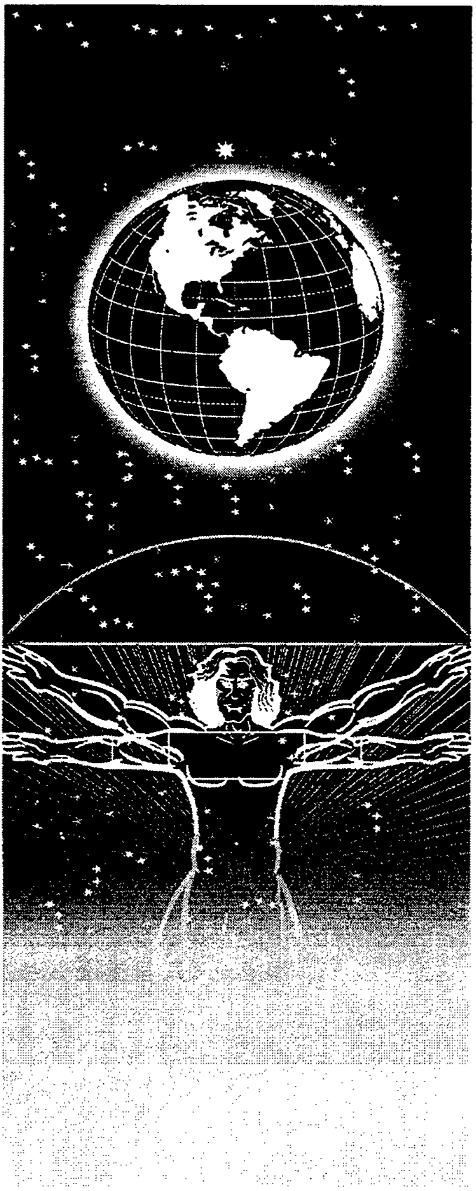
AL/HR-TR-1997- 0179



UNITED STATES AIR FORCE
ARMSTRONG LABORATORY

MENTAL MODELS AS FINITE-STATE MACHINES: EXAMPLES AND COMPUTATIONAL METHODS

Martin J. Ippel
NRC Senior Research Associate

HUMAN RESOURCES DIRECTORATE
MISSION CRITICAL SKILLS DIVISION
7909 Lindbergh Drive
Brooks AFB, Texas 78235-5352

A. Leo Beem

Department of Social Sciences
Leiden University
The Netherlands

October 1998

Approved for public release; distribution unlimited.

AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
HUMAN EFFECTIVENESS DIRECTORATE
7909 Lindbergh Drive
Brooks Air Force Base, TX   78235-5352

NOTICES

This report has been reviewed and is approved for publication.

MARTIN J. IPPEL, Ph.D.
Project Scientist

R. BRUCE GOULD, Ph.D.
Chief, Mission Critical Skills Division

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** October 1998	**3. REPORT TYPE AND DATES COVERED** Interim Report – March-June 1997

4. TITLE AND SUBTITLE Mental Models as Finite-State Machines: Examples and Computational Methods	**5. FUNDING NUMBERS** PE – 61102F PR – 2313 TA – T1 WU – 54
6. AUTHOR(S) Martin J. Ippel A. Leo Beem	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Human Effectiveness Directorate Mission Critical Skills Division 7909 Lindbergh Drive Brooks AFB, TX 78235-5352	**8. PERFORMING ORGANIZATION REPORT NUMBER** AL/HR-TR-1997-0179
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Technical Monitor: Dr. Patrick C. Kyllonen, (210) 536-3921

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (Maximum 200 words)

The term "mental model" often refers to an internal representation that can be mentally "run" to produce inferences, explanations, and predictions about the environment. Holland, Holyoak, Nisbett, and Thagard (1986) propose a formalism to capture the dynamics of mental models: a transition function defined on a set of model states, the result of a categorizing of environmental states. This transition function mimics the state changes that unfold in the environment. The paper shows that the addition of a few reasonable constraints to this formalism results in a class of transition functions with well-known properties—the general class of finite state machines. Finite state machines can be used to model interactions between humans and artifices (e.g., an application program, an ATM, or a car). We present a method to test a hypothesis involving the partitioning of a set of environment states into equivalence classes, which are identified as states of the model. The method is demonstrated on a spatial reasoning task performed by second and third grade children.

14. SUBJECT TERMS Categorization function, Cognitive skill, Finite-state machine, Human-computer interaction, Mental model, Transition function, User model	**15. NUMBER OF PAGES** 37
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

CONTENTS

FIGURES AND TABLES

Figure

Table

PREFACE

This report presents the results of a study that started at Leiden University, The Netherlands. The goal of the study was to develop a quantitative approach to test the efficacy of mental models that humans use to interact with artifices such as ATMs and computer programs. As a matter of good fortune the particular mental model studied appeared to have great relevance for the increased USAF interest in UAV technology. We hope the approach will be applied fruitfully in this context.

The major part of this work was performed while the first author was a Senior Research Associate with the National Research Council at the Armstrong Laboratory.

The authors are indebted to Dr. Richard Roberts and Dr. Anna Rowe for many suggestions that improved the quality of the report.

SUMMARY OF RESULTS

1. Holland, Holyoak, Nisbett, and Thagard (1986) propose a formalism to capture the dynamics of mental models: a transition function defined on a set of model states, the result of a categorizing of environmental states. This transition function mimics the state changes that unfold in the environment. The paper shows that the addition of a few reasonable constraints to this formalism results in a class of transition functions with well-known properties -- the general class of finite-state machines.
2. It is shown that finite-state machines can be empirically tested by contigency tables in which symbols of an input alphabet (rows) are mapped on symbols of an output alphabet (columns) and each state is represented as a different layer of cross-classification. Model testing of Probabilistic Finite Automata can be straightforwardly accomplished using chi-square based statistics.
3. Empirical evaluation of Deterministic Finite Automata can be accomplished by applying techniques derived from Information Theory. Information Theory defines a Deterministic Finite Automaton as a perfect channel. That is, the information transmitted is equal to the maximum uncertainty. Model deviations can be quantified as information loss (i.e., the difference between maximum uncertainty and information transmitted).

INTRODUCTION

Whenever someone has acquired the cognitive skill to use some interactive device (e.g., a fighter jet, a car, an ATM, or a word processing system), he or she is assumed to have a cognitive representation of it that is like a working model. Cognitive scientists have used the term 'mental model' in these contexts to refer to a theoretical construct that interrelates conceptual knowledge with procedural skills. The common core of many theoretical treatments is the notion that cognitive systems construct models of a particular content domain. These models can be mentally 'run', or manipulated to produce inferences, explanations, and predictions about the system (Holland, Holyoak, Nisbett, & Thagard, 1986; Holyoak, 1985; Holyoak, Koh, & Nisbett, 1989; Johnson-Laird, 1983, 1989; Payne, 1988; Rogers & Rutherford, 1992).

Holland et al. (1986) propose a transition function as a formalism to capture the dynamics of mental models. They propose to conceive of a mental model as a mental representation that encodes a particular environment into categories and subsequently employs such categories to define an internal transition function that mimics the state changes unfolding in this environment. A mental model is considered valid to the extent that the relationship between the mental model and the corresponding part of the environment is a homomorphism, that is, a many-to-one structure mapping of states, and state change operators, from the external environment to the mental model (Holyoak, 1985).[1] This abstract characterization of mental models is neutral concerning the issue of the information processing mechanisms that may be employed to construct mental models. However, Holland et al. (1986) argue that mental models are assembled from sets of production rules. The empirical credibility of their theory is tested through comparison of the performance of humans or lower animals with the operation of particular production systems (e.g., Holyoak et al., 1989).

In this paper we investigate how the abstract characterization of mental models postulated by Holland et al. (1986) can be tested without simultaneously being confounded with a particular cognitive architecture. Students of mental models are primarily interested in the behavioral effects of mental models as knowledge representations, rather than in grand theories of cognitive architecture (e.g., Payne, 1992). The focus of this investigation will be those mental models that govern the interaction with relatively complex devices (e.g., such as studied in the field of human-computer interaction). These models are often referred to as "user models" (Norman, 1983, 1986).

In the first part of the paper a formal definition of mental models similar to that of Holland et al. (1986) is presented. It will be shown that the addition of a few reasonable constraints to the formalism proposed by Holland et al. (1986) results in a class of transition functions with well-known properties -- the general class of finite-state machines (Davis, Sigal, & Weyuker, 1994; Denning, Dennis, & Qualitz, 1978; Kolman & Busby, 1987; Minsky, 1967). Modeling mental models as finite-state machines has several advantages. First, the finite-state formalism provides systematic ways to achieve a minimal

[1] Or rather, a quasi-homomorphism (i.e., q-morphism), to allow for the fact that most mental representations in real life will be imperfect.

form of a machine that accounts for a given set of input-output mappings (Denning et al.. 1978). Thus securing a fully parsimonious account of any particular mental model (unlike the production rules system approach). Second, because the theory of finite-state machines is intertwined with the theory of abstract grammars, the likelihood of a given sequence of input actions performed by a user (conceived of as an input alphabet) can be estimated under the hypothesis of a specific finite-state machine. In this paper we present a method to empirically test mental models conceptualized as finite-state machines.

Part two of the paper presents detailed calculation methods to demonstrate certain insights that may be gained from this approach. Data from a spatial reasoning experiment will be used. In this experiment second grade and third grade children were asked to move an object from an initial state to a goal state on the basis of a schematic diagram of a spatial structure. There were two versions of the task. One version resembles a city plan with recognizable markers (i.e., shops and churches) and a recognizable object (i.e., a model of a bus). The other version had abstract markers (such as triangles and squares), and a more abstract object to be moved (i.e., a pawn). In this paper only data from the version employing the bus model is discussed.

PART I

A formal definition of mental models

The Holland et al. (1986) conception of mental models consists of two elements. First, a categorization function that categorizes environmental states and, second, a transition function defined on these categories of environmental states which mimics the state changes in the environment to be predicted. For ease of exposition the elements are discussed in their reverse order.

The transition function

A finite-state function

Following Holland et al. (1986) we characterize a mental model **M** as a transition function (or a next-state function) δ defined over a set of states Q and a set of inputs I.[2]

$$\delta\colon Q \times I \rightarrow Q \qquad (1)$$

The domain of the transition function δ is the set of all state-input pairs, and its range is a subset of states. The transition function in Holland et al. (1986) and Holyoak (1985) does not provide a model in the strictest sense. Rather, it provides a general metaphor for

[2] Actually, we follow a notation more similar to the notation presented in Holyoak (1985), which is clearer than Holland et al.'s (1986) notation.

theorizing about changes in an environment. It applies equally well equally to a very 'local environment' (e.g., a chessboard, pieces and players) as to the whole of the universe (Holland et al. 1986, p. 30). The input or action term of the transition function can represent (discrete) actions of human participants, as well as (continuous) autonomous effects such as those caused by the working of the laws of nature (e.g., all fast-moving objects slow down (Holland et al. 1986, p. 31)).

In this paper a special class of transition functions is investigated. These functions may be derived by constraining the general model further and by then assuming the following properties of **M**, Q, I, and δ

- The behavior of **M** is defined only at the moments $t = 0, 1, 2, \ldots$
- The states q_t are chosen from a finite set of states Q (i.e., the set of model states).
- The input symbols s_t are chosen from a finite alphabet I (i.e., the input alphabet).
- The output symbols o_t are chosen from a finite alphabet O (i.e., the output alphabet).
- The behavior of **M** is uniquely determined by the sequence of input symbols that are presented.

Now, **M** is constrained to describe discrete phenomena. The occurrence of discrete phenomena may be represented as a sequence of events, in which any event is a 'next-to' event, and some events may be 'initial', whilst others may be 'terminal' events. It is convenient to think of system **M** as a machine that can accept input, possibly produce output, and have some sort of internal memory that can keep track of certain information about these previous inputs. It is assumed that **M**'s memory for past events is of a fixed, finite size. As a consequence, **M** can only distinguish between some finite number of classes of possible event sequences. These classes will be called the states of the machine. Two additional assumptions determine the finitude of **M**: the input and output parameters of system **M** can only assume a finite number of distinct values. By convention the sets of values which these parameters can assume are called the input alphabet (I) and output alphabet (O), respectively. Each element in I and in O is called a symbol. Further, it is assumed that **M** works at discrete intervals of time. At each time **M** is in one of these states, say q_t. The state q_{t+1} at the next time interval only depends on the previous state q_t and the input s_t given at time t.

Together, these properties characterize the triple $\mathbf{M} = (Q, I, \delta)$ as a finite automaton (Davis et al, 1994; Denning et al, 1978; Kolman & Busby, 1987; Minsky, 1967). The transition function δ, which maps $Q \times I$ on Q, consists of a finite set of productions. Formally, productions that map the current state and input signal onto the next state are designated $\delta\ (q_t\ ,\ s_t) \rightarrow q_{t+1}$. The states $q \in Q$ refer to *states* of model **M**. In order to produce observable output, **M** must encompass a function to relate the various model states, or state transitions, to its output.

Let λ be an output function λ, which maps $Q \times I$ on O. The output function λ consists of a finite set of productions $\lambda\ (q_t, s_t) \rightarrow o_{t+1}$, which map the current state and input signal onto the next output. The output function λ qualifies **M** as a so-called Mealy machine and is usually referred to as a sixtuple, $\mathbf{M} = (Q, I, O, \delta, \lambda, q_I)$, where Q, I, O, δ, and λ are as defined previously, and q_I designates the initial state of **M** (i.e., $q_I \in Q$). A

Mealy machine is a special type of automaton where the output is associated with the transition between states. Thus, $\lambda\ (q_t, s_t) \rightarrow o_{t+1}$ gives an output which shows the transition from state q_t on input s_t. The output of a Mealy machine **M** in response to a sequence of input symbols $s_1, s_2, ... s_n$ is $\lambda\ (q_1, s_1), \lambda\ (q_2, s_2), ..., \lambda\ (q_n, s_n)$, where $o_2, o_3, ..., o_{n+1}$ is the sequence of outputs produced in parallel to the state sequence $q_2, q_3, ..., q_{n+1}$.[3]

To further characterize finite-state machines, note that a special case of a finite-state machine arises when

$$\delta\ (q, s) = \delta\ (s) \qquad (2)$$

Such a machine is called a trivial machine. A trivial machine determines a fixed function between input and output. Finite-state machines belong to the group of computational systems that can compute various functions. Thus, $\delta\ (q, s) \neq \delta\ (s)$, with q providing a specification of the function to be computed. It is preferable to consider q a system parameter, rather then a second argument of the function. The specific nature of the state set of a finite-state machine will be further explicated in the next section.

Deterministic versus Probabilistic Automata

We will now introduce a distinction that has significant implications for the empirical test of mental representations modeled as finite-state machines. These implications will be elaborated upon in a later section.

The finite automaton discussed so far is strictly deterministic in its actions: at each moment the next state and the output symbol are uniquely determined by the present state and input symbol. A deterministic finite automaton (DFA) can be considered as a special case of a probabilistic finite automaton (PFA), where S_I and $\delta\ (q_t, s_t)$ consists of one state, and $\lambda\ (q_t, s_t)$ consists of one output symbol. In a PFA the productions of the DFA state transition function $\delta\ (q_t, s_t) \rightarrow q_{t+1}$, are replaced by productions of the form $\delta\ (q_t, s_t) \rightarrow \{q_{t+1,1}\ ..., q_{t+1,n}\}$. Thus, given a present state q_i and an input symbol s_j various states $q \in Q$ can be the next state. To each of the n possible transitions a probability $P_i\ (q_i, s_j)$ is assigned. Associated with each state q_i and input symbol s_j is a stochastic (column) vector $f\,(q_i, s_j)$ of transition probabilities (i.e., an n-dimensional vector with non-negative components, the sum of which equals 1).

In a similar way the productions of the output function $\lambda\ (q_t, s_t) \rightarrow o_{t+1}$ are written as $\lambda\ (q_t, s_t) \rightarrow \{o_{t+1,1}, ..., o_{t+1,\ m}\}$. Thus given a present state q_i and input symbol s_j

[3] An alternative way to assign an output is by a function $\lambda' = Q \rightarrow O$. An automaton with a λ'-type output function assigns an output symbol to each state. This type of machine is known in the literature as a Moore machine. A prototypical example of a Moore machine, or recognition machine, is a parity checker, that is, a machine that indicates by its output whether the parity of a sequence of input symbols in the binary alphabet {0, 1} is odd or even. Although this type of automaton can be shown to be formally equivalent to a Mealy machine (e.g., Denning et al, 1978) in this paper we will only deal with Mealy machines.

various output symbols $o \in O$ are possible. Each of the m possible outputs is assigned a probability. The set of output probabilities is also a stochastic (row) vector φ (q_i, s_j). Note that a stochastic (row or column) vector is a coordinate vector iff one of its components equals 1. Thus, a deterministic Mealy machine represents the special case of a probabilistic automaton, where all of the vectors f and φ, as well as S_i (the initial state distribution) are coordinate vectors.

Probabilistic finite automata may be useful to model mental representations of interpersonal interactions or other processes with a stochastic component, such as running a business, or planning interventions in a macro economic system. In this paper we investigate the assessment of mental models of devices that are constructed such that their behavior is (or may be expected to be) perfectly predictable. For example, airplanes, cars, ATMs, application programs and other high-tech products of our culture. Throughout the paper the term finite-state machines will be used to refer to *deterministic* finite-state machines, unless explicitly stated otherwise.

Engineers and naive users

In summary, a finite-state machine can be viewed as a general mathematical model of an interactive system defined by a finite number of states. When it is presented with an input from an action performed by a user, then, as a function of this input and its current internal state, it will respond by moving to another of its internal states, and produce an output. Finite-state machines have been used to model devices ranging in complexity from simple 'flip-flops', such as light switches, to entire computers. Most engineering and scientific investigations use finite-state machine models to characterize a particular system in order to achieve effective control over and predictability of its behavior. This enterprise is not principally different from the attempts of naive users to construct mental models in order to gain control over a device and utilize it effectively.

The categorization function

At any point in time, a person interacting with a computer or another device will be able to observe a set of n different physical states, in which such a system can be. These states characterize the system's dynamic behavior and we will refer to these states as system states. It is reasonable to expect that a person attempts to construct a simplified model by aggregating the system states into useful categories and ignoring details that are irrelevant to the purpose of the model (Holland et al. 1986).

Some m dimensions can describe each of the system states. Each dimension k has p_k values. A single dimension, or a combination of individual dimensions, can form a basis to partition the set of system states E. If the person is able to detect such a dimension (or subset of dimensions) and considers it relevant for his or her purposes, then the person will use that dimension (or subset of dimensions) to aggregate the set of system states E into the set of model states Q. Formally, this aggregation process can be described as a mapping of system states to model states. In the literature this mapping has been referred to as a categorization function P (Holland et al., 1986), or an instantiation function *IF* (Pylyshyn, 1984). We will follow the Holland et al. terminology. Subsequently, some aspects of this function will be discussed.

Functionally equivalent system states

Any function necessarily determines an equivalence relation on its domain, which partitions the domain into equivalence classes (see APPENDIX 1, section A). The significance of the partitioning of the set of system states E into equivalence classes or categories is based on the intensional definition of its categories. For each system state $e \in [e]$, where $[e]$ denotes a partition of E, an identical function $f\colon I \rightarrow O$ is defined, where I denotes the finite set of input symbols and O the finite set of output symbols. Each category $[e] \in E$ has a unique function $f\colon I \rightarrow O$. Each category thus defines an equivalence class of system states that are indistinguishable from a functional point of view. In other words, a person who wants to utilize a device effectively has to partition the set of system states into mutually exclusive categories. The partition should maximize what the person can predict about the system's behavior in response to his or her actions.

For illustrative purposes, let us consider the well-known children's programming environment called the LOGO 'Turtle World' (Papert, 1980; Abelson & diSessa, 1980). This microworld has been especially designed to help children in developing useful ways of thinking about computing. The LOGO 'Turtle World' provides a graphical interface in which children can explore the effects of simple programming commands on the behavior of a screen object, called the 'turtle'. A set of very simple commands such as FORWARD and BACK (move the turtle), and LEFT and RIGHT (turn the turtle) is available to effect the state of the turtle on the computer screen. Each series of concrete actions performed by the turtle create a graphical trace on the computer screen. In this way, LOGO programming permits the turtle to draw regular polygons and other geometrical shapes. Each system state of the turtle can be described as a set of physical dimensions. For example, a particular location (i.e., X- and Y-coordinates), orientation, and color of the turtle. Only one of these dimensions is causally relevant for any of the state changes that may occur in response to an input action performed by the student. Unless the student has discovered this dimension, he or she will be confused by what seems like the unpredictable behavior of the turtle. For example, in the LOGO 'Turtle World' the impact of the commands given to the turtle is conditional upon its orientation on the screen. In the turtle's primary position, that is, facing up, the command FORWARD 100 results in a movement across the Y-axis. However, when the turtle is facing east, and it is given the command FORWARD 100, the turtle moves across the X-axis. This difference in the turtle's behavior has been shown to be a substantial source of confusion in young children (Fay & Mayer, 1987; Cohen, 1987).

For each orientation of the turtle a different mapping is defined of input symbols (i.e., the programming commands FORWARD, BACK, RIGHT and LEFT) to output symbols (changes in location or orientation of the turtle), irrespective of its location and/or color. Thus, each orientation of the turtle defines an equivalent class of system states that may differ in location and/or color, but share the same function relating programming commands to the turtle's behavior.

In summary, the basic claim is that the human user of an interactive system will categorize system states such that the prediction of the effect of an input action is maximally accurate. Knowledge of the current state of the machine, reduces the uncertainty about the effect of a possible action.

Encoding and decoding

To further explore the relationship between mental model and the system being modeled, we will now investigate some characteristics of P. Let $e \in E$ and $q \in Q$. The function P: $E \rightarrow Q$ is then defined by

$$P(e) = q \qquad (3)$$

Let p: $E \rightarrow E/R$ (see APPENDIX, section A) and m: $E/R \rightarrow Q$ be functions. Let $[e] \in E/R$. The function p: $E \rightarrow E/R$ is then defined by

$$p(e) = [e] \qquad (4)$$

and the function m: $E/R \rightarrow Q$ is then defined by

$$m([e]) = q \qquad (5)$$

The function P can be written as a composition of p and m: $P(e) = (p \circ m)(e) = q$, or $P(e) = m\,[p\,(e)] = q$. Since the function m: $E/R \rightarrow Q$ is a one-to-one function (see APPENDIX 1, section B), the function m is invertible, that is, its inverse m^{-1} is also a function. The function $p : E \rightarrow E/R$ is not one-to-one. Thus, p is not invertible. We will call the functions P, p and m encoding functions, because they account for the aggregation of the set of system states into the set of model states. Their inverse functions, if defined, will be referred to as decoding functions. Note, there is only one decoding function: m^{-1}.

Now consider $\mathbf{M} = \{Q, I, \delta)$, a finite-state machine with state set $Q = \{q_1, ..., q_n\}$, input set I, and state transition functions $\delta = \{\delta_x \mid x \in I\}$. For any $q, q' \in Q$ and s $\in I$, we write $\delta_s(q) = q'$, that is, input s takes state q into state q'. The structure $\mathbf{M}$ is assumed to be a representation of some behavior of a system in the environment. Therefore, the representation law (e.g., Newell, 1990) applies, which in its general form states that the essence of a representation is to allow to go from one external situation to another by a different path, that is, by manipulation of a internal representation, rather than by actually effecting the initial external situation itself. In symbols:

$$\text{decode}\,[\text{encode}\,(T)(\text{encode}\,(e))] = e' \qquad (6)$$

where e and e′ are external situations and T is the external transformation and [encode (T)] maps T onto symbol $s \in I$, [encode (e)] maps e onto $q \in Q$, and [decode (q')] maps q' onto e' (after Newell, 1990, p. 59). Since P is a many-to-one function the predictive power of structure **M** is somewhat limited by the specificity of the partition. Note that P has not an inverse that is also a function. Thus, we write P as a composition of p and m. At least m has an inverse function (i.e., the decoding function m^{-1}). Thus,

$$m^{-1}\{\delta_s[m(p(e))]\} = [e'] \qquad (7)$$

where δ_s denotes the state transition function which takes the symbol $s \in I$ as input. Equation (7) readily demonstrates that the predictive power of a model **M** is limited to the prediction of categories of system states (i.e., the finite state machine can predict an equivalent class $[e']$ instead of a single state e'). This can be a somewhat global prediction (Holyoak, 1985). Only in the case of a maximally specific partition, that is, a partition with only one system state per category, this would lead to the exact prediction of the next system state. It seems plausible to assume that this limitation of the representational structure requires the (human) cognitive system first to generate the model state corresponding with the next system state (i.e., goal state or sub goal state), and then to chose an input symbol to transfer the current model state into the next model state (compare the operator-difference table of Newell & Simon's (1972) General Problem Solver (see also Charniak & McDermott, 1985)).

Empirical consequences

To derive empirical consequences from this abstract characterization of the representational structure implied by Holland et al.'s (1986) notion of a mental model, two auxiliary assumptions have to be specified. First, participants are able to verbalize the symbols $s \in I$ of **M**, which stand for the internal representations of the input actions performed by a user. Second, the output symbols $o \in O$ of **M**, representing the system's actions, may be made observable. Then, the productions

$$\lambda(q_t, s_t) \rightarrow o_{t+1} \qquad (8)$$

contain two sets of observable entities, viz., the input symbols $s \in I$, and the output symbols $o \in O$, as well as one set of non-directly observable entities (i.e., the internal states $q \in Q$). However, by substituting (3) in (8), it is possible to obtain

$$\lambda(P(e_t), s_t) \rightarrow o_{t+1} \qquad (9)$$

where e_t refers to the outcome e that realizes an event q at moment t. In Equation (9) all terms represent observable events.

Equation (9) implies that all factors which connect s_t and o_{t+1} are explicitly included in q_t. It is appropriate to map this conjecture into a probability framework by interpreting s_t and o_{t+1} as events in two finite sample spaces I and O, respectively. Further, the set of states E can be considered a sample space partitioned by P into a number of equivalence classes $[e]$, each of which corresponds with a particular model state. Thus, an outcome $e \in [e]$ is said to realize event q. In the next section we present a method to test a hypothesis involving the partitioning of a set of environment states into equivalence classes that may be identified as states of the model.

Model predictions

Consider a $I \times J \times K$ cross classification of states q_i, $i = 1, \ldots, n_i$, input symbols $s_j, j = 1, \ldots, n_j$, and output symbols o_k, $k = 1, \ldots, n_k$. Assuming the events s_j and o_k are directly observable, the event q_i is realized by an outcome $e \in [e_i]$, based on a partitioning P_k of the set of states E. The relationship among s_j and o_k is supposed to be mediated by the state variable q_i, such that $\lambda(q_i, s_j) \rightarrow o_k$. That is, for every combination of a state and an input signal, one and only one output signal may be observed, although a given output signal can be associated with more than one combination of state and input signal.

Let $P\ (q_i)$ represent the unconditional probability of being in state q_i. That is, given the set of all conceivable observations for a participant, $P\ (q_i)$ represents the probability for a randomly selected observation of the system while being in state q_i. Similar unconditional probabilities can be defined for s_j and o_k. Write $P\ (q_i, s_j)$ for the probability of the joint event of being in state q_i and receiving input signal s_j, and $P\ (q_i, s_j, o_k)$ for the probability of the joint event of being in state q_i, receiving input signal s_j, and producing an output signal o_k. Each cell in the $I \times J \times K$ cross-classification represents the frequency of occurrence of a joint event $(q_i,\ s_j, o_k)$, which can be easily converted into a probability. In general, the following equality holds for this probability: $P\ (q_i, s_j, o_k) = P\ (o_k \mid s_j, q_i)\ P\ (q_i, s_j)$. Similarly, $P\ (q_i, s_j) = P\ (q_i \mid s_j)\ P\ (s_j) = P(s_j \mid q_i)\ P(q_i)$. Thus we get the basic equality:

$$P\ (q_i, s_j, o_k) = P\ (o_k \mid s_j, q_i)\ P\ (q_i \mid s_j)\ P\ (s_j) \qquad (10)$$

Therefore, a finite state model defines a pattern of expected frequencies in the $I \times J \times K$ cross-classification table, which can be easily calculated from the marginals. As for probabilistic finite automata this poses no specific problems for model tests using a chi-square based statistic (i.e., χ^2, likelihood ratio). Note, however, that the definition of the DFA implies $P\ (o_k \mid s_j, q_i) = 0$ or 1. The dependency of o_k on q_i and s_j can be incorporated in the notation by writing $i'j'$ in stead of k, and writing $P\ (o_{i'j'} \mid s_j, q_i) = 1$ if $i' = i$ and $j' = j$, and zero otherwise. From this it follows that $P\ (q_i, s_j, o_k) = P\ (q_i \mid s_j)\ P\ (s_j)$, or equals zero.

Statistics are not required in order to test the deterministic model. When a purely deterministic model predicts one (or zero) for a particular cell in a cross-classification table, then every deviation from this predicted probability implies that the model should be rejected. Thus, visual inspection of the table(s) is sufficient to assess whether the hypothesis is confirmed or should be rejected. However, if model deviations are observed, it may be useful to characterize the deviations from the model by measures that reflect how well predictions from the model conform to the observations. The finite-state machine describes essentially a functional, asymmetric relationship between a combination of input signal and state on the one hand and an output signal on the other. Thus, the measure of fit should allow for a characterization of such an asymmetric relationship. One such measure is the lambda statistic, proposed by Goodman and Kruskal (1979). This statistic quantifies the reduction in uncertainty in the prediction of one category from the other as a function of their association. A disadvantage of this measure is that it only takes into account the largest probability in a row (or column). Some of the other measures are based on the concept of entropy as developed within Information Theory (see e.g., Krippendorff, 1986). These measures are more closely related to notions familiar from the multivariate analysis of numerical variables, such as the variance and proportion of explained variance for a dependent variable, and (semi)partial correlations or covariances. These measures are now defined.

An information theoretic evaluation of model coherence

Entropy. Let the number of categories of a variable X be denoted by n_X and let the relative frequency in category i of X be p_i ($0 \leq p_i \leq 1$). The entropy $H(X)$ of the variable is then defined as

$$H(X) = -\Sigma_i \, p_i \log_2 p_i \qquad (11)$$

Entropy can be interpreted as a measure of the variability of a numerical or nonnumeric distribution. It shares with the variance of a distribution of a numerical variable the property that it attains its maximum value if, for all i, p_i is equal to $1/n$ (with n the number of categories of the variable). It also attains its minimum value if p_i is equal to one for one i and consequently zero for other categories (i.e., if the variable has only one category). The maximum and minimum values are equal to $\log_2 n$ and 0, respectively ($0 \log_2 0$ is defined as zero).

The logarithm in Equation (11) is taken to the base two, which leads to an interesting interpretation of $H(X)$. It expresses the variability in X in terms of the basic unit of measurement within Information Theory: the average number of binary decisions, or bits of information, necessary to make a classification within a system of categories (Attneave, 1959; Krippendorf, 1986; Shannon & Weaver, 1949). For example, consider a user of a particular device (such as an application program) who chooses an input action s from a set of possible input actions I, then, $H(I)$ expresses the average amount of uncertainty under which the user operates. Note that actions that are logically possible, but never actually chosen, do not enter the measure (i.e., actions s such that $s \in I$, but with $p_s = 0$). To this purpose the convention $0 \log_2 0 = 0$ is adopted here. The entropy measure $H(X)$ can be extended to the joint distribution of two or more variables as $H(XYZ...) = - \Sigma_i \Sigma_j \Sigma_k ... p_{ijk..} \log_2 p_{ijk...}$. Analysis of the joint distributions of variables is at the heart of Information Theory.

Entropy can be defined not only for marginal and joint distributions, but also for conditional distributions. For a given value x of X, the conditional entropy of Y, $H(Y \mid x)$ is defined as $- \Sigma_j (p_{ij} \mid p_i) \log_2 (p_{ij} \mid p_i)$. Note that this definition is completely analogous to the entropy of the marginal distribution. The only difference is that the marginal probabilities p_i are replaced by the conditional probabilities $p_{ij} \mid p_i$. The average conditional entropy of Y, given X, is a weighted sum of the conditional entropies for given values of X, with weights equal to p_i, the proportion of observations for the particular value of X on which the conditioning has occurred. Thus, $H(Y \mid X) = - \Sigma_i \, p_i \, \Sigma_j \, (p_{ij} \mid p_i) \log_2 (p_{ij} \mid p_i) = - \Sigma_i \Sigma_j \, p_{ij} \log_2 (p_{ij} \mid p_i)$.

Information Transmission. The predictability of output symbols $o \in O$ from the set of input symbols $s \in I$ is in Information Theory defined as information transmission, $T(I, O)$. The amount of information transmission can be expressed in several different ways that are mathematically equivalent (Krippendorff, 1986). We will discuss two conceptions of information transmission. The first conception defines information transmission, $T(I, O)$, as the difference between the maximum entropy and the observed entropy. Hence:

$$T(I, O) = H(I) + H(O) - H(IO) \qquad (12)$$

A participant endowed with a perfectly valid mental model of a device that can be characterized as a DFA (i.e., with a mental model that is a DFA) would render an observed entropy that is equal to zero (i.e., $H(IO) = 0$), that is, per row (or column, whatever is appropriate), $p_i = 1$ for one cell and zero for all others, in which case both 1 $\log_2 1 = 0$ and by convention $0 \log_2 0 = 0$. In the case of a total lack of predictability the observed entropy would equal its maximum value, $H(IO) = H(I) + H(O)$. That is, the observed entropy equals the sum of the marginal entropies $H(I)$ and $H(O)$, respectively.

Notice that participants can not only differ in the validity of the model they are using, that is, the extent to which the observed entropy approaches zero, but also in the way they use the model. Participants can differ in preference for certain input actions by which they act upon the device. For example, recall the previously discussed LOGO 'Turtle World'. Let us assume that the turtle is facing "north", then a vertical line with a length of 100 units can be drawn southward in several ways: (a) by a single command (i.e., BACK 100), and (b) by a series of commands (i.e., [RIGHT 180 FORWARD 100] or [LEFT 180 FORWARD 100]). Research shows that children have a preference for the latter way of drawing the line (Campbell, Fein, Scholnick, Schwarts, & Frank, 1986; Fay & Mayer, 1987). As a result the observed variety of input symbols, $H(I)$, will be lower than can be expected on the basis of a uniform distribution of probabilities, since the use of one particular symbol (i.c., BACK) is systematically avoided. Notice also that reduction in observed variety in I does not necessarily imply a similar reduction in O. This may even occur in the case of a perfectly deterministic mental model (as the example shows).

This conceptualization of predictability as a (general) reduction of uncertainty is clear and straightforward as a first approach, but it does not provide precise information about the contribution of each individual input symbol to the observed entropy. The second conception of information transmission is based on the notion that knowing about I may reduce the uncertainty about O. Thus, $H(O \mid I) \leq H(O)$ with information transmission defined as

$$T(I, O) = H(O) - H(O \mid I) \qquad (13)$$

where $H(O \mid I)$ denotes the entropy in O given I. In information theoretical terms $H(O \mid I)$ represents the noise produced by the input symbols. $H(O \mid I)$ gives the average entropy of O over each observed input symbol, that is, a weighted sum of entropies associated with each row of the matrix (i.e., $H\ (O \mid i)$). Thus, $H(O \mid I) = \Sigma_i\, p(i).H(O \mid i))$, where $p(i)$ denotes the relative frequency of symbol i (or row i). The information transmission statistic can be straightforwardly extended to three-way classifications. For example, $T\ (I, O \mid Q) = H\ (O \mid \mathrm{Q}) - H\ (O \mid IQ)$, where the dependence of O on the joint occurrence of I and Q is modeled (see Part II).

Proportional Reduction in Uncertainty (*PRU*). The association between categories in a two-way classification is conventionally expressed as

$$\eta_{\mathrm{H}}\ (O,I) = T(I, O)\ /\ H(O) = [H(O) - H(O \mid I)]\ /\ H(O) \quad (14)$$

In a three-way classification the additional factor leads to the following: $\eta_{\mathrm{H}}\ (O \mid IQ) = T\ (IQ, O)\ /\ H\ (O) = [H\ (O) - H\ (O \mid IQ)]\ /\ H\ (O)$. The interpretation of $\eta_{\mathrm{H}}\ (O \mid IQ)$ is analogous to the partial correlation coefficient in continuous-variable statistics.

The observed probabilities in a cross-classification do not in general coincide with their population values. Hence measures of transmitted information are partly determined by sampling fluctuations (i.e., they are generally biased). For a two-dimensional table the observed measure may be larger than zero whereas it's true population value is not. The measure of transmitted information for a two-dimensional table is a simple function of the likelihood-ratio statistic G^2 for the chi-square test of independence of the variables that form the cross-classification (i.e., $T\ (I, O) = \mathrm{G}^2\ /\ 2n$, where n is the total number of observations). Thus, the G^2 statistic can be used to test whether $T\ (I,\ O)$ is different from zero (alternatively, the asymptotically equivalent Pearson χ^2 statistic can be used). If G^2 is not statistically significant, the variables are assumed independent and $T\ (I, O)$ is assumed to be zero. The G^2 statistic is often used to compare the fit of loglinear models that are fitted to a cross-classification by maximum likelihood. For a two-dimensional table, G^2 compares the fit of the independence model versus a model of (unrestricted) dependence of the variables. The latter model fits the data perfectly, because the observed and predicted cell frequencies in the table are identical (it is a so-called saturated model). If G^2 is not significant, the variables are presumably independent. In the independence model the cell probabilities are fitted as the product of the marginal probabilities (i.e., $p\ (\mathrm{s}_i,\ o_k) = p\ (s_i) \times p\ (o_k)$). If the value of $p\ (\mathrm{s}_i,\ o_k)$ under independence is substituted in $T\ (I,\ O)$ the latter becomes zero (under the dependence model, the value of $T\ (I, \mathrm{O})$ is the one computed from the observed probabilities). Thus, setting $T\ (I, O)$ to zero if G^2 is not significant can also be interpreted as using test statistics to fit an appropriate model (e.g., a loglinear model is first fitted to the table) and subsequently computing measures of transmitted information from the probabilities generated by the model. For a three dimensional table, where the third dimension is formed by the states, the important models to consider are: (1) a model of complete independence, (2) a model of dependence of I and O, where the dependence is the same for the different states, and (3) a model where the dependence of I and O is different for different states. In practice, model fitting and testing may be complicated by observed frequencies of zero in the table. If zero frequencies occur, it may not be possible to fit some models or some estimated frequencies may become zero, in which case the degrees of freedom need to be adjusted. A second problem that in practice can occur is that the generated frequencies are too small for the chi-squared approximation to be valid.

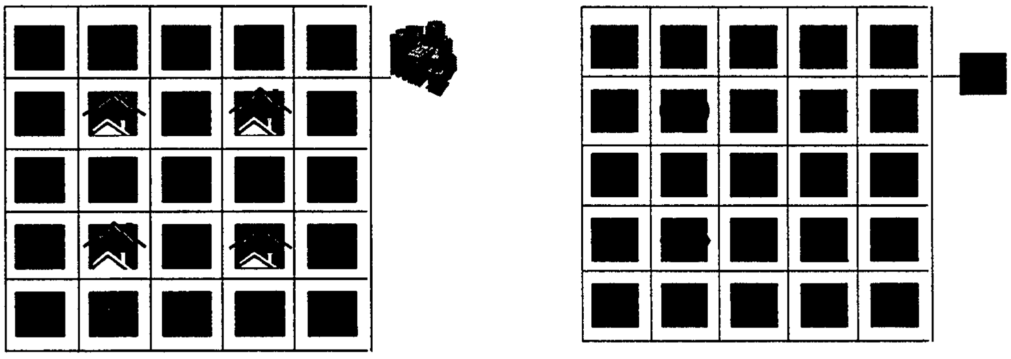
PART II

The MAP test: A spatial reasoning task

The MAP test is a spatial reasoning task which requires children to move an object from an initial state (*IS*) to a goal state (*GS*) on a schematic diagram of a spatial structure. There are two versions of the task. One version suggests a city plan with recognizable markers (i.e., shops and churches) and a recognizable object (i.e., a bus). The other version has abstract markers (such as triangles and squares), and a more abstract object to be moved (i.e., a pawn) (see Figure 1). The hypothesis to be tested is that the different experimental conditions evoke basically the same input alphabets (operators: forward, back, right, and left), but otherwise different finite-state machines (i.e., mental models). This paper does not include a complete report of the experimental results. For this the interested reader is referred to Ippel and Beem (1997).

The MAP test consists of 32 items; 8 items are introductory items, 12 items in which *IS* and *GS* have identical *Y*-coordinates, but an interrupted path from X_{IS} to X_{GS}, and 12 items in which both the *X*- and *Y*-coordinates of *IS* and *GS* differ. The *Y* differences between *IS* and *GS* are systematically manipulated with *y*'s of +5, +3, -3, and -5, where y = $Y_{GS} - Y_{IS}$. For 12 items *GS* is located at the left side of the diagram and in the remaining items *GS* is located at the right side of the diagram. The first 8 introductory items involve easy problems, that is, these items have uninterrupted paths between *IS* and *GS*. These items were not used in our analysis.

Participants. Participants were 48 second-grade (mean age 7.75 years, 23 female) and 52 third-grade students (mean age 8.75 years, 22 female) from elementary schools near Leiden (The Netherlands).



(a) "bus" condition

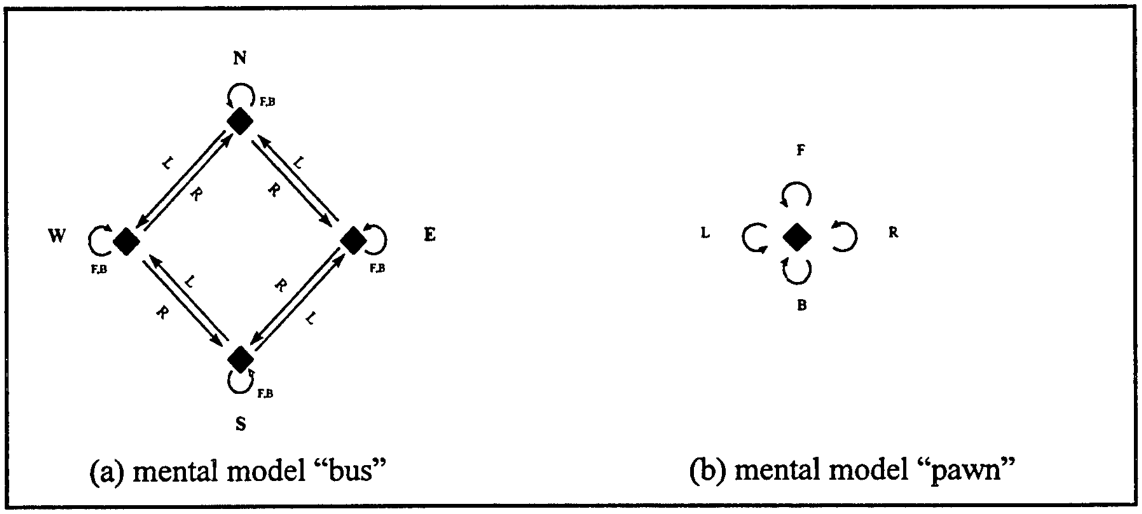
(b) "pawn" condition

Figure 1.
Different diagram representing an identical spatial structure in the two experimetal conditions.

Procedure. Students were randomly assigned to one of the two object conditions, grade and gender were equally distributed over the conditions. Three experimenters were involved in administering the tests. The tests were individually administered in separate offices in a quiet part of the school. Test time varied between 30 and 45 minutes. The experimenter instructed the students: (1) to place the object (either a pawn or a bus) in its initial position; (2) to move the object from its initial state to the goal state while following the shortest possible route; (3) to talk aloud while moving the object. Students were asked to describe the moves of the objects such that (a) in the pawn condition, the experimenter could perform the same move, even though he or she could not see the actual move being performed, and (b) in the bus condition, an imaginary bus driver could follow up the instructions and drive the bus to its goal. The experimenter scored the actual physical state of the object after a command had been executed. Verbal protocols were audio taped and later typed out. Two raters independently scored each protocol. The raters scored each statement according to a predefined input category system. Divergent scorings were discussed until agreement was reached in a separate session and scored accordingly.

The mental model hypothesis

For each of the two conditions a detailed hypothesis concerning the mental model to be evoked by the experimental conditions can be formulated. Figure 2 shows the state transition diagrams representing the mental models for the bus and the pawn. The blocks represent the states, the arrows represent the state transitions, and the letters next to the arrows denote the input symbols, viz., *L*, *R*, *F* and *B* denote LEFT, RIGHT, FORWARD and BACK, respectively. The letters *N*, *E*, *S* and *W*, in the state transition diagram representing the mental model of a bus, refer to names of the four states q_1, ..., q_4 (i.e., *north*, *east*, *south* and *west*), respectively.



(a) mental model "bus"

(b) mental model "pawn"

Figure 2.
The state transition functions represented as a state transition diagrams.

Bus model. The bus starts out in state q_1, that is, facing 'north'. In this state the next output function comprises four productions which map an element from the set of input symbols $I = \{F, B, L, R\}$ onto the set of output symbols $O = \{y\text{-}, y\text{+}, x\text{-}, x\text{+}, r\text{-}, r\text{+}\}$. The symbols *y-*, *y+*, *x-*, and *x+* refer to movements along the *Y*- and *X*-axis and the symbols *r+* and *r-* refer to a clockwise and a counter clockwise rotation of 90 degrees, respectively. These productions are $\lambda_{(F)} \rightarrow y\text{-}$, $\lambda_{(B)} \rightarrow y\text{+}$, $\lambda_{(L)} \rightarrow r\text{-}$, and $\lambda_{(R)} \rightarrow r\text{+}$. Figure 2 indicates that the choice of either *F* or *B* effects the system's physical state, that is, *F* and *B* do effect the location of the bus. However, these input symbols would not change the state of the mental model. Thus, the same set of productions holds for the next input. If one of the input symbols *L* and *R* is chosen, not only the system's physical state will be affected, but the mental model of the bus also will enter a new state (either q_4 or q_2). That is, the bus turns either *west* or *east* and enters either q_4 or q_2, respectively. For example, let us assume that the bus is facing *north* and that the student instructs the imaginary bus driver to take a turn to the right, such that the bus now faces *east*. At the same time, a new next output function holds with the productions: $\lambda_{(F)} \rightarrow x\text{+}$, $\lambda_{(B)} \rightarrow x\text{-}$, $\lambda_{(L)} \rightarrow r\text{-}$, and $\lambda_{(R)} \rightarrow r\text{+}$. Table 1 summarizes the four different next output functions that define the four states of the mental model of the bus.

Table 1. Next output functions of each of the four states of the mental model for the bus.

λ	q_1	q_2	q_3	q_4
F	y–	x+	y+	x–
B	y+	x–	y–	x+
L	r–	r–	r–	r–
R	r+	r+	r+	r+

Pawn model. Objects such as a ball, or a pawn, do not have intrinsic perceptual features such as a front side, or backside, and therefore, no left or right side. Sentences such as "move the pawn forward" and "turn the pawn to the right" cannot be meaningfully related to spatial features of the object itself, and therefore, most likely, will be interpreted in relation to the direction the student is facing. As the student's position does not change during the test session the same function is expected to map the set of input symbols $I = \{F, B, L, R\}$ onto the set of output symbols $O = \{ y\text{-}, y\text{+}, x\text{-}, x\text{+}\}$, that is, $\lambda_{(F)} \rightarrow y\text{-}$, $\lambda_{(B)} \rightarrow y\text{+}$, $\lambda_{(L)} \rightarrow r\text{-}$, and $\lambda_{(R)} \rightarrow r\text{+}$. Therefore, the mental model of a pawn is a one state finite-state machine, an example of a so-called trivial machine. In this paper we will not be concerned with the pawn condition (see Ippel and Beem, 1997).

Empirical evaluation

The first aspect of a mental model as a finite-state machine to be considered concerns the input alphabet (I) and output alphabet (O) that the participant uses to describe the interaction with the artifact under investigation. These alphabets may differ from the alphabets that define the proposed mental model. The latter alphabets will be referred to as I^* and O^*, respectively. Participants can deviate from model specifications in two distinct ways. First, the participant uses less input and/or output symbols than the hypothetical model assumes, that is, $I \subseteq I^*$ and/or $O \subseteq O^*$. This poses no analytical problem, because entropy (i.e., $H(I)$ and $H(O)$) is a measure of observational variety and unobserved possibilities do not enter into the measure (Krippendorff, 1986). Of course, this in turn suggests that the participant does not utilize the artifact's possibilities to its fullest extent.

More problematic is the possible event that I and I^* and/or O and O^* only partially overlap. That is, the participant uses input symbols and/or output symbols, which are not member of the finite sets I^* and O^* that define the proposed finite-state machine. Partial overlap represents a more serious misconception of the functioning of the artifact. Notice that non-overlap of O and O^* is more serious than non-overlap of I and I^*, because the latter dissimilarity may result from differences in labeling the input symbols.

The second aspect to be evaluated relates to the contribution of the mental model in the predictability, or control, of the behavior of the device. Information Theory expresses this predictability, or control, aspect as information transmission. In Part II we will adopt the second approach that defines information transmission as

$$T(I, O) = H(O) - H(O \mid I) \qquad (15)$$

where $H(O \mid I)$ denotes the entropy in O given I. The question of whether or not the participant's mental model encompasses a postulated set of model states may be investigated by testing the statistical significance of the increase in proportional reduction of error variance when the postulated set of model states is included in the analysis. This analysis requires the comparison of a two-way classification table (see above) with a three-way table. Several approaches for analyzing three-way classification tables are possible. We choose an approach discussed by Wickens (1989) in which the transmission of information between two factors is conditioned on levels of the third. More specifically, we will consider the mapping of input symbols $s \in I$ on output symbols $o \in O$ given the model states $q \in Q$. Let Q be restricted to level q_i, then the transmission between I and O is:

$$T(I, O \mid q_i) = H(O \mid q_i) - H(O \mid I \cap q_i) \qquad (16)$$

Thus, $T(I, O \mid q_i)$ is a two-way transmission statistic for each level q_i. The conditional information transmission $T(I, O \mid Q)$ is the weighted mean over these two-way statistics, $T(I, O \mid Q) = \Sigma_i\, p(q_i).T(I, O \mid q_i)$. In the passages that follow, data of two participants (Appendix 2) will be discussed to illustrate the analytical possibilities of the approach.

Participant #75. This participant demonstrates an awareness of the complete set of input and output symbols that are part of the definition of the mental model as a finite-state machine. Table 2 presents the data of participant #75 as a two-way classification table of input symbols by output symbols, without considering the different model states. It displays the amount of noise associated with each input symbol and its contribution to the average amount of entropy that remains when the input symbols are known. The columns 2 to 7 contain the conditional probabilities, $p\ (O \mid i)$ and column 9 the entropies $H(O \mid i)$ associated with each input symbol. The final column displays the contribution of each symbol to the average observed entropy in O, that is, $H(O \mid I)$. Each symbol's contribution is weighted according to its relative frequency (i.e., $p(i).H(O \mid i)$). The total entropy in O is equal to 2.57, and $H(O \mid I)$ equals 1.64. Consequently, $T(I, O) = .93$. The proportional reduction of error according to Equation (13) is .36. Note that input symbol MoForw has the largest contribution to the average amount of noise produced by the input symbols. As can be inferred from Appendix 2 this source of noise disappears when different model states are taken into account (see also Table 3). Table 3 demonstrates that the prediction of the output symbols was substantially improved by taking the postulated model states into consideration. For each of the states of the mental model the participant correctly maps the input symbols onto the output symbols, except for some noise in the mapping of TuLeft onto the output symbols *r-* and *r+,* while participant's mental model is in state "north". This noise represents the familiar phenomenon of left-right confusion found in young children. Note that this analysis does not quantify the amount of equivocation of TuLeft and TuRight in model state "west" (compare Table 3 with participant #75 data in Appendix 2). In fact, the analysis shows that left-right confusion can take two forms, either it represents uncertainty about the actions attached to an input symbol, or uncertainty about the input symbols attached to an action. Table 4 presents some conditional information transmission statistics. In summary, the conditional uncertainty in O, $H\ (O \mid Q)$, amounts to 1.022. The conditional information transmission $T\ (I,\ O \mid Q)$ equals .895. The proportional reduction in error now equals .876.

Observe now that $T\ (I,\ O)$ and thus the proportional reduction in error in Table 2 would be zero if the probabilities in every row would be the same as the marginal

Table 2. Association of input symbols and output symbols without considering different model states (data participant #75).

input	output symbols						p(i)	H(O|i)	p(i).H(O|i)
symbols	y-	y+	x-	x+	r-	r+			
MoForw	0.277	0.266	0.213	0.245	0	0	0.686	1.993	1.367
MoBack	0	1	0		0	0	0.022	0	0
TuLeft	0	0	0	0	0.529	0.471	0.248	0.998	0.248
TuRight	0	0	0	0	0.167	0.833	0.044	0.650	0.028
p(o):	**0.177**	**0.19**	**0.136**	**0.156**	**0.129**	**0.143**		**H(O|I)=**	**1.643**

probabilities $p(o_k)$. This condition defines independence of *I* and *O*. Deviations from this condition define dependence or 'interaction' of *I* and *O*. In general, an interaction exists if $p\ (s_i,\ o_k)$ is not the same as $p\ (s_i) \times p\ (o_k)$ for at least one cell in the table. Because the observed probabilities are subject to sampling fluctuations, this condition may be tested formally by a statistical test for independence. For a two-way contingency table such as table 2, the well-known Pearson χ^2 statistic or the loglikelihood ratio statistic may be used. Both have approximately a chi-squared distribution in large samples. More precisely, the validity of the chi-squared approximation depends on the expected frequencies under the null hypothesis of independence. As observed earlier, such tests of the statistical significance can be interpreted as fitting models and testing the differences of fit between models. Loglinear models for contingency tables are often applied for this purpose. For the two-way contingency table, the model of independence is $M + I + O$, where M is a "general mean" and I and O are the effects of input and output. The model $M + I + O + I * O$ is the model for dependence or interaction of I and O. This is the so-called saturated model, which always fits the data perfectly (i.e., the expected frequencies generated by the model are the same as the observed frequencies). The loglikelihood ratio statistic for the

Table 3. Contributions to the entropy of input symbol per model state (data: participant #75).

model states:	input symbols:	p(i)	H(O|i)	p(i).H(O|i)
north	MoForw	0.500	0	0
	MoBack	0.058	0	0
	TuLeft	0.346	0.964	0.334
	TuRight	0.096	0	0
east	MoForw	0.719	0	0
	TuLeft	0.281	0	0
south	MoForw	1.000	0	0
west	MoForw	0.714	0	0
	TuLeft	0.250	0	0
	TuRight	0.036	0	0

independence model can thus be interpreted as comparing the fit of this model with the model that includes the interaction. The Pearson χ^2 statistic compares the observed and expected frequencies. Loglinear models generalize to higher-dimensional contingency tables. The saturated model for table 4 is $M + I + O + Q + I * O + I * Q + O * Q + I * O * Q$. The last term signifies that the interactions $I * O$ are different for different states, which effectively means that inclusion of the state effect increases the proportional reduction in error.

All loglinear models possible with the three factors *I*, *O* and *Q* were fitted to the data of participant #75. Evaluating the fit of these models by the standard Pearson or loglikelihood ratio statistics was however problematic, because many expected frequencies were less than one, which probably makes the chi-squared approximation invalid.[4] Models including two two-factor interactions generated so many zero expected frequencies that no degrees of freedom remained for evaluating the models. However, it seems safe to conclude that the no-interaction model (i.e., model $M + I + O + Q$) does not fit the data of participant #75, χ^2(80 df) = 753.176. This statistics compares the model with the saturated model, which is the model for which the input-output interaction is different for different states. The model including the main effects and input-output interaction could not be evaluated because too many expected frequencies were zero.

Table 4. Conditional information transmission statistics (data participant #75).

model states:	p(q.)	H(O | q.)	p(q.).H(O|q.)	H(O|I∩q.)	T(I,O|q.)	p(q.).T(I,O|q.)
q1: north	0.380	1.700	0.645	0.334	1.366	0.518
q2: east	0.234	0.857	0.200	0	0.857	0.200
q3: south	0.182	0	0	0	0	0
q4: west	0.204	0.863	0.176	0	0.863	0.176
		H(O|Q) =	**1.022**		T(I,O|Q)=	**0.895**

Finally, the data provide some insight into a redundancy caused by the mental model. Recall that in each problem the bus starts out facing "north". The problems were designed such that the goal state required the bus to make 6 times an *y*+ translation (either forward or back), and 6 times an *y*- translation (idem). The remaining 12 problems had an initial state (*IS*) and a goal state (*GS*) with identical Y-coordinates, but the path between *IS* and *GS* was blocked so that the bus initially had to be moved across the *Y*-axis (either a *y*- move or a *y*+ move). Also, in 12 problems reaching the goal state required an *x*- translation and 12 problems required an *x*+ translation. Since the bus can only move forward or back, this implies that the bus first must make a turn (*r*- or *r*+) in order to be able to move along the *X*-axis. Further, recall that the instruction urged the students to take the shortest route from *IS* to *GS*. In summary, the experimental conditions were designed such that a uniform distribution of input symbols was not hampered by extraneous variables. Redundancy is defined as the difference between the entropy of a uniform distribution, $H\,(I)_{\max}$, and the observed entropy, $H\,(I)$ (Krippendorff, 1986). The relative frequencies by which participant #75 used the input symbols *F*, *B*, *L*, and *R* in model state "north" were .500, .057, .346, .096, respectively. Or rather, if we

[4] There is evidence that Pearson's χ^2 more closely resembles the chi-squared distribution in sparse tables (see e.g., Read & Cressie, 1988). Read and Cressie also present statistics for which the chi-squared approximation is closer to the distribution in small samples with small expected frequencies, but these were not used here.

correct for the left-right confusion in model state "north" these percentages would be .500, .057, .211, and .230, respectively. The maximum uncertainty, $H\ (I)_{\mathrm{max},}$, equals 2 bits, whereas the observed entropy (after left-right correction), $H\ (I)$, is equal to 1.697 bits, which means an uncertainty reduction of 16.5 %. A plausible explanation for this redundancy seems to be the notion that a bus cannot easily be driven backward. Therefore, a low likelihood of choosing input symbol *B* seems to follow from the mental model. This redundancy may be an indicator of the strength of the mental model and may differ across participants.

Table 5. Two-way cross-tabulation of input and output symbols (data: participant #32).

				output symbols				
input symbols	y-	y+	x-	x+	r-	r+	not-O	total
MoForw	13		5	2			7	**27**
MoBack		13						**13**
TuLeft					2			**2**
TuRight						2		**2**
not-I							31	**31**
	13	**13**	**5**	**2**	**2**	**2**	**38**	**75**

Participant #32. We will discuss only one aspect of the data of this participant. Participant #32 appears to use an input and an output alphabet which contain the postulated alphabets as subsets (see data participant #32 in Appendix 2). Table 5 presents a simplified overview of the mappings of the input symbols to the output symbols. Verbal instructions and physical moves of the bus that could not be categorized as elements of the postulated input and output alphabet was categorized as not-*I* and not-*O*, respectively. Table 5 clearly shows that most input symbols $s \in I$ map onto output symbols $o \in O$. All input symbols $s \notin I$ map onto output symbols $o \notin O$. This suggests the existence of a core model with an input alphabet and output alphabet similar to the postulated alphabets. In addition, participant #32 uses symbols, which seem to represent a relatively independent somewhat degenerated version[5] of the core model. Figure 3 shows the conditional probabilities of the occurrence of a not-*I* symbol given the occurrence of a particular model state. The relationship between these probabilities and the model states can be accurately described ($R^2 = .93$) by a second degree polynomial, suggesting that the difficulty the participant experienced in utilizing the core mental model (i.e., more or less similar to the postulated model of bus model) is a function of the absolute difference between the participant's orientation and the orientation of the object on the map.

[5] Degenerated version because these (input and output) symbols have lost the distinction between turn and move.

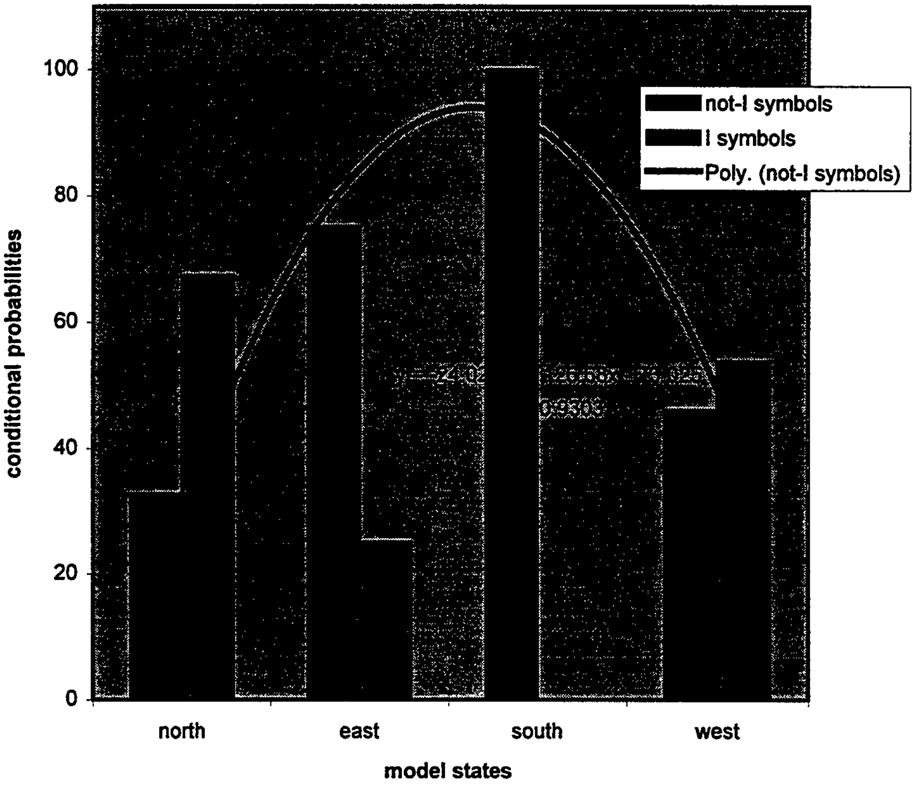


Figure 3.

Conditional probabilities of *I* symbols and not-*I* symbols given the occurrence of a model state.

General Discussion

In this paper we proposed to formally characterize mental models as a finite-state machine, which is fully described by its functions δ and λ. The motivation to provide such a formal description was that the theory of finite-state machines defines a class of mathematical structures with well-known properties. Thus, results of formal analysis can serve as a basis for studying cognitive processes and cognitive representations of interactive environments in several ways. First, this formalism makes it possible to achieve a fully parsimonious description of any particular mental model, because the theory of finite-state machines provides systematic ways of achieving a minimal form of a machine that defines a given set of input-output mappings (Denning et al., 1978). Second, this formalism can also be used as a tool to identify factors that determine the complexity of an interactive device such that devices with different complexity can be designed and experimentally compared (see e.g., Ippel and Meulemans, 1997). In particular, in combination with an information theoretical analysis the information load of an interactive device in terms of information to be transmitted can be objectively quantified (i.e., the maximum entropy to be reduced in order to attain control over a device). Mental models can be evaluated in terms of the reduction in information transmission load they produce. If we consider the example presented in Part II of this paper, it becomes clear that the mental model of the bus induces a preference in participants for the input signal F over B and a slight preference for R over L. The redundancy created by this input action pattern in turn implies a reduction in information transmission load. At the same time, however, it requires longer input strings (i.e., more input actions) to achieve the same goal. To answer the question as to how beneficial this is to the effective control of devices, further research is needed.

References

Abelson, H., & diSessa, A. A. (1980). Turtle Geometry: the computer as a medium for exploring mathematics. Cambridge, MA: The MIT Press.

Attneave, F. (1959). Applications of Information Theory to Psychology. New York: Holt.

Campbell, P.F., Fein, G.G., Scholnick, E.K., Schwarts, S.S., & Frank, R.E. (1986). Initial mastery of the syntax and semantics of Logo positioning commands. Journal of Educational Computing Research, 2 (3), 357-377.

Charniak, E., & McDermott, D. (1985). Introduction to Artificial Intelligence. Reading, MA: Addison-Wesley.

Cohen, R. (1987). Implementing Logo in the grade two classroom: Acquisition of the basic programming concepts. Journal of Computer-Based Instruction, 14, 4, 124-132.

Davis, M.D., Sigal, R., & Weyuker, E.J. (1994). Computability, Complexity, and Languages. San Diego, CA: Academic Press.

Denning, P.J., Dennis, J.B., and Qualitz, J.E. (1978). Machines, Languages and Computation. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Fay, A.L., & Mayer, R.E. (1988). Children's naive conceptions and confusions about Logo graphics commands. Journal of Educational Psychology, 79, 3, 254-268.

Goodman, L.A., and Kruskal, W.H. (1979). Measures of association for cross-classification. New York: Springer Verlag.

Holland, J.H., Holyoak, K.J., Nisbett, R.E., & Thagard, P.R. (1986). Induction. Processing of Inference, Learning and Discovery. Cambridge, MA: The MIT Press.

Holyoak, K.J., Koh, K. & Nisbett, R.E. (1989). A theory of conditioning: Inductive learning within rule-based default hierarchies. Psychological Review, 96, 2, 315-340.

Holyoak, K.J. (1985). Mental Models in Problem Solving. In J.R. Anderson and S. Kosslyn (Eds.), Tutorials in learning and memory.

Ippel, M.J., and Beem, A.L. (1997). Mental Models As Finite-State Machines: Assessment of Dynamic Knowledge Representations. Manuscript submitted for publication.

Ippel, M.J., and Meulemans, C.J.M. (1997). Simplifying the Semantic Structure of the LOGO Turtle World: Its Effects on the Acquisition of the Syntax and Semantics of the LOGO Basic Commands. Manuscript submitted for publication.

Johnson-Laird, P.N. (1983). Mental Models. Cambridge, U.K.: Cambridge University Press.

Johnson-Laird, P.N. (1989). Mental Models. In M.I. Posner (Ed.), Foundations of Cognitive Science (pp. 469-499). Cambridge, MA.: The M.I.T. Press.

Kolman, B., and Busby, R.C. (1987). Discrete Mathematical Structures For Computer Science. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Krippendorff, K. (1986). Information Theory. Structural Models for Qualitative Data. London: Sage Publications.

Minsky, M.L. (1967). Computation. Finite and Infinite Machines. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Newell, A. (1990). Unified Theories of Cognition. Cambridge, MA.: Harvard

University Press.

Newell, A., and Simon, H.A. (1972). Human Problem Solving. Englewood Cliffs: Prentice-Hall.

Papert, S. (1980). Mindstorms: Children, Computers and Powerful Ideas. New York: Basic Books.

Payne, S.J. (1988). Methods and mental models in theories of cognitive skill. In J. Self (Ed.). Artificial Intelligence and Human Learning. London: Chapman and Hall.

Payne, S.J. (1992). On Mental Models and Cognitive Artifacts. In Y.Rogers, A. Rutherford, and P.Bibby (Eds.). Models in the Mind. Theory, Perspective and Application (pp. 103-118). London: Academic Press.

Pylyshyn, Z.P. (1984). Computation and Cognition. Toward a Foundation for Cognitive Science. Cambridge, MA.: The M.I.T. Press.

Read, T.R.C., & Cressie, N.A.C. (1988). Goodness-of-Fit Statistics for Discrete Multivariate Data. New York: Springer Verlag.

Rogers, Y., and Rutherford, A. (1992). Future Directions in Mental Models Research. In Y.Rogers, A. Rutherford, and P.Bibby (Eds.). Models in the Mind. Theory, Perspective and Application (pp. 289-314). London: Academic Press.

Shannon, C.E., and Weaver, W. (1949). The Mathematical Theory of Communication. Urbana: University of Illinois Press.

Wickens, Th. D. (1989). Multiway Contigency Tables Analysis for the Social Sciences. Hillsdale, N.J.: Lawrence Erlbaum.

APPENDIX 1

Let $P : W \rightarrow Q$ be a function of W into Q. Let R be the relation on W defined by sRt iff $P(s) = P(t)$, for s and t in W. Then:

A. R is an equivalence relation. A proof of this proposition can be found in Denning et al. (1978). It is included in this Appendix in order to facilitate the reading of proposition B.

B. The function m: $W/R \rightarrow Q$ is a one-to-one function.

Proof.

A. Since P: $W \rightarrow Q$ is a function, for each $q \in Q$, let W_q be the set of states in W that map onto q:

$$W_q = \{x \in W \mid q = P\,(x)\}$$

We assume that P is an everywhere defined function. Thus, each $x \in W$ is element in exactly one set W_q and the sets W_q form a partition of W. The corresponding relation R is defined by

$$sRt \qquad \text{iff}\; P\,(s) = P\,(t)$$

Note that R is clearly reflexive and symmetric. Suppose now sRt and tRu for s, t, and u in W. Then, sRu holds, so R is transitive, and therefore R is an equivalence relation. R partitions W into equivalence classes usually denoted by $[w]$:

$$[w] = \{x \in W \mid P(x) = P(w)\}$$

The partition of W consists of all equivalence classes of W and is denoted W/R. W/R is also called the quotient set defined on W by R. This establishes that a function determines an equivalence relation on its domain.

B. Let R be the equivalence relation defined on a set W and let W/R be the corresponding quotient set (see ad a). Let p: $W \rightarrow W/R$ and m: $W/R \rightarrow Q$ be functions. If $[w] \in W/R$, and $q \in Q$, then the function m: $W/R \rightarrow Q$ is defined by

$$m\,([w]) = q$$

Since m: $W/R \rightarrow Q$ is a function $m([w])$ has a single value. Also $[w] = W_q = \{x \in W \mid q = P(x)\}$ (see ad A). Thus, $m^{-1}([w])$ is also a function, that is, a function from Q onto W/R. Therefore, the function m: $W/R \rightarrow Q$ is a one-to-one function.

APPENDIX 2

Participant #75

model	input			output	symbols				
states:	symbols:	y-	y+	x-	x+	r-	r+	not-O	Total
north	MoForw	13						6	**19**
	MoBack		13						**13**
	TuLeft					1			**1**
	TuRight						2		**2**
	not-I							17	**17**
east	MoForw				2				**2**
	not-I							6	**6**
south	not-I							2	**2**
west	MoForw			5				1	**6**
	TuLeft					1			**1**
	not-I							6	**6**
Total		**13**	**13**	**5**	**2**	**2**	**2**	**38**	**75**

Participant #32

model	input			Output symbols				Total
states:	symbols:	y-	y+	x-	x+	r-	r+	
north	MoForw	26						**26**
	MoBack		3					**3**
	TuLeft					11	7	**18**
	TuRight						5	**5**
east	MoForw				23			**23**
	TuLeft						9	**9**
south	MoForw		25					**25**
west	MoForw			20				**20**
	TuLeft					7		**7**
	TuRight					1		**1**
Total		**26**	**28**	**20**	**23**	**19**	**21**	**137**