}$, and $H_{200}^{\mathrm{f}}\}$, for example, and complete the cue (partial design experience) using the focused concept from an activated experience selected for the current memory construction;
- $G_{E_{200}}$: ground experience E_{200};
- P_2: cycle the memory system according to another cue, say, $\{V_{300}^{\mathrm{i}}, H_{300}^{\mathrm{i}}$, and $H_{300}^{\mathrm{f}}\}$ (representing a new situation where H_{300}^{i} and H_{300}^{f} are new focused concepts and V_{300}^{i} is a new exogenous variable) and complete the cue using

[2]Sensation, perception, derivation, and performance of actions are not demonstrated here, because they are not within the scope of this research. The results of these processes are created manually and embedded into the system.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

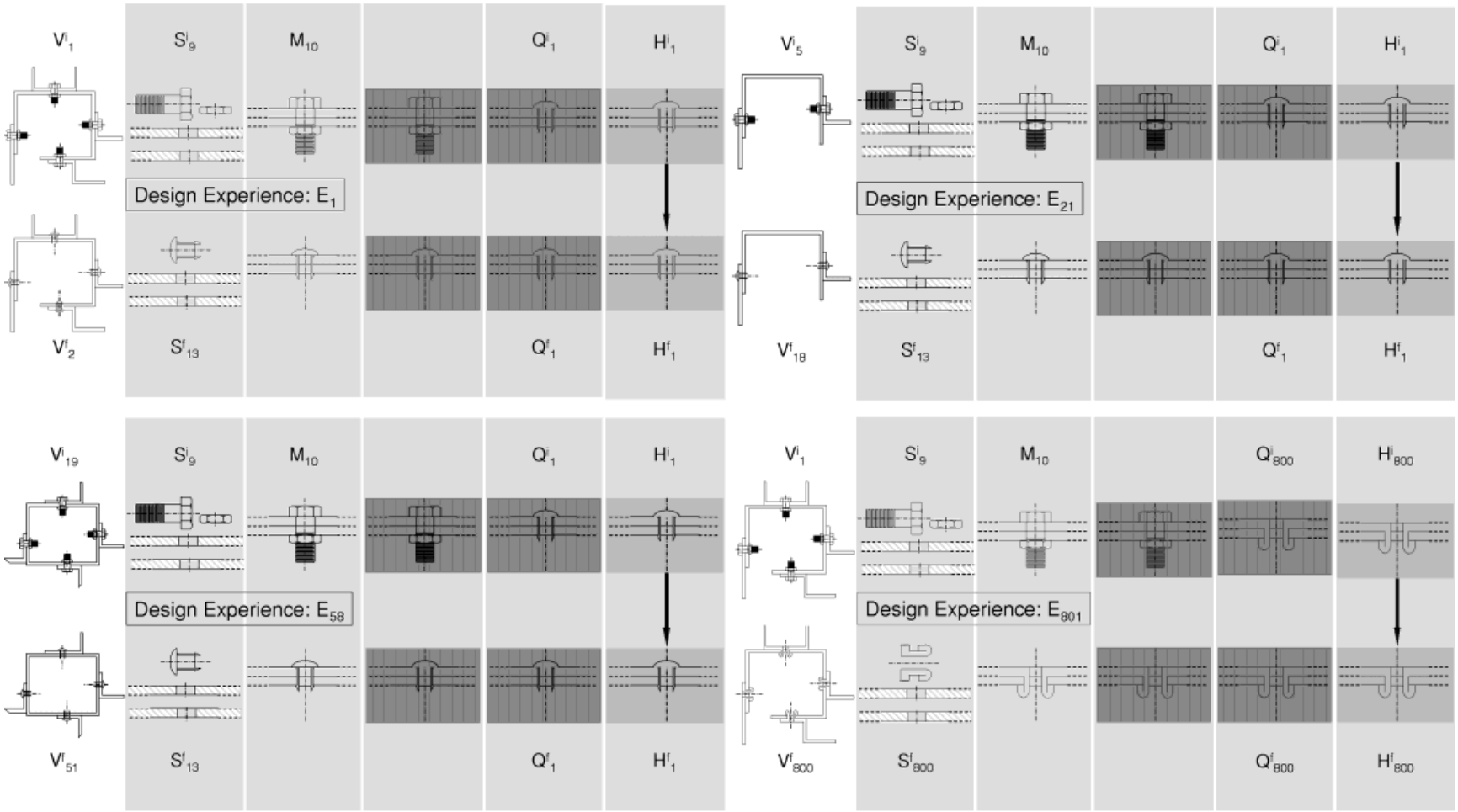


Fig. 8. (*continued on next page*)

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

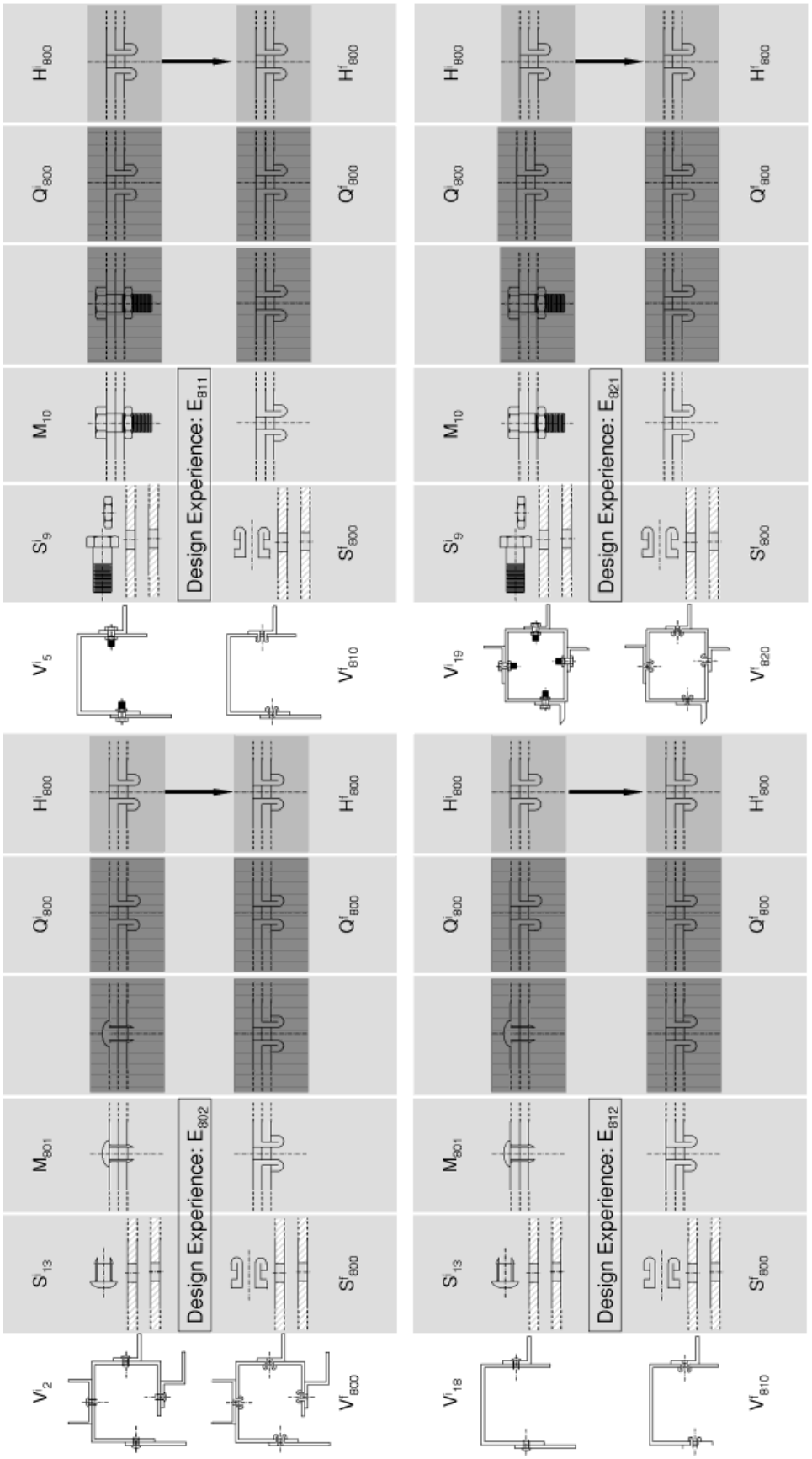


Fig. 8. (*continued*)

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

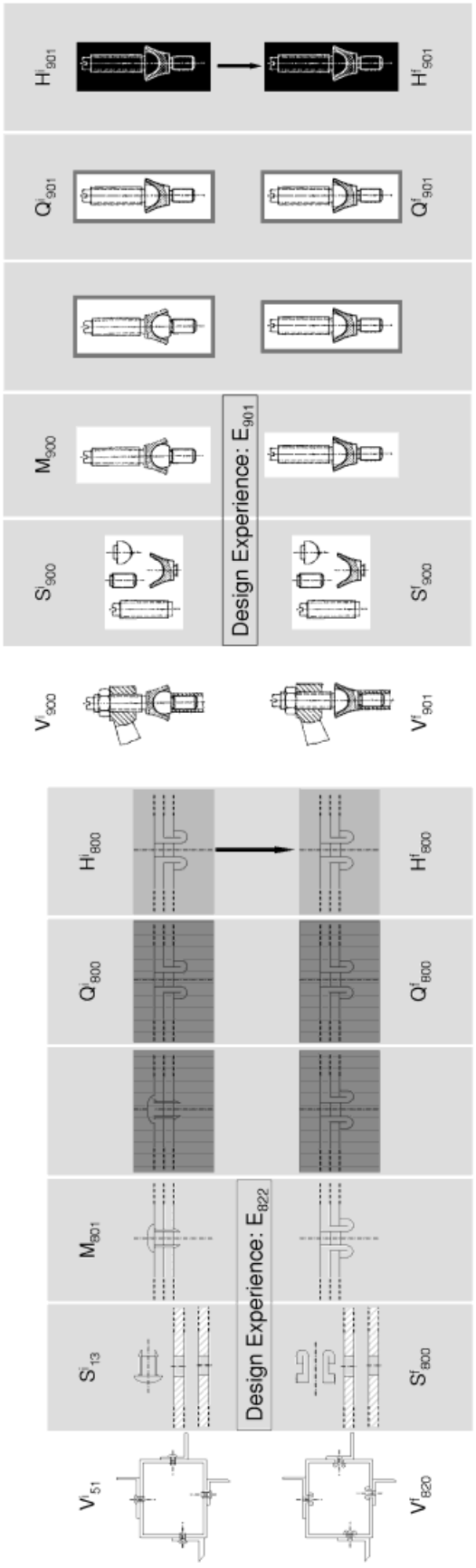


Fig. 8. The initial design experiences processed by the agent; the last experience is different from the rest to allow the demonstration of different behaviors through the grounding of a totally unrelated experience.

the focused concept from an activated experience selected for the current memory construction;

- $G_{E_{200}}$: ground experience E_{200} again;
- P_1: cycle the memory system according to the cue: $\{V^{i}_{200}, H^{i}_{200}$, and $H^{f}_{200}\}$ (representing the same situation encountered later) and complete the cue using the focused concept from an activated experience selected for the current memory construction.

Only one activated experience is selected for use in one memory construction based on a single cue. The selected experience is also assumed to be applicable to the current situation. A memory system may respond with more than one experience, but only one experience is arbitrarily selected here to illustrate the required operating mode and response.

The results from each experiment (see Fig. 9 for an example) are illustrated in the simplified form of Figure 1. Only elements relevant to illustrate the effects of grounding are included. Each figure (Figs. 9–19) illustrates the internal states of the memory system through the display of positive activation values (labeled y within brackets in the figures) of various instance nodes (labeled E_x in the figures) within the extended IAC network. An experience above an instance node label (E_x) corresponds to an instance node within the extended IAC network.

4.2. Common response via different operating modes

The first set of experiments (Experiments 1–3) demonstrates the different operating modes of separate memory systems due to the grounding of different experiences. All the memory systems start off the same and produce the same response in the same situation later although the internal mechanisms of these systems operate differently.

The following cases are considered:

- different operating modes of separate memory systems in the same situation due to the grounding of different activated experiences (Experiments 1 and 2); and
- different operating modes of separate memory systems in the same situation due to the grounding of different activated experiences across a different situation (Experiment 3).

4.2.1. Different operating modes in the same situation

The purpose of Experiments 1 and 2 is to highlight the effects of grounding different experiences on the operating modes of separate memory systems (residing in different agents) in a common situation. Two experiments that produce the same response are performed to illustrate the internal changes within different memory systems before the threshold that produces a different response is crossed. The

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

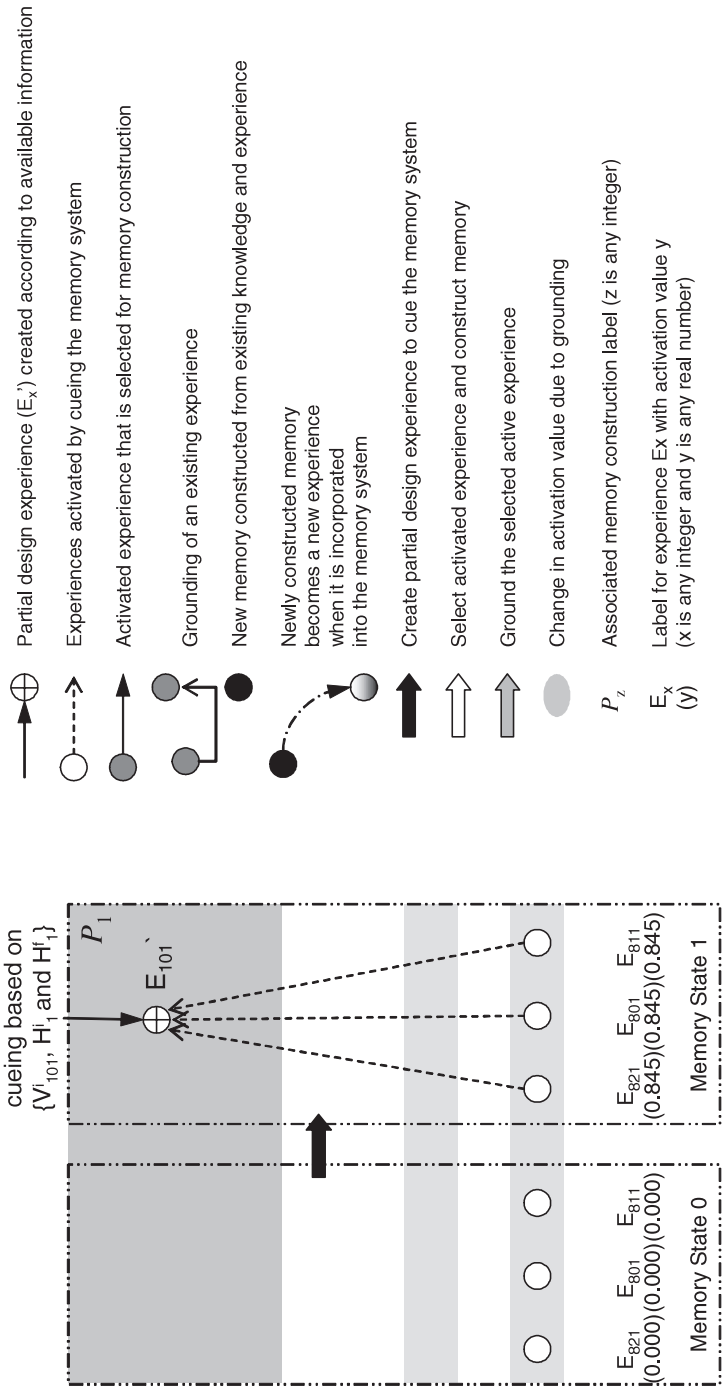


Fig. 9. Different states of the memory system for Experiment 1, Run 1.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

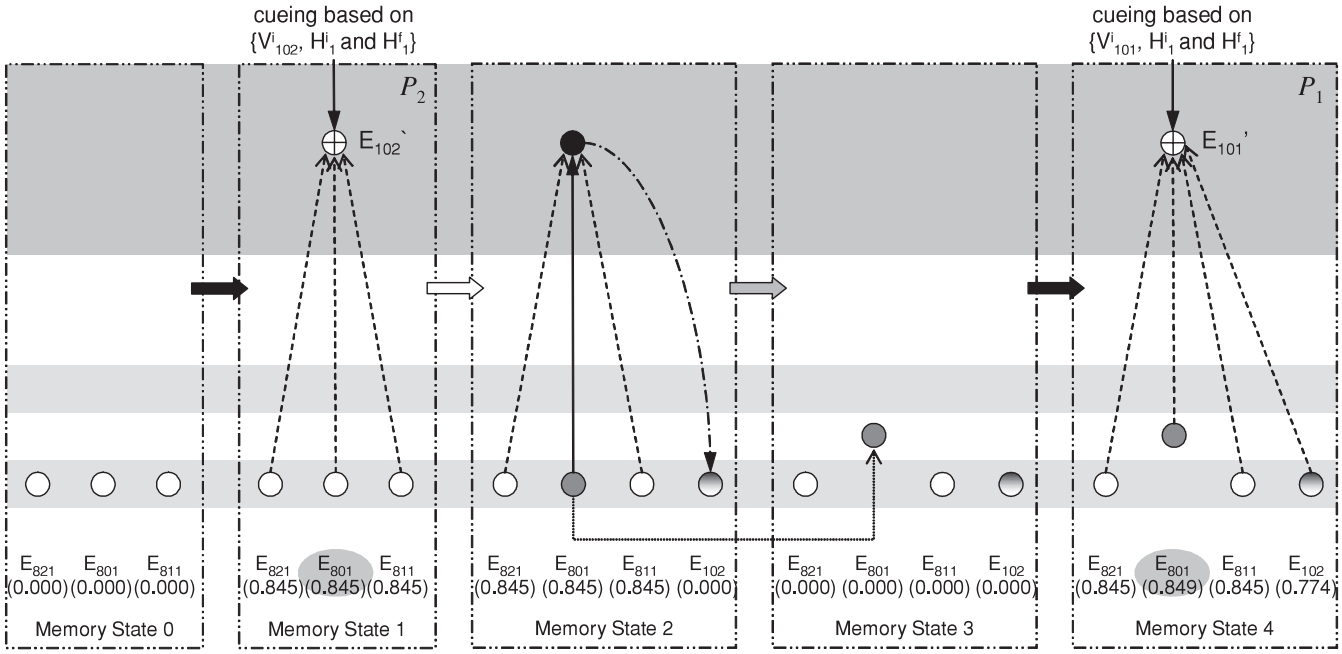


Fig. 10. Different states of the memory system for Experiment 1, Run 2.

two experiments, each with two runs of memory constructions, are

- Experiment 1:
  - Run 1: P_1;
  - Run 2: $P_1 \rightarrow G_{E_{801}} \rightarrow P_1$.

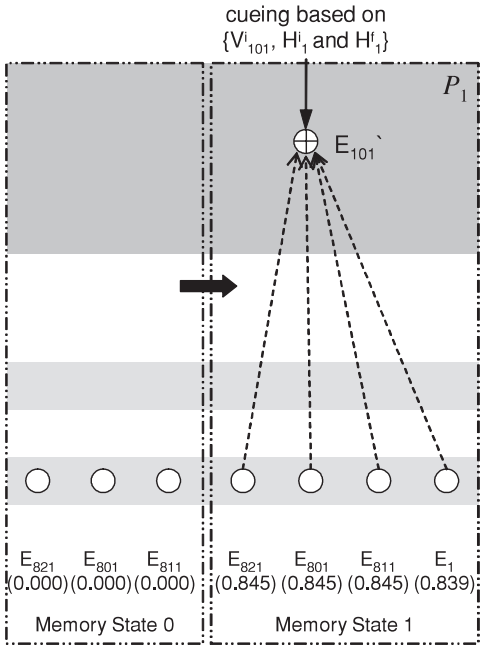


Fig. 11. Different states of the memory system for Experiment 2, Run 1 and Experiment 4, Run 1.

- Experiment 2:
  - Run 1: P_1;
  - Run 2: $P_3 \rightarrow G_{E_1} \rightarrow P_1$

where P_1 has a cue based on $\{V^{\mathrm{i}}_{101}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, P_2 has a cue based on $\{V^{\mathrm{i}}_{102}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, and P_3 has a cue based on $\{V^{\mathrm{i}}_{104}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$. The first run in each experiment represents a common memory system used in both experiments. The effects of grounding E_{801} and E_1 on the operating modes of separate memory systems in the same situation are illustrated. Among these memory systems,

- the same response within the same situation is indicated by the same experience being activated with the maximum activation value when their respective agents encounter the same situation;
- different operating modes are indicated by different maximum activation values (in terms of their magnitudes) of the common response (experience) from each system when their respective agents encounter the same situation; and
- the same situation encountered by their respective agents are indicated by the use of the same cue.

Experiment 1. Figures 9 and 10 illustrate the internal states of two memory systems based on the Run 1 (P_1) and 2 ($P_2 \rightarrow G_{E_{801}} \rightarrow P_1$) of Experiment 1. The response of each system within a common situation (modeled by P_1 at the end of each run) is the same (E_{801} having the maximum activation value) but each memory system operates differently to arrive at this response. This is indicated by the different activation values of E_{801} (0.845 at the end of

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

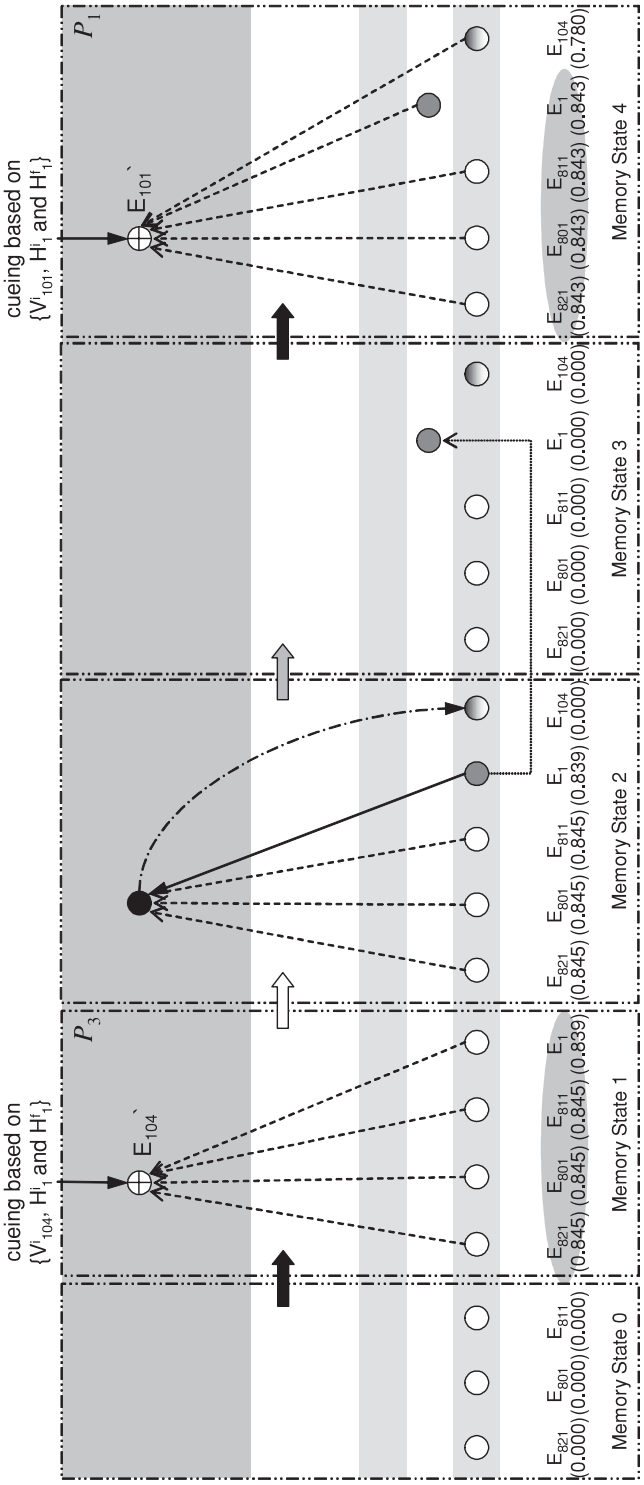


Fig. 12. Different states of the memory system for Experiment 2, Run 2.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

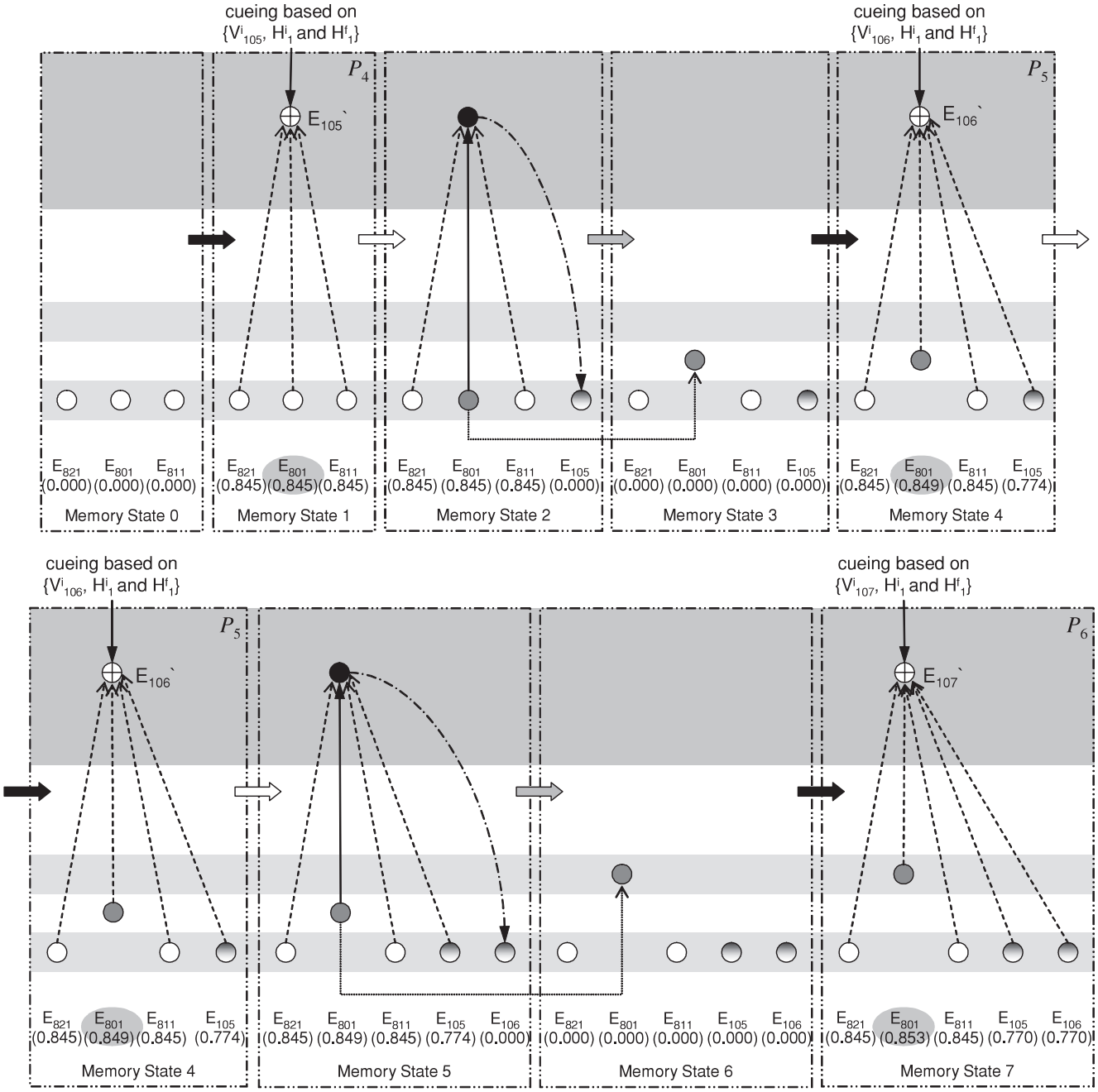


Fig. 13. Different states of the memory system for Experiment 3, Run 1. Memory State 4 is repeated in the bottom row to show the continuation from the top.

Run 1, Memory State 1, Fig. 9; and 0.849 at the end of Run 2, Memory State 4, Fig. 10) within the two memory systems. The difference in value is significant. The increase in activation is due to the additional grounding of experience E_{801} thus increasing its likelihood of usage within the corresponding memory system.

Discussion of results: Experiment 1. The two runs of the experiment illustrate the memory construction histories of two separate agents through a series of memory states. Although both agents start off with the same set of experiences (Fig. 8), the second agent (with history modeled by Experiment 1, Run 2) learned about the usefulness of experience E_{801}. This exposure to the use of E_{801} causes its memory system to respond with E_{801} at a higher activation value when both agents are placed in the same situation later.

When interpreted within the domain of DFA, the behaviors of the two agents can be interpreted as follows: both agents start off with the same set of experiences that deals

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

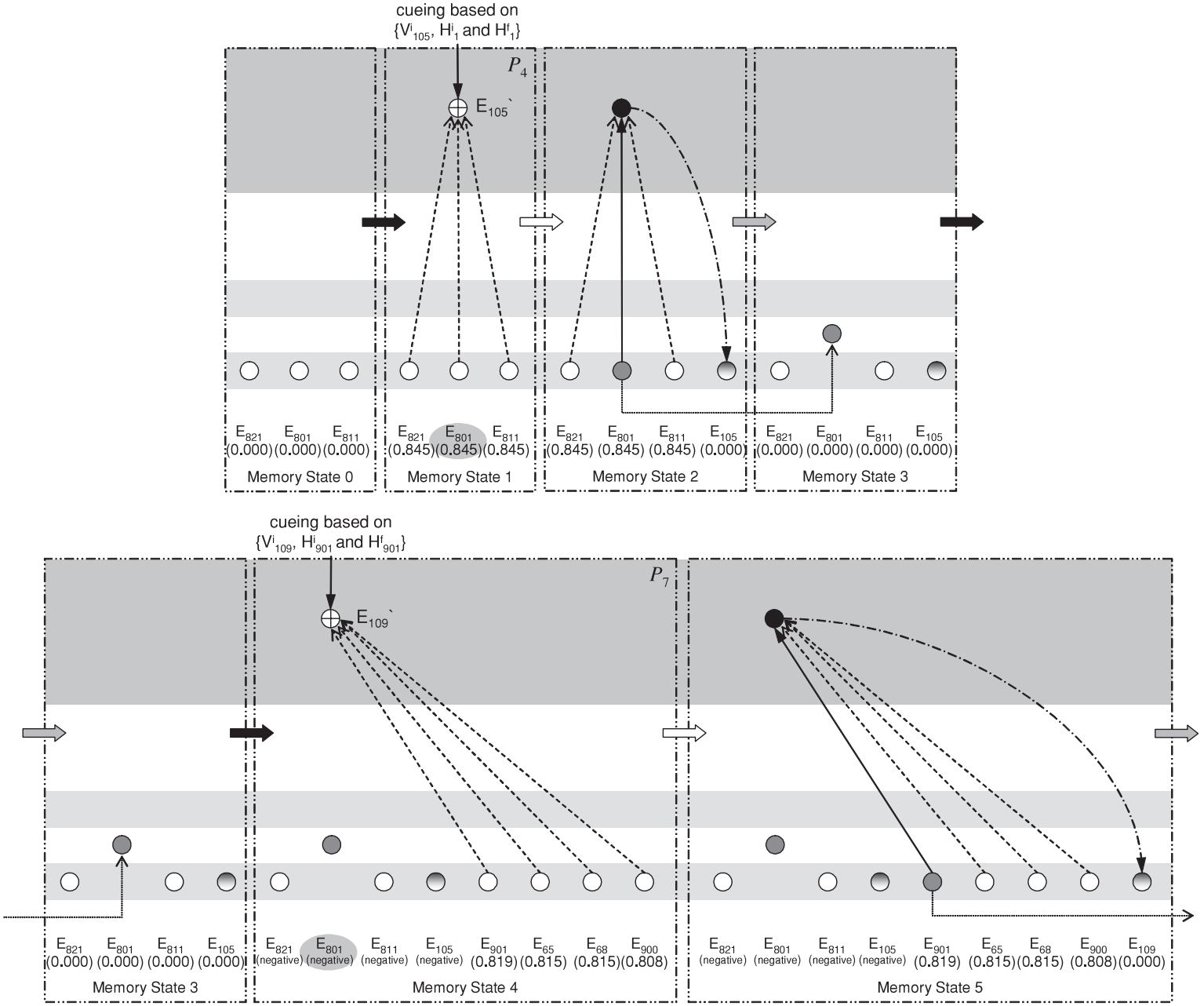


Fig. 14. Memory States 0–5 of the system for Experiment 3, Run 2. Memory State 3 is repeated in the bottom row to show the continuation from the top.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

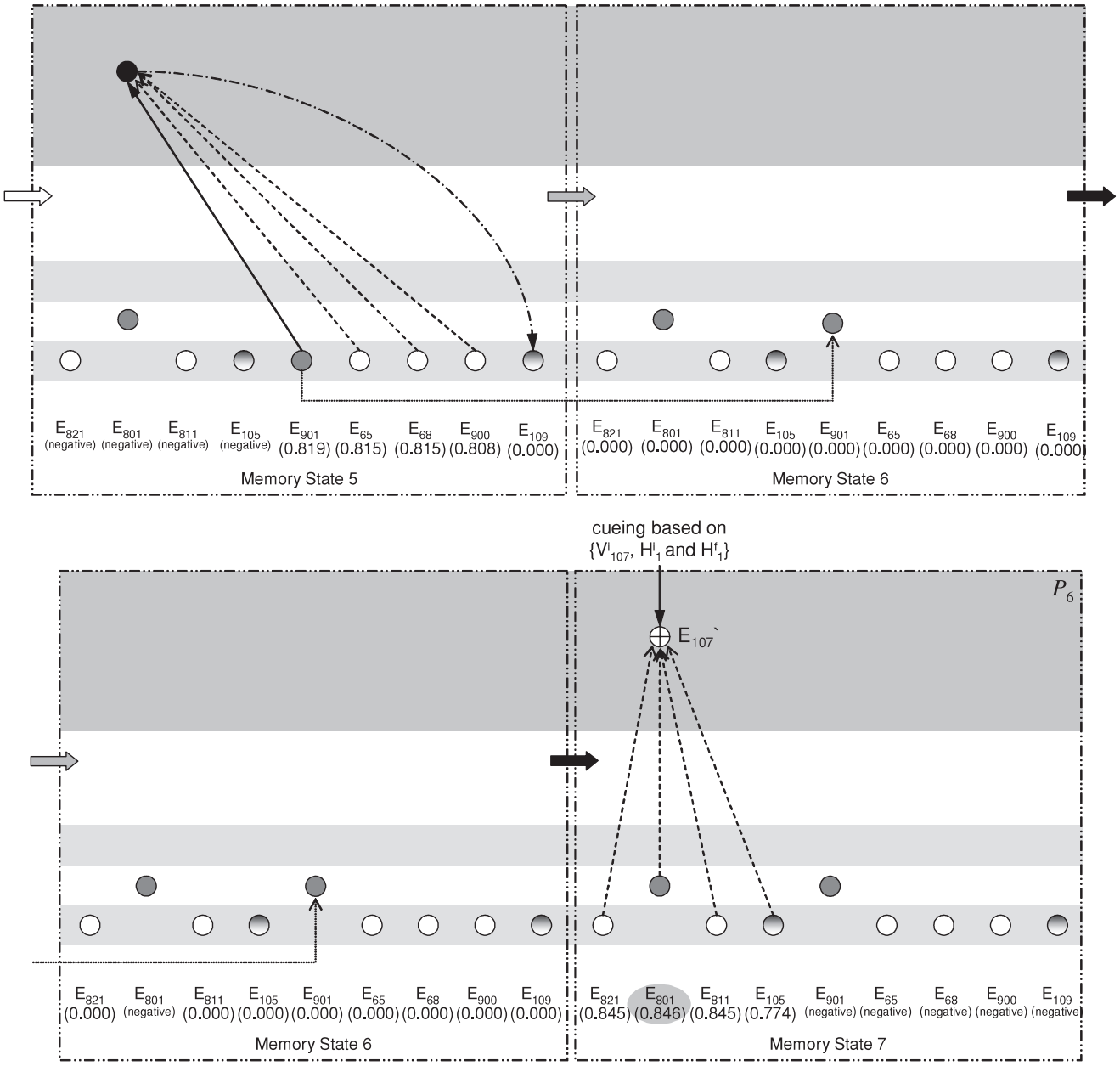


Fig. 15. Memory States 5–7 of the system for Experiment 3, Run 2. Memory State 5 is repeated in the top row to show the continuation from the previous diagram (Fig. 14), and Memory State 6 is repeated in the bottom row to show the continuation from the top row of the current figure.

with redesigning according to DFA. Before being put into the same situation, the second agent encounters an external structure (V^{i}_{102}) that it tries to redesign according to a DFA guideline that proposes the use of a snap-fit assembly (H^{i}_{1} and H^{f}_{1}). This agent creates a partial design experience E'_{102} according to these parameters $\{V^{i}_{102}, H^{i}_{1}$, and $H^{f}_{1}\}$, and uses it as a cue for memory construction (Memory State 1, Fig. 10). The memory system responds with an experience E_{801} that proposes the use of an integrated snap-fit assembly (H^{i}_{800} and H^{f}_{800}). This preference for integrated snap-fit is caused by a larger number of experiences that uses integrated snap-fit assemblies for better performance in Figure 8. The initial guideline modeled by H^{i}_{1} and H^{f}_{1} is also modeled by H^{i}_{800} and H^{f}_{800}, but the latter case is given preference (constructive learning allows a new focused concept to model a DFA guideline in place of another focused concept used previously, however, this part of the research is not within the scope of this paper). Through the selection of E_{801}, the agent constructs the required memory and restructures itself and the environment for the application of the integrated snap-fit assembly.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

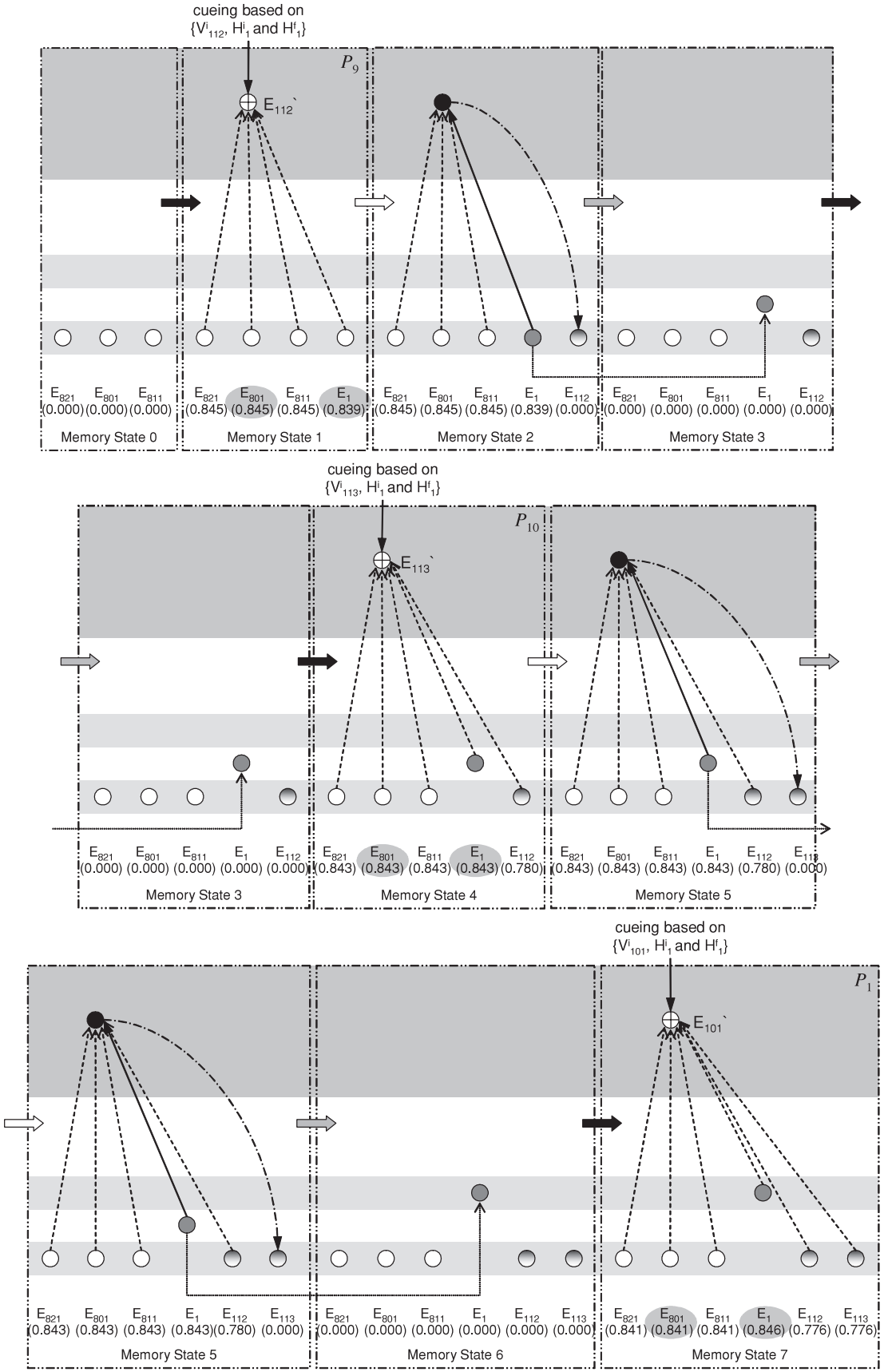


Fig. 16. Different states of the memory system for Experiment 4, Run 2. Memory States 3 and 5 are repeated in the bottom row to show the continuation from the top row.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

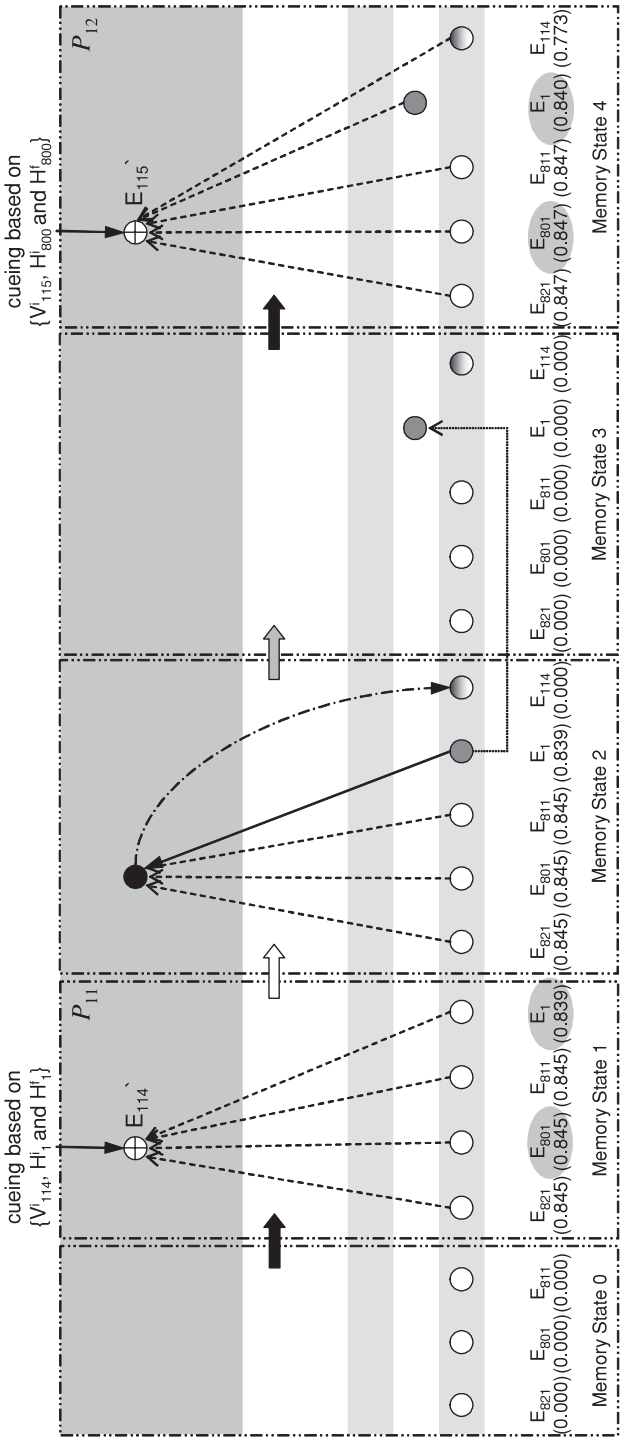


Fig. 17. Different states of the memory system for Experiment 5, Run 1.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

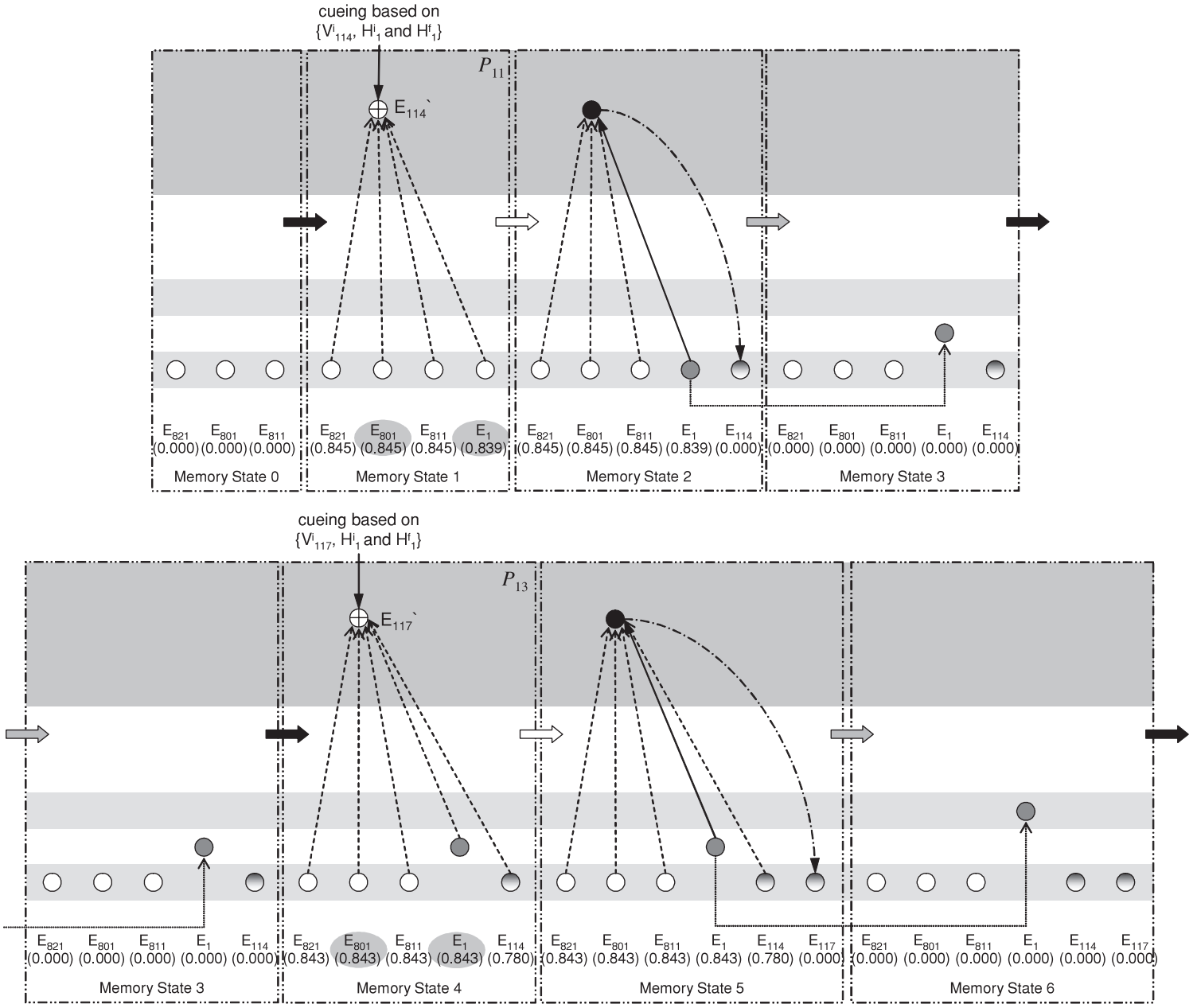


Fig. 18. Memory States 0–6 of the system for Experiment 5, Run 2. Memory State 3 is repeated in the bottom row to show the continuation from the top row of the current figure.

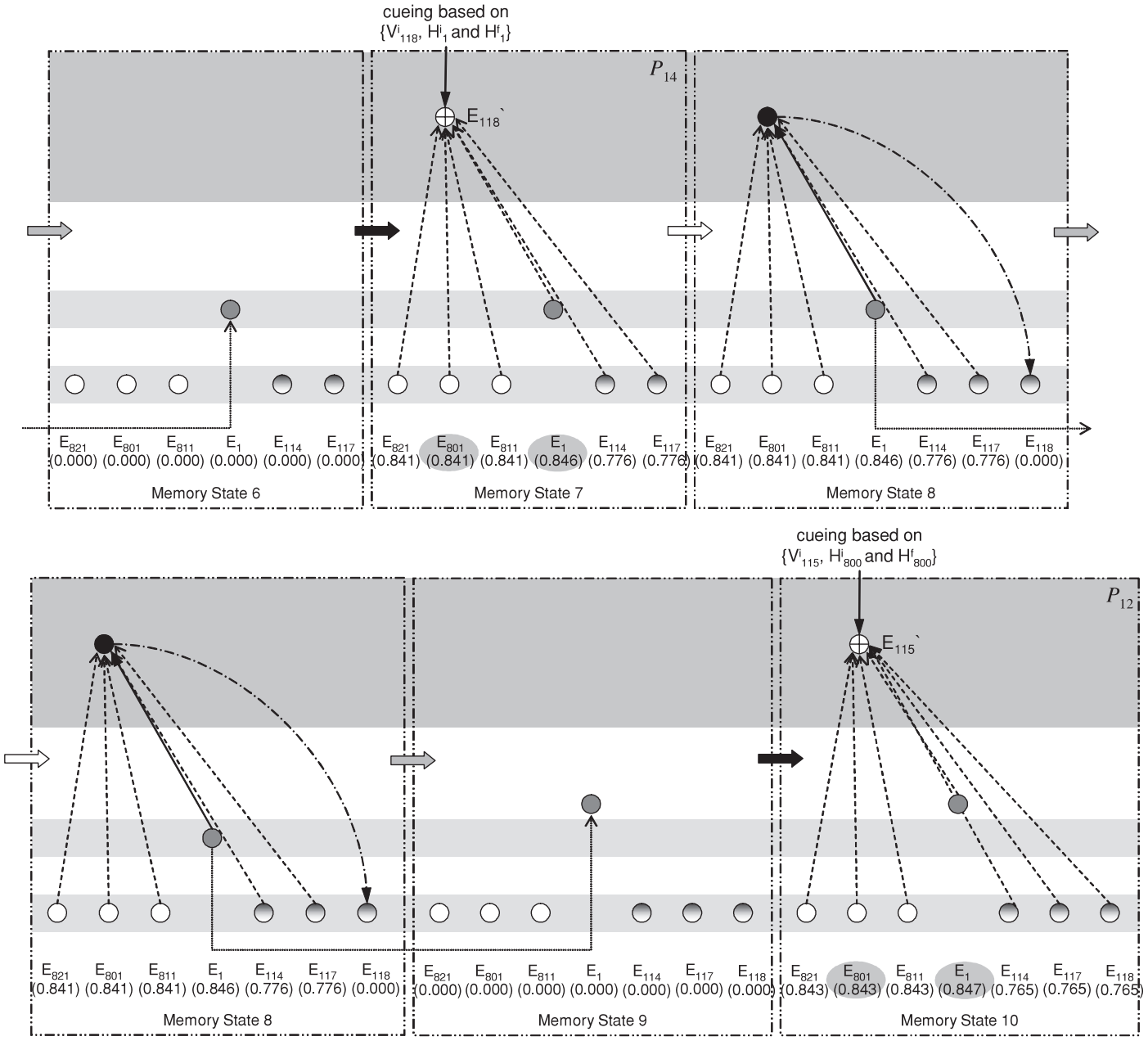


Fig. 19. Memory States 6–10 of the system for Experiment 5, Run 2. Memory State 6 is repeated in the top row to show the continuation from the previous diagram (Fig. 18), and Memory State 8 is repeated in the bottom row to show the continuation from the top row of the current figure.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

The selected experience (E_{801}) is grounded (Memory States 2 and 3, Fig. 10). This makes the likelihood of responding with the integrated snap-fit assembly even more likely in subsequent memory constructions. This higher probability of usage is demonstrated when both agents are exposed to the same situation later. Both memory systems are cued with the same partial design experience E'_{101} (Memory State 1, Fig. 9, and Memory State 4, Fig. 10), and each responded with E_{801} (the use of an integrated snap-fit assembly) through different emphasis to reflect its previous usage within each system.

The difference in operating modes caused by grounding gives different agents the potential to behave differently within the same situation according to what they have learned to be useful. The increase in activation value of a grounded experience in subsequent operations of its corresponding memory system is indicative of the possibility that the response of the system can be altered if sufficient grounding is provided. In terms of DFA, this means that it is possible to change the proposed use of an integrated snap-fit assembly to that of a snap-fit assembly (as the response of the system) if an experience containing the snap-fit assembly (E_1) is given sufficient grounding. This hypothesis is based on the increase in activation value of E_{801} due to its grounding. If E_1 is given enough grounding, it is postulated that its activation value will be elevated to such an extent that the system responds with E_1 instead of E_{801}. The next experiment illustrates the boundary condition just before this happens.

Experiment 2. Figures 11 and 12 illustrate the internal states of two memory systems based on runs 1 (P_1) and 2 ($P_3 \rightarrow G_{E_1} \rightarrow P_1$) of Experiment 2. This time experience E_1 is grounded in the second memory system (with history modeled by Experiment 2, Run 2). The response of each system within a common situation (modeled by P_1 at the end of each run) is the same (E_{801} having the same maximum activation value) but each memory system operates differently to arrive at this response. This is indicated by the different activation values of E_{801} (0.845 at the end of Run 1, Memory State 1, Fig. 11, and 0.843 at the end of Run 2, Memory State 4, Fig. 12) within the two memory systems. The decrease in activation is due to the grounding of experience E_1, which causes its likelihood of usage to increase within the corresponding memory system.

Discussion of results: Experiment 2. Although both agents start off with the same set of experiences, the second agent learned about the usefulness of experience E_1. This exposure to the use of E_1 causes its memory system to increase the activation value of E_1 when both agents are placed in the same situation later.

In terms of DFA, both agents start off with the same set of experiences. Before being put into the same situation, the second agent encounters an external structure (V^{i}_{104}) that it tries to redesign according to a DFA guideline that proposes the use of a snap-fit assembly (H^{i}_1 and H^{f}_1). This agent creates a partial design experience E'_{104} according to these parameters $\{V^{\text{i}}_{104}, H^{\text{i}}_1$, and $H^{\text{f}}_1\}$, and uses it as a cue for memory construction (Memory State 1, Fig. 12). The memory system responds with an experience E_{801} that proposes the use of an integrated snap-fit assembly (H^{i}_{800} and H^{f}_{800}). The proposed integrated snap-fit assembly is not used this time. The system is forced to use the snap-fit assembly based on E_1 in the current situation. This may be due to a change of conditions that renders the production cost of an integrated snap-fit assembly to be higher and thus a snap-fit assembly is preferred. Through the selection of this experience, the agent constructs the required memory and restructures itself and the environment for the application of the snap-fit assembly.

The selected E_1 is grounded (Memory States 2 and 3, Fig. 12). This makes the likelihood of responding with the snap-fit assembly even more likely in subsequent memory constructions. This higher probability of usage is demonstrated when both agents are exposed to the same situation later. Both memory systems are cued with the same partial design experience E'_{101} (Memory State 1, Fig. 11, and Memory State 4, Fig. 12). Although the second memory system responds with the same suggestion of using the integrated snap-fit assembly, this time it is given less emphasis, as indicated by the decrease of activation of E_{801} (from 0.845 in Run 1, Memory State 1, Fig. 11 to 0.843 in Run 2, Memory State 4, Fig. 12). The activation of E_1 has increased (from 0.839 in Run 1 to 0.843 in Run 2) due to its previous grounding. When the second memory system of both Experiments 1 and 2 are compared, the final activation of E_{801} is also observed to decrease (from 0.849 in Experiment 1, Run 2, Memory State 4, Fig. 10, to 0.843 in Experiment 2, Run 2, Memory State 4, Fig. 12). This is indicative of the change within the two memory systems due to the learning of different experiences.

The behavior of the second memory system in this experiment is starting to confirm the hypothesis stated in Experiment 1 that it is possible to alter the response of the system if sufficient grounding is provided. The activation value of any activated E_1 can be increased through subsequent grounding to enable it to gain maximum activation and to alter the response of the system in the same situation subsequently. This alteration of response is demonstrated in Experiment 4.

4.2.2. Different operating modes across different situations

The purpose of Experiment 3 is to highlight the effects of grounding on the operating modes of separate memory systems (residing in different agents) in a common situation through the grounding of different experiences across a different situation. The emphasis is on showing the different operating modes of separate memory systems caused by the learning of a different experience in another situation. The experiment is performed without crossing the threshold of response change, to produce similar responses

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

to illustrate the internal changes within each memory system. One experiment with two runs of memory construction is considered.

- Experiment 3:
  - Run 1: $P_4 \rightarrow G_{E_{801}} \rightarrow P_5 \rightarrow G_{E_{801}} \rightarrow P_6$;
  - Run 2: $P_4 \rightarrow G_{E_{801}} \rightarrow P_7 \rightarrow G_{E_{901}} \rightarrow P_6$,

where P_4 and P_6 have cues based on $\{V^{\mathrm{i}}_{105}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$ and $\{V^{\mathrm{i}}_{107}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, respectively; P_5 has a cue based on $\{V^{\mathrm{i}}_{106}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$; and P_7 has a cue based on $\{V^{\mathrm{i}}_{109}, H^{\mathrm{i}}_{901}$, and $H^{\mathrm{f}}_{901}\}$. The effects of grounding E_{901} in a totally unrelated situation on the operating mode of a memory system in the same situation are illustrated.

Experiment 3. Figures 13, 14, and 15 illustrate the internal states of two memory systems based on Run 1 ($P_4 \rightarrow G_{E_{801}} \rightarrow P_5 \rightarrow G_{E_{801}} \rightarrow P_6$), and Run 2 ($P_4 \rightarrow G_{E_{801}} \rightarrow P_7 \rightarrow G_{E_{901}} \rightarrow P_6$) of Experiment 3. The response of each system within a common situation (modeled by P_6 at the end of each run) is the same (E_{801} having the maximum activation value) but each memory system operates differently to arrive at this response. This is indicated by the reduced activation value of E_{801} (from 0.853 at the end of the first run, Memory State 7, Fig. 13, to 0.846 at the end of the second run, Memory State 7, Fig. 15) within the second memory system (modeled by the second run) due to the grounding of E_{901} in another situation. As E_{901} is grounded, all other experiences within the system are subjected to a decay process. The effect of grounding that occurs previously ($G_{\mathrm{E}_{801}}$) is reduced. The increase of activation value due to grounding within the memory system is not permanent. When an experience (E_{801}) is not used in subsequent memory constructions, the effect of its previous grounding will eventually wear off. The memory system's mode of operation changes to reflect the effect of "use it or lose it" as a result of grounding another experience in a different situation.

Discussion of results: Experiment 3. This experiment shows that what an agent has found to be useful is subjected to its continual usage. If the usefulness of an experience (E_{801}) is not reinforced in subsequent situations, its elevated emphasis due to a previous utility will eventually wear off.

In terms of DFA, this means that the increase in emphasis on the use of a snap-fit assembly based on grounding alone (Experiment 2) is not permanent. Only through the increase in the number of new experiences created with more use of snap-fit assemblies in subsequent situations will the emphasis on the use of the snap-fit assembly be permanent. However, the effect of these new experiences cannot be seen immediately until a sufficient number of them have been created. This effect due to the number of experiences can be seen in Experiments 1 and 2 where a memory system responds with an integrated snap-fit assembly when it is cued with a snap-fit assembly initially.

4.3. Different responses via different operating modes

The second set of experiments (Experiments 4 and 5) demonstrates the different outputs produced by separate memory systems due to the amount of grounding. All memory systems start off the same but produce different responses in the same situation later.

4.3.1. Different responses in the same situation

The purpose of the Experiments 4 and 5 is to demonstrate that with enough grounding, a memory system can be forced to respond with another activated experience when the threshold for response change has been reached. This behavior is important to situate the operations of the memory system according to the current interactions with the environment. Two experiments, each with two runs of memory constructions are considered:

- Experiment 4:
  - Run 1: P_1;
  - Run 2: $P_9 \rightarrow G_{E_1} \rightarrow P_{10} \rightarrow G_{E_1} \rightarrow P_1$.
- Experiment 5:
  - Run 1: $P_{11} \rightarrow G_{E_1} \rightarrow P_{12}$;
  - Run 2: $P_{11} \rightarrow G_{E_1} \rightarrow P_{13} \rightarrow G_{\mathrm{E}_1} \rightarrow P_{14} \rightarrow G_{\mathrm{E}_1} \rightarrow P_{12}$

where P_1 has cue based on $\{V^{\mathrm{i}}_{101}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, P_9 has a cue based on $\{V^{\mathrm{i}}_{112}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, P_{10} has a cue based on $\{V^{\mathrm{i}}_{113}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, P_{11} has cues based on $\{V^{\mathrm{i}}_{114}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, P_{12} has cues based on $\{V^{\mathrm{i}}_{115}, H^{\mathrm{i}}_{800}$, and $H^{\mathrm{f}}_{800}\}$, P_{13} has a cue based on $\{V^{\mathrm{i}}_{117}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$, and P_{14} has a cue based on $\{V^{\mathrm{i}}_{118}, H^{\mathrm{i}}_1$, and $H^{\mathrm{f}}_1\}$.

The effects of grounding E_1 repeatedly on the responses of separate memory systems in the same situation are illustrated. The suppression of the original response E_{801} by E_1 is also demonstrated. Among these memory systems,

- different responses within the same situation is indicated by different experiences being activated with the maximum activation value when their respective agents encounter the same situation;
- different operating modes are indicated by different maximum activation values (in terms of their magnitudes) of a common activated experience from each system when their respective agents encounter the same situation; and
- the same situation encountered by their respective agents are indicated by the use of the same cue.

Experiment 4. Figure 11 (reusing the results from Experiment 2, Run 1), and Figure 16 illustrate the internal states of two memory systems based on Runs 1 (P_1) and 2 ($P_9 \rightarrow G_{E_1} \rightarrow P_{10} \rightarrow G_{E_1} \rightarrow P_1$) of Experiment 4. The response of the second system (modeled by Experiment 4, Run 2) is different from that of the first system (modeled by Experi-

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

ment 4, Run 1) in the same situation (modeled by P_1 at the end of each run). Due to its repeated used of E_1, the same memory system used in the second agent has changed its response from E_{801} to E_1. Its activation value of E_{801} has decreased from 0.845 (Memory State 1, Fig. 16) to 0.843 (Memory State 4, Fig. 16) and finally to 0.841 (Memory State 7, Fig. 16), while the activation value of E_1 has increased from 0.839 to 0.843 and finally to 0.846.

Discussion of results: Experiment 4. This experiment confirms the hypothesis that it is possible to alter the response of a memory system if sufficient grounding is provides as postulated in Experiment 1. Although both agents start off with the same set of experiences, the second agent learned about the usefulness of E_1 repeatedly. This exposure to the use of E_1 causes its memory system to change its response when both agent are placed in the same situation. In terms of DFA, as the snap-fit assembly is repeatedly used, the groundings of the E_1 has effectively altered the suggested use of an integrated snap-fit assembly (H^{i}_{800} and H^{f}_{800} in E_{801}) to a snap-fit assembly (H^{i}_{800} and H^{f}_{800} in E_1).

This change of response within the memory system implies that even though an agent may have learned of the usefulness of an integrated snap-fit assembly in previous situations, the use of this learned usage is not absolute, as it is subjected to the current interactions between the agent and the environment. The application of knowledge (focused concepts) is situated. Situations may exist that render the use of the integrated snap-fit assembly infeasible, and these situations will cause the system to respond differently in other similar situations. The system updates what it had learned previously according the most recent interactions with the environment.

Experiment 5. Figures 17, 18, and 19 illustrate the internal states of two memory systems based on Run 1 ($P_{11} \rightarrow G_{E_1} \rightarrow P_{12}$) and Run 2 ($P_{11} \rightarrow G_{E_1} \rightarrow P_{13} \rightarrow G_{E_1} \rightarrow P_{14} \rightarrow G_{E_1} \rightarrow P_{12}$) of Experiment 5. The situation at the final memory construction stage is deliberately chosen to highlight the influence of grounding in settings that would normally cause a memory system to behave differently.

After the first memory system (modeled by Experiment 5, Run 1) has learned of the usefulness of E_1, E_1 is grounded instead of E_{801} (the activated experience with the maximum activation). This grounding is attempting to change the subsequent response of the memory system from E_{801} to E_1. However, the effect of this grounding is not strong enough. When the memory system is cued with H^{i}_{800} and H^{f}_{800} (contained within E_{801}) in the final situation (P_{12}), it responds with E_{801} with a value of 0.847 over E_1 with a value of 0.840 (Memory State 1, Fig. 17).

The second memory system (modeled by Experiment 5, Run 2) undergoes three consecutive groundings of experience E_1. This grounding changes the subsequent response of the memory system from E_{801} to E_1 (containing H^{i}_1 and H^{f}_1). The system responds with E_1 with a value of 0.847 over E_{801} with a value of 0.843 when exposed to the same final situation as that of the first agent (Memory State 10, Fig. 19).

Discussion of results: Experiment 5. From this experiment, it can be seen that the response change of a memory system due to grounding is observable even when the memory system is subsequently cued with a focused concepts H^{i}_{800} and H^{f}_{800}. These concepts (as cues) would normally give greater emphasis to E_{801} because they are part of the experience. However, the influence of grounding E_1 repeatedly is so strong that a memory system can be made to respond with E_1 (Experiment 5, Run 2, Memory State 10, Fig. 19) instead of E_{801} as it would have normally (Experiment 5, Run 1, Memory State 4, Fig. 17).

In terms of DFA, the repeated use of snap-fit assemblies (as proposed through E_1) by a agent, via its interactions with the environment, has the effect of forcing the use of a snap-fit assembly over the use of an integrated snap-fit assembly as proposed initially by the system. This effect is also evident when its memory system is cued according to an integrated snap-fit assembly. This behavior highlights the strong influence of repeated groundings on the response of the system. Without grounding, the cue based on the integrated snap-fit assembly would normally produce a response that uses an integrated snap-fit assembly, E_{801} (Memory State 4, Fig. 17). With the repeated grounding of E_1, the response from the system is altered to one that proposes the use of the snap-fit assembly.

4.4. Summary of results

The experiments described in this paper demonstrate the effects of different amounts and types of grounding on the operating modes, and consequently, the responses of separate memory systems. This, in turn, illustrates the situated behaviors of agents containing these systems based on the utility (grounding through usage) aspect of situatedness in designing. Two agents based on two separate memory systems were used as vehicles to demonstrate the following:

1. the grounding of different experiences causes a memory system to operate differently in the same situation (Experiments 1 and 2);
2. the increased emphasis of a grounded experience is temporary and is subjected to decay as other experiences are grounded (Experiment 3);
3. with enough grounding, a memory system can be made to change its response according to what it has learned frequently in recent times (Experiment 4); and
4. the influence of multiple grounding is strong enough to alter the response of a memory system in situations that would otherwise evoke a different response (Experiment 5).

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

These behaviors of a memory system imply that the use of design experience and knowledge is highly dependent on what has been learned recently and frequently through the interactions between the agent and the environment. What was termed useful or "better" in the past is not absolute but varies according to most current interactions with the environment.

5. COMPARISON WITH OTHER EXPERIENTIAL-BASED DESIGN SYSTEMS

The model of constructive memory operates within a framework that is based on the idea of situatedness. This framework is the key element that sets the constructive memory system apart from other experiential-based design systems such as case-based design (CBD) systems (Maher et al., 1995), memory reconstructions system (Kumar & Raphael, 2001), and design systems based on dynamic memory (Williams, 1995).

5.1. Framework for memory construction

The notion of situatedness encompasses the fundamental ideas of *interaction*, *memory construction*, and *interpretation*. Interaction provides the basis for memory construction. Memory construction facilitates interpretations of the environment and interpretations of existing experiences according to the current situation. Based on this these interpretations, constructive learning occurs to facilitate the derivation and performance of actions for the fulfilment of an agent's goals.

Interaction implies that the contents of a memory system are not encoded *a priori* and indexed for use later, but rather, the knowledge and experiences are developed through interactions with the environment. The development of these contents entails the construction of memories about related past experiences according to the prevailing situation.

Memory construction provides the basis for interpretation. A constructed memory defines the interpretation of a relevant experience and the interpretation of the environment in light of the current interactions between the agent and the environment. The experiences processed by the system are interpreted through a "filter" defined by the present situation. A past experience is not "copied" into the present, but rather, it is interpreted according to the current situation. This interpretation situates the relevant experience through the current environment so that it is not necessary to encode all possible forms of related knowledge *a priori*. A new interpretation of an experience is produced by every constructed memory of that experience. This new interpretation is added to the experience (when the constructed memory is incorporated into the system as a new experience), and is available to be interpreted later as if it were part of the original experience. A constructed memory also interprets the environment according to the current expectation of the agent. This expectation is derived from the goal of the agent. The expectation dictates what is to be focused upon, and the way the environment is to be interpreted for the agent to perform its task according to its goal. The different interpretations of the environment are effected by biasing sensation, perception, and conception.

Interpretation of the current situation starts when an activated experience is selected for memory construction. This experience provides an interpretation of a previous situation consisting of an interpretation of a previous environment, and an interpretation of a related experience or knowledge. The interpretation of the previous environment provides the basis for interpreting the current environment. The interpretation of the current environment resulting from this interpretation, in turn, determines the way the selected experience is to be interpreted according to the current interaction. This new interpretation determines yet another new interpretation of the environment in a recursive manner. Both the environment and the experience are interpreted through the lens of the current interaction in this recursive fashion. The recursion behavior is resolved when the interpretations of the environment or experience remain the same after a complete cycle of interpretations: interpretation of the environment or experience followed by an interpretation of the experience or environment. A constructed memory provides the interpretation of the environment and the interpretation of a relevant experience within a single cohesive structure. When a memory has been successfully constructed, a coherent interpretation of both the environment and the relevant experience is created and the recursive behavior is resolved. The experiences contained within a memory system play an important role during interpretation. Each experience processed by an agent provides a different starting point for the recursive interpretations of the current situation that result in different memories being constructed.

Through successful interpretations of the environment and a relevant experience selected for memory construction, constructive learning occurs. Three cases of successful interpretations are possible:

1. successful interpretations according to a past interpretation modeled by the selected experience (Interpretation I);
2. successful interpretations through an external process that reinterprets the environment according to a transformed experience derived from the selected experience (Reinterpretation II); and
3. successful interpretations through a reconfiguration of the memory system that reinterprets the selected experience according to elements from the environment (Reinterpretation I).

In the case of Interpretation I, the current environment is interpreted according to the selected experience without any

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

further processing. In the case of Reinterpretation II, the environment cannot be interpreted according to the interpretation of a previous environment provided by the selected experience. An external process based on analogy is used to transform the selected experience into a form suitable for reinterpreting the current environment. In the case of Reinterpretation I, the environment provides the structure necessary for reinterpreting existing experiences. This occurs when structures from the environment have better performance than those contained within the selected experience in terms of the agent's goal.

After the successful interpretation of the environment and experience, the actions that were performed previously are transformed into a suitable form for the current situation and effected into the environment, as external actions, or to the agent, as internal actions, to perform the necessary task according to the current goal of the agent.

5.2. CBD systems

CBD is a form of analogic reasoning that uses previous design episodes to generate new designs (Maher et al., 1995). The episodes are usually represented as a repository of design cases. These cases are recalled during designing to provide the bases upon which solutions are created. The fundamental processes that are involved in using case-based reasoning for designing are

- case recall where relevant cases are retrieved from the case repository; and
- case adaptation where the selected case is modified to produce a solution according to the current situation.

The operations of case adaptation are very similar to those of memory construction described in this paper. Both process uses an analogic process that matches and subsequently maps structures from the environment to those contained within the system and both learn through the application of old experiences in new situations. There are, however, fundamental differences between the two approaches to designing based on the different frameworks within which they operate.

Case retrieval within a CBD system is based on pattern matching between specific features from the environment and indices contained within the design cases (Maher et al., 1995). A related case is retrieved from the case memory without modification according to these indices. A constructed memory, on the other hand, is constructed according to the interpretation of a related design experience that is activated by cues created from the current situation. This activated experience is retrieved without modification from the memory system to provide a starting point for memory construction. During construction, this retrieved experience is interpreted according to the current situation. The final form of the constructed memory may appear in a form totally different from that of the activated experience. This interpretation mechanism allows related "cases" to be constructed "on the fly" as required by the current situation without the need to enumerate all possible "cases" *a priori*.

During case adaptation, the differences between the contents of the retrieved case and the current situation determine the type and amount of modification that is to be made to the elements within the case. The adaptation process is trying to "pull" the case toward the current situation presented by the environment to solve the current problem according to past experiences. The view of the environment is in no way affected by the adaptation process. A constructed memory, however, provides the necessary mechanisms that allow the environment to be interpreted. This interpretation of the environment provides the basis for the interpretation of the activated experience, which in turn, affects the way the environment is interpreted again. This feedback loop causes the two interpretations to occur recursively until a common interpretation is reached after a complete cycle of environment and experience interpretations.

The CBD system implicitly assumes that the case contained within the system will provide a better solution for the current design problem. During adaptation, the contents of the retrieved case are transformed into a form that is suitable to solve the current design problem. The use of a case is not situated, as its superior performance is assumed. In a constructive memory model of designing, this assumption is challenged as existing experiences are reinterpreted in view of the current situation when the constructed memory provides the mechanisms for interpreting the activated design experiences.

It is possible to embed the process of interaction, memory construction, and interpretation into a CBD system by stretching the definitions of case indexing, case retrieval, or case adaptation to produce the same results as those determined through a memory construction process. It should be possible to change the operations of a CBD system to conform to the framework dictated by the situatedness in designing. However, the resulting CBD system is nothing but an implementation of the concepts of constructive memory described here. This paper is concerned with the concepts associated to the computational model of a constructive memory system. These concepts are implemented in terms of an extended neural network. There is nothing inherent within the computational model that prevents the same ideas from being implemented in a CBD system.

6. CONCLUSION

The notion of memory involves a dynamic construction process. Memory is not a static imprint to be stored in a specific location and retrieved for use later. The memory construction processes are not predefined but are influ-

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

enced by the situatedness in designing. These behaviors of the memory system are dependent on the current environment, the current internal state of the agent, and the interactions between the agent and the environment. The work presented in this paper provides a perspective of constructive memory within the domain of designing that allows a computational constructive memory system to be developed within a situated design agent. A series of experiments is used to illustrate the effects of grounding what the agent knows according to the interactions between it and the external world. The quality of this knowledge is constantly improved through its utility so that when the agent is reused in a similar situation it will have different knowledge than it had before. This new knowledge has the potential to give the agent an ability to produce better results and improve its performance beyond what was initially encoded into the system.

ACKNOWLEDGMENTS

This research has been supported by a University of Sydney Postgraduate Research Award. Computational resources have been provided by the Key Centre of Design Computing and Cognition.

REFERENCES

Atkinson, R.C., & Shiffrin, R.M. (1968). Human memory: A proposed system and its control processes. In *The Psychology of Learning and Motivation: Advances in Research and Theory* (Spence, K.W., Ed.), pp. 89–195. New York: Academic.

Ausubel, D.P. (1968). *Educational Psychology: A Cognitive View*. New York: Holt, Rinehart & Winston.

Baddeley, A.D. (1999). Memory. In *The MIT Encyclopedia of the Cognitive Sciences* (Wilson, R.A., & Keil, F.C., Eds.), pp. 514–517. Cambridge, MA: MIT Press.

Boothroyd, G. (1994). Product design for manufacture and assembly. *Computer-Aided Design 26(7)*, 505–519.

Boothroyd, G., & Dewhurst, P. (1989). *Product Design for Assembly*. Wakefield, RI: Boothroyd Dewhurst, Inc.

Boothroyd, G., Dewhurst, P., & Knight, W. (1994). *Product Design for Manufacture and Assembly*. New York: Marcel Dekker.

Boothroyd, G., Poli, C., & Murch, L.E. (1982). *Automatic Assembly*. New York: Marcel Dekker.

Bruner, J.S. (1966). *Toward a Theory of Instruction*. New York: Norton.

Clancey, W.J. (1997). *Situated Cognition: On Human Knowledge and Computer Representations*. New York: Cambridge University Press.

Dewey, J. (1896). The reflex arc concept in psychology. *Psychological Review 3*, 357–370.

Fiesler, E., & Beale, R. (Eds.). (1997). *Handbook of Neural Computation*. New York: Oxford University Press.

Gero, J.S. (1990). Design prototypes: A knowledge representation schema for design. *AI Magazine 11(4)*, 26–36.

Gero, J.S. (1998*a*). Design tools that learn. *Advances in Engineering Software 29(10)*, 755–761.

Gero, J.S. (1998*b*). Towards a model of designing which includes its situatedness. In *Universal Design Theory* (Grabowski, H., Rude, S., & Green, G., Eds.), pp. 47–56. Aachen, Germany: Shaker Verlag.

Gero, J.S., & Fujii, H. (2000). A computational framework for concept formation in a situated design agent. *Knowledge-Based Systems 13(6)*, 361–368.

Gero, J.S., & Kannengiesser, U. (2000). Towards a situated function–behaviour–structure framework as the basis for a theory of designing. In *Workshop on Development and Application of Design Theories in AI in Design Research, Artificial Intelligence in Design '00* (Smithers, T., Ed.), pp. 1–5, Worcester, MA.

Gero, J.S., & Kannengiesser, U. (2002). The situated function–behaviour–structure framework. In *Artificial Intelligence in Design '02* (Gero, J.S., Ed.), pp. 89–104. Dordrecht: Kluwer Academic.

Gero, J.S., Tham, K.W., & Lee, H.S. (1992). Behaviour: A link between function and structure in design. In *Intelligent Computer Aided Design* (Brown, D.C., Waldron, M.B., & Yoshikawa, H., Eds.), pp. 193–225. Amsterdam: North–Holland.

Goker, M.H. (1997). The effects of experience during design problem solving. *Design Studies 18*, 405–426.

Goldschmidt, G. (1997). Capturing indeterminism: Representation in the design problem space. *Design Studies 18*, 441–455.

Haykin, S.S. (1998). *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice–Hall.

Hoehfeld, M., & Fahlman, S.E. (1992). Learning with limited numerical precision using the cascadecorrelation learning algorithm. *IEEE Transactions on Neural Networks 3(4)*, 602–611.

Jonassen, D.H. (1991). Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research and Development 40(3)*, 5–14.

Kumar, B., & Raphael, B. (2001). *Derivational Analogy Based Structural Design*. Stirling: Saxe–Coburg Publications.

Logie, R.H. (1995). *Visuo-Spatial Working Memory*. Hove: Erlbaum.

Lueg, C., & Pfeifer, R. (1997). Cognition, situatedness, and situated design. *Proc. Second Int. Conf. Cognitive Technology (CT 97)*, pp. 124–135, Aizu, Japan, August 25–28.

Maher, M.L., Balachandran, M.B., & Zhang, D.M. (1995). *Case-Based Reasoning in Design*. Mahwah, NJ: Erlbaum.

Mandic, D., & Chambers, J. (2001). *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Singapore: Wiley.

McClelland, J.L. (1981). Retrieving general and specific information from stored knowledge of specifics. *Proc. Third Annual Meeting of the Cognitive Science Society*, pp. 170–172, Hillsdale, NJ: Erlbaum.

McClelland, J.L. (1988). *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*. Cambridge, MA: MIT Press.

McClelland, J.L. (1995). Constructive memory and memory distortion: A parallel-distributed processing approach. In *Memory Distortion: How Minds, Brains, and Societies Reconstruct the Past* (Schacter, D.L., Ed.), pp. 69–90. Cambridge, MA: Harvard University Press.

Medler, D.A. (1998). A brief history of connectionism. *Neural Computing Surveys 1(1)*, 61–101.

Nehaniv, C., & Dautenhahn, K. (1998). Embodiment and memories—Algebras of time and history for autobiographic agents. *Proc. 14th European Meeting on Cybernetics and Systems Research Symposium on Embodied Cognition and Artificial Intelligence—Cybernetics and Systems '98*, Vol. 2, pp. 651–656, Vienna, Austria.

Nehaniv, C.L. (1999). Meaning for observers and agents. *Proc. IEEE Int. Symp. Intelligent Control/Intelligent Systems and Semiotics, ISIC/ISAS '99*, pp. 435–440, Cambridge, MA.

O'Reilly, R.C., & Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. Cambridge, MA: MIT Press.

Piaget, J. (1971). *Psychology and Epistemology: Towards a Theory of Knowledge*. New York: Harcourt, Brace.

Qian, L. (1994). *Creative design by analogy*. PhD Thesis. University of Sydney.

Reed, R. (1993). Pruning algorithms: A survey. *IEEE Transactions on Neural Networks 4*, 740–747.

Rosenfield, I. (1988). *The Invention of Memory: A New View of the Brain*. New York: Basic Books.

Rosenman, M.A., & Gero, J.S. (1993). Creativity in design using a design prototype approach. In *Modeling Creativity and Knowledge-Based Creative Design* (Gero, J.S., & Maher, M.L., Eds.), pp. 119–148. Hillsdale, NJ: Erlbaum.

Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press.

Russell, S., & Norvig, P. (1995). *Artificial Intelligence*. Englewood Cliffs, NJ: Prentice–Hall.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press

Rylatt, R.M., & Czarnecki, C.A. (1998). Beyond physical grounding and naive time: Investigations into short term memory for autonomous agents. In *From Animals to Animats, Proc. of the Fifth International Conference of the Society for Adaptive Behaviour, SAB 5* (Pfeifer, R., Ed.), pp. 17–21. Cambridge, MA: MIT Press.

Seifert, C.M., Patalano, A.L., Hammond, K.J., & Converse, T.M. (1997). Experience and expertise: The role of memory in planning for opportunities. In *Expertise in Context* (Feltovich, P.J., Ford, K.M., & Hoffman, R.R., Eds.), pp. 101–123. Cambridge, MA: MIT Press.

Szykman, S., Sriram, R., Bochenek, C., & Racz, J. (1998). The NIST design repository project. In *Advances in Soft Computing: Engineering Design and Manufacturing* (Roy, R., Furuhashi, T., Chawdhry, P.K., & Ruggieri, M., Eds.), pp. 5–19. London: Springer–Verlag.

Tait, B. (1997). Constructive internet based learning. *Active Learning 7*, 3–8.

Williams, P. (1995). *Dynamic memory for design*. PhD Thesis. University of Sydney.

Ziemke, T. (1999). Rethinking grounding. In *Understanding Representation in the Cognitive Sciences* (Riegler, A., Peschl, M., & Stein, A., Eds.), pp. 177–190. New York: Plenum.

Pak-San Liew is a graduate student at the Key Centre of Design Computing and Cognition, University of Sydney. He has completed his PhD on the development of a computational constructive memory system.

John S. Gero is Professor of design science and Director of the Key Centre of Design Computing and Cognition, University of Sydney. His research spans both design computing and design cognition.

https://doi.org/10.1017/S0890060404040120 Published online by Cambridge University Press