ELSEVIER

Cognitive Science 27 (2003) 227–257

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Modeling developmental transitions on the balance scale task

Hedderik van Rijn*, Maarten van Someren, Han van der Maas

Department of Psychology, University of Amsterdam, Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands

Received 20 June 2001; received in revised form 30 April 2002; accepted 31 July 2002

Abstract

Periods of relatively stable, rule-like behavior alternated with short transition periods characterize cognitive development on reasoning tasks like the balance scale task. Each transition gives rise to an improvement in behavior, until a phase is reached in which performance is flawless or improvement is not worthwhile given the necessary extra effort. Several computational models have been developed to capture the developmental phenomena associated with the balance scale task. These models, which originate from different computational traditions, explain the main phenomena of development. Recently, empirical phenomena have been reported that these models cannot easily accommodate. We propose a computational model that is implemented in ACT-R and that is based on the evaluation of success of applied knowledge, combined with a mechanism to construct new knowledge by searching for differences between the left- and right-hand sides of presented balance scale problems. This model accounts for the main empirical phenomena as well as for the recently reported empirical phenomena such as learning without feedback.

Keywords: Cognitive development; Balance scale task; ACT-R model

Cognitive development in many domains is conceptualized as a progression through a series of increasingly complex and increasingly accurate task-specific stable phases. Behavior on the balance scale task, a task that is related to proportional reasoning, displays this type of developmental progression (Chlestos, De Lisi, Turner, & McGillicuddy-De Lisi, 1989; Jansen & Van der Maas, in press; Siegler, 1981). Periods of relatively stable performance ("phases") alternate with short periods of unstable performance during which a new phase is attained. In

* Corresponding author.
E-mail address: hedderik@van-rijn.org (H. van Rijn).

0364-0213/02/$ – see front matter
PII: S0364-0213(02)00116-7

a balance scale task, a child is asked to predict the direction of movement of a balance scale. Pegs are situated at equal distances from each other and the fulcrum. Identical weights are placed on one of the pegs at both sides of the fulcrum. The balance will tip to either side, or remain in balance, depending on the number and the positions of the weights. This task is used to investigate the strategies that children employ in solving this task, the effect of training on the children's behavior, and the transition from one strategy to the next (e.g., see Siegler, 1976, 1981; Siegler & Chen, 1998). Siegler distinguished six types of balance scale items,[1] which he used to categorize behavior in terms of the inferred strategies. The six types are divided into three simple types and three so-called conflict-types. The simple item types are: (a) *Balance items* with equal numbers of weights at each side and equidistant to the fulcrum; (b) *Weight items* with unequal numbers of weights at each side, equidistant from the fulcrum; and (c) *Distance items* with an equal number of weights at each side, but at different distances from the fulcrum. These item types are called simple item types as the weight and distance information do not conflict. The three conflict item types are characterized by both different numbers of weights and different distances from the fulcrum. Moreover, the answers based on each dimension conflict: if the weight dimension predicts the answer to be "the balance scale tips to the left side", the distance predicts the opposite. Depending on the differences between the weights and distances, the scale tilts to the side with the largest number of weights or to the side where the distance to the fulcrum is greatest. In (d) *Conflict Weight items*, the balance scale tilts to the side with the larger number of weights. In (e) *Conflict Distance items*, the scale tilts to the side where the distance to the fulcrum is greater. In (f) *Conflict Balance items*, the effects of the two dimensions compensate: the scale remains in balance. The behavior of children on the balance scale task has been studied extensively by comparing their performance on the different item types. These studies have established repeatedly a number of empirical phenomena which will be discussed in the next section.

Identifying phenomena does not provide an account of the mechanisms underlying behavior. As argued by many authors (e.g., Boden, 1979; Anderson, 1987; Siegler, 1989), one needs to understand the mechanisms that cause the changes in behavior to obtain a causal explanation of development. In attempts to explain these mechanisms in the case of the balance scale task, several computational models have been proposed. These models, which originate from different computational traditions, explain the main phenomena of development. Recently, new empirical phenomena have been reported which these models cannot easily accommodate.

This paper consists of three parts. We first describe the main empirical phenomena associated with development on the balance scale task. Second, we discuss existing computational models of the balance scale task. Finally, we present a computational model that does account for the major empirical phenomena and we discuss its explanatory value.

1. Empirical phenomena and criteria

In this section, we discuss the major empirical phenomena (EP) associated with the balance scale task. On the basis of these phenomena, we formulate four sets of empirical criteria. Only if a computational model satisfies these four criteria, it can be considered a model of the full range of behavior associated with the balance scale task.

1.1. Empirical phenomena

1.1.1. EP1: Stable phases and transitions

The behavior of children on the balance scale task is often classified according to their inferred use of a small number of "Rules".[2] The notion of Rule use is based on the observation of homogeneous behavior on balance scale items of a particular type. This "consistency criterion" (Reese, 1989) is often cited as evidence for rule use. Within the balance scale domain, the consistency criterion is operationalized by Siegler's (1981) rule assessment methodology. Using this methodology, children's responses are compared with the responses implied by the various Rules. Their behavior is classified according to the closest match between their responses and those associated with a given Rule. Jansen and Van der Maas (1997) improved the rule assessment methodology by applying latent class analysis. Latent class analysis[3] is a statistical technique used to determine the number and kind of unobservable classes of behavior which give rise to observed responses on balance scale items. Using this technique, Jansen and Van der Maas (1997, 2002) provided a statistically sound foundation of Siegler's (1976, 1981) earlier results.

The overall picture that emerges from these studies is that children adopt progressively more powerful Rules during development, which results in increasingly better performance. Between transitions, children display stable and consistent behavior on the balance scale task. During the transition from one Rule to the next, performance is more erratic as some items of a particular type are already answered correctly, whereas others are still answered according to the earlier Rule yielding incorrect answers. The exact mechanism of transition may differ from Rule to Rule.

1.1.2. EP2: The Rules

Siegler (1976, 1981) has identified four Rules that characterize children's behavior during different phases. These Rules are shown in Fig. 1. Rule I involves the following steps: consider the number of weights on each side: if equal, predict that the balance scale will remain level, if unequal, predict that the balance scale will tip to the side with greater weight. Like Rule I, Rule II states that the scale will tip to the side with the greater number of weights. However, if the weights are equal, the distance dimension is taken into account. Rule III states that both the weight and distance dimension are taken into account. If one side has greater weight and the other greater distance, the child guesses. (Described as "Muddle Through" in Siegler, 1976.) Rule IV states that both dimensions are considered and if the dimensions conflict (as in the conflict item types), the torque is computed by multiplication of the weights and distances. Note that these Rules have multiple conditional responses, and are therefore more complex than "standard" IF–THEN rules.

The unsystematic behavior *before* Rule I is often described as Rule 0 or pre-Rule I. The first three Rules (Rule I–Rule III) are comparable because they consist of simple steps consisting only of comparisons. Rule IV includes an additional step that involves multiplication. Rule I is completely unidimensional, as only the weights are taken into account. Rule II is neither completely unidimensional nor multidimensional, as it can be applied by switching attention from one dimension to another. Only Rule III and Rule IV require full multidimensional reasoning as both weight and distance values are considered simultaneously (Siegler, 1996).

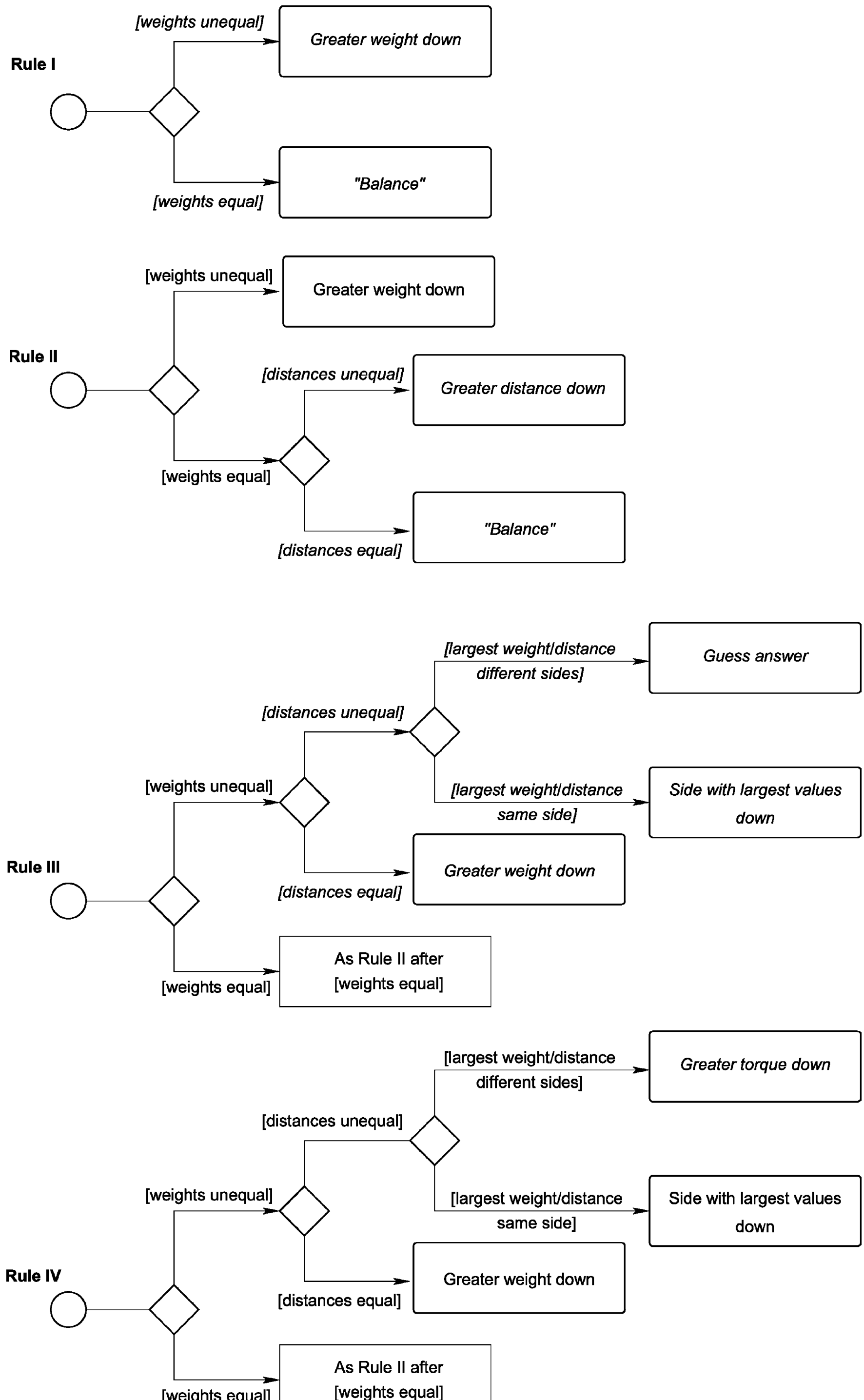

Fig. 1. Rules that categorize behavior as identified by Siegler (1981).

The period in which a child uses a particular Rule is referred to as a Phase (e.g., Rule I is used in Phase I).

After Siegler's initial study, these results have been replicated in numerous other studies. Besides confirmation of the originally presented Rules, some new Rules were proposed. However, a number of these Rules are hard to identify empirically as these predict the same performance as other Rules (e.g., see Jansen & Van der Maas, in press, for a latent class based analysis and categorization of Rules used by children). An exception is the Addition Rule (Ferretti, Butterfield, Cah, & Kerkman, 1985; Normandeau, Larivee, Roulin, & Longeot, 1989). The Addition Rule states that the answer is based on the addition of weight and distance values when presented with conflict items.

It is sometimes doubted whether subjects classified as Rule IV use multiplication rather than some other procedure, such as perceptual approximation or problem–answer mappings. However, in studies in which the rule assessment classification is studied by asking for verbal explanations, most subjects classified as Rule IV refer to multiplication in their explanations (Siegler, 1976, 1981). Another issue with regard to Rule IV is whether Rule IV can be learned without explicit instruction which would have serious implications for balance scale models. As advanced subjects new to the balance scale task regularly use Rule IV to solve the problems, the Rule can be learned without explicit balance scale task instruction. Of course, it is possible that they received explicit instruction in related domains. But irrespective of how the Rule is learned, a significant group of subjects master Rule IV, and so should complete balance scale models do.

1.1.3. EP3: Transition phenomena without feedback

In a recent study, Jansen and Van der Maas (2001) describe phenomena associated with the transition from Rule I to Rule II. Jansen and Van der Maas presented a sequence of distance items to Phase I children. The weights of the first item were situated on the first peg left from the fulcrum, and on the second peg right from the fulcrum. During the first part of the sequence, the distance between the fulcrum and the weights was increased by moving the weights on the right side one peg outward per presentation. During the second part of the sequence, which started once the weights had reached the most extreme peg, the distance was decreased again by moving the weights back toward the fulcrum. This way, the difference between the left and right distance values first increased and then decreased. The choice to manipulate distance was based on research that suggests that the *availability* or *encoding* of the distance dimension influences performance on the balance scale task (Siegler, 1976; Siegler & Chen, 1998, Exp. 3). As in most balance scale studies, the children were *not* given any feedback concerning their responses. Of the 314 children tested (age range: 6–10 years), 27% displayed behavior that was not consistent with Rule I over the sequence of difference distance manipulations. Three interesting deviations from Rule I were found. Fig. 2 illustrates these patterns and the associated responses.

Four percent of the children displayed a change from Rule I to Rule II behavior and from Rule II to Rule I at exactly the same point in the sequence. This pattern, which is shown in Fig. 2a, is consistent with the so-called "Maxwell convention" (see Jansen & Van der Maas, 2001). Three percent of the children displayed a switch at a *different* point in the sequence depending on whether the difference in distances was increasing or decreasing. This pattern, which is shown in Fig. 2b, is consistent with the so-called "delay convention". Nine percent of the children

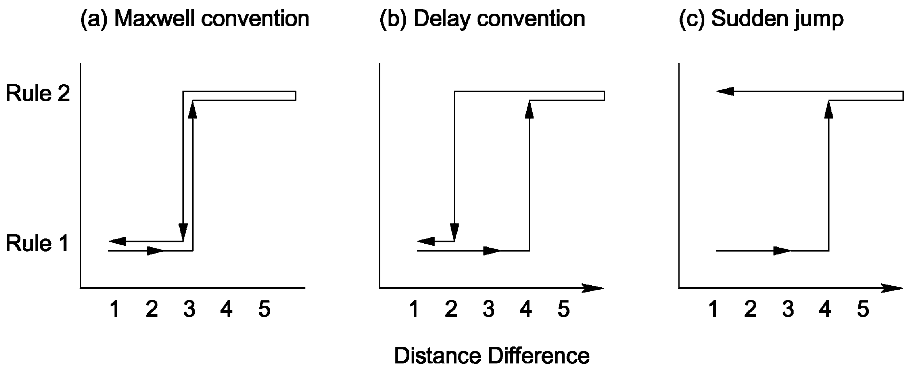

Fig. 2. Three transition patterns, x-axis denotes the difference in distances left and right of the fulcrum, the y-axis denotes the used Rule. Note that the actual distance difference values are not constrained, for example, the Maxwell convention can also occur at distance difference four.

displayed a sudden jump from Rule I to Rule II as the distance difference was increased, but did not return to Rule I as the distance difference was decreased. This so-called "sudden-jump" response pattern is shown in Fig. 2c. Besides these three groups, eleven percent of the children displayed behavior that deviated from Rule I, but that was not classifiable. Regrettably, such detailed information is not available for other transitions; Jansen and Van der Maas (2001) only studied the transition from Rule I to Rule II. Note that these results do not necessarily imply that manipulating the distance dimension is the only or even the most important way to trigger transitions. However, the results do show that transitions can be induced without feedback. In general, this behavior is comparable to learning without feedback which has also been posited in areas as language learning (Pinker, 1984; Taatgen & Anderson, 2002) and implicit learning (e.g., Cleeremans & McClelland, 1991).

1.1.4. EP4: Torque Difference Effect

A similar but less specific effect is the "Torque Difference Effect". This effect refers to the finding (Ferretti & Butterfield, 1992; Ferretti et al., 1985) that within item types, items with a large difference in torque are more likely to be answered in a manner consistent with a more advanced Rule than items with a smaller torque difference. This suggests heterogeneous behavior within item types. Such heterogeneity challenges the notion of Rules and the rule assessment method as proposed by Siegler. However, reanalysis of the data (Jansen & Van der Maas, 1997) shows that the Torque Difference Effect only occurs with items with extreme torque differences. Therefore, it is concluded that the children's behavior is homogeneous for moderate torque difference levels and that a torque difference effect is limited to large torque differences.

1.2. Empirical criteria

Ideally, these behavioral phenomena should be reproduced by models of development on the balance scale. Therefore, we translate these phenomena into four sets of empirical criteria for models of balance scale behavior.

1.2.1. EC1: Rule-like behavior

The behavior of the model should be classifiable into Rules. This classification can be conducted by latent class analysis or by inspecting the models' implementation directly.[4] As different levels of rule-like behavior implies transitions, the model must explain transitions from one Rule to another.

1.2.2. EC2: Rule sets

A complete model of balance scale behavior should include the four Rules as identified by Siegler (1976), and the later identified Addition Rule (Normandeau et al., 1989). The Rules should appear in a fixed order.

1.2.3. EC3: Transition patterns without feedback

The model should explain how transitions occur on series of items with increasing and decreasing distance differences *in the absence of feedback*. In particular, the model should give rise to the three transition patterns as presented in Fig. 2. This criterion only applies to the transition from Rule I to Rule II as detailed results concerning the transitions between the other Rules are not available.

1.2.4. EC4: Torque Difference Effect for large torque differences

A model should reproduce the Torque Difference Effect for large torque differences but not for small difference values.

2. Computational models

Several computational models have been proposed to explain development on the balance scale. We discuss models of balance scale behavior based on production rules, decision trees, and neural network models. Without entering into the discussion on the psychological validity of these architectures, the models are discussed in terms of the degree to which the empirical criteria are satisfied.

Obviously, the models should have access to some form of feedback. Without feedback, none of the models would be able to show development through the phases. However, in accordance with EC3, the models should be able to explain how local developmental transitions can take place without feedback.

2.1. Production rule models

We discuss two production rule models.[5] The first computational model of balance scale behavior is Klahr and Siegler's (1978) production rule model. They showed that different sets of static production rules were able to capture the observed Rules. The model did not specify a transition mechanism and it therefore failed to meet the first, most important empirical criterion (EC1).

Sage and Langley (Langley, 1987; Sage & Langley, 1983) presented the first computational model that actually shows development from Rule I to Rule III. As shown in Fig. 1, the central idea is that Rule II adds extra conditions to Rule I, as does Rule III to Rule II. Their model starts with a set of production rules that produce random answers. The model learns by discrimination: if the model produces an incorrect response to an item, a new production rule is created based on an analysis of the differences between this item and the last item which the production rule solved correctly. Performance is kept stable for some time by a strength parameter. A new production rule is created when no production rule is available with a strength level above a certain threshold. This model shows rule-like behavior and transitions (EC1) from Rule I to Rule III. The model does not acquire Rule IV, because this Rule cannot be constructed as the concept of multiplication is not available to the model. The same holds for the Addition Rule. However, incorporation of multiplication and addition mechanisms will probably result in the construction of the appropriate Rules.

The discrimination method used in this model often gives rise to production rules that result in behavior not found in empirical data. The model therefore satisfies EC2 only partially. It is unclear how this model may incorporate the non-feedback related phenomena (EC3). The model only checks whether weights and distances differ and does not use the actual values in conditions of production rules. Therefore, it is not sensitive to differences within item types and fails to satisfy EC4.

The ACT-R architecture and the model that we present below share a number of features with this production rule model. ACT-R incorporates the idea of usefulness parameters for production rules, but in a more sophisticated way. As in the models of Sage and Langley, our model also constructs new production rules. However, as discussed below, the new production rules are based on knowledge available to the ACT-R model, instead of on purely architectural mechanisms. In the Langley (1987) model, new production rules are constructed by adding conditions to existing rules. In the ACT-R model, new production rules are created that operate in cooperation with existing rules.

2.2. Decision tree model

Schmidt and Ling (1996) presented a model based on decision tree learning. This model acquires behavior on the balance scale by constructing decision trees. At the top-level of the tree is the most predictive test, one branch below are the next most predictive tests given the value on the top level test, etc. For example, a decision tree for Rule II (Schmidt & Ling, 1996, p. 217) starts with "Are weights and distances equal?" If both are equal, the model predicts that the balance scale will remain in "balance", otherwise the next test is "Which side has the most weights?" During the construction of the tree, new conditions are added until the tree reaches a pre-specified "error tolerance" threshold. The algorithm used by Schmidt and Ling uses a complete training set when constructing the tree. However, as pointed out by the authors, algorithms exists that produce similar trees by presenting training items incrementally.

The decision tree model is similar to the production rule models of Sage and Langley. A path from the top-level of the tree to an end-point can be viewed as a rule. The concepts from which the tests are constructed are binary concepts representing (a) the side with the larger number of weights, (b) the side with the larger distance and (c) a concept that represents whether the weights and distances for both sides are equal. Besides these concepts, which were also available in the above discussed models of Sage and Langley, the decision tree model also has access to concepts expressing the (d) numerical difference in weights and the (e) numerical difference in distances between both sides.

The model satisfies EC1 as it acquires trees that correspond to the Rules. Moreover, the transitions from one Rule to another are by definition sudden. As the model learns the first three Rules in the right order, it partially adheres to EC2. However, the concepts of addition and multiplication are not included in the model. Therefore, the model is not able to learn the Addition Rule and Rule IV is approximated by specific item-to-answer mappings. If multiplication were included in the model, it is unlikely that the other strategies would remain available. It is unclear whether this model can incorporate the non-feedback related phenomena (EC3). With respect to EC4, different Rules are constructed for the values of the numerical difference concepts (d and e), hereby reproducing the Torque Difference Effect. However, the Torque Difference Effect is produced consistently, that is, both for smaller and larger torque difference values.

2.3. Connectionist models

McClelland (1989, 1995) presented a connectionist model of balance scale behavior that learns by back-propagation of feedback. The network consists of two output nodes representing the predicted answer and ten input nodes, five for each side. Only one of the input nodes per side receives activation. The distance dimension determines which of the input units is activated, and the weight dimension determines the amount of activation. The input and output nodes are connected by separate hidden layers for the distance and weight dimensions. To explain the order of the Rules (i.e., the use of the weight dimension before the distance dimension), McClelland assumes that initially weight items occur more frequently than distance items.

The first empirical criterion poses a problem for this model. Using Siegler's rule assessment methodology, McClelland claimed a successful fit to human data. However, reanalysis of the model's behavior using latent class analysis showed that the behavior of the model cannot be described by a set of distinct rules (Jansen & Van der Maas, 1997). The model therefore fails to show homogeneity within item types. Furthermore, Raijmakers, Van Koten, and Molenaar (1996) showed that the model fails to show qualitative transitions between the Rules.

With respect to EC2, the model is not able to learn stable Rule behavior as it sometimes regresses after a period of stable behavior. Moreover, Rule IV is never learned. Because the model is only able to learn by feedback-dependent back-propagation, it fails the no-feedback criterion (EC3). With respect to EC4, the model does show the Torque Difference Effect (McClelland, 1995).

Shultz and his co-workers (Shultz & Schmidt, 1991; Shultz, Mareschal, & Schmidt, 1994; Schmidt, Buckingham, & Mareschal, 1995) modeled balance scale behavior with cascade-correlation networks. The central tenet of cascade-correlation is the introduction of new hidden nodes into the model if the back-propagation-like learning mechanism does not reduce the error between predictions and feedback fast enough. Here we will discuss the model as presented in Shultz et al. (1995). This model has five input units for each side of the balance scale and two output units. The distance dimension determines which of the input units is activated, and the weight dimension determines the amount of activation. Initially, as in McClelland's model, weight items are over-represented in the training problems to accommodate the initial preference for weight.

Table 1
Overview of empirical criteria per model

Empirical criteria	Model type				
	Production rules		Decision trees	Neural networks	
	KS’78	SL’87	SL’96	McC’95	S’95
EC1: Rule-like behavior	–	✓	✓	–	–
EC2: Rule sets	✓	+	+	–	+
EC3: Transition patterns without feedback		–	–	–	–
EC4: Torque Difference Effect		–	+	✓	✓

Note. ✓: criterion fully satisfied, +: criterion partly satisfied, −: criterion not satisfied. Note that the evaluations are based on the descriptions of the models as available in literature. This does *not* mean that models with “+” and “−” evaluations cannot be extended to match the criteria better. KS’78: Klahr and Siegler (1978), SL’87: Sage and Langley (1983), SL’96: Schmidt and Ling (1996), McC’95: McClelland (1995), S’95: Shultz et al. (1995). See text for a complete list of references.

According to analyses with the rule assessment methodology,[6] the behavior over time of the cascade correlation network follows the Rules. However, because the learning mechanism of the cascade correlation network is comparable to that of McClelland’s model, it is likely that the transitions are better described by *gradual* adaptation than by sudden transitions. This impression of gradual adaptation is reinforced by the remark made by Shultz (1994, p. 81) that: “The cascade correlation model suggests that balance scale [. . .] transitions are soft and tentative rather than abrupt and definitive”. The threshold for adding new hidden nodes plays a role in how long performance remains stable. This model therefore occupies a position between the production rule model and the back-propagation model with respect to the generation of Rules (EC1). This model learns all four Rules in the correct order (EC2). However, as no torque is calculated, the model can only approximate Rule IV behavior. Shultz et al. (1994) did not attempt to incorporate the Addition Rule in their model. Like McClelland’s model, it fails the no-feedback criterion (EC3), because the model is only able to learn by feedback-dependent back-propagation. The model does show the Torque Difference Effect (EC4).

2.4. Conclusions

Table 1 summarizes the empirical criteria that each model satisfies. Although each model provides an interesting perspective on the process of balance scale development, none is able to model all empirical data. The most prominent problem is that none of the models can explain learning without feedback. In the next section we present an account of behavior on the balance scale task that satisfies all four empirical criteria. This model is based on the ACT-R cognitive architecture.

3. An ACT-R model of balance scale behavior

ACT-R (Anderson, 1990; Anderson & Lebiere, 1998) is a hybrid cognitive architecture in which the use of symbolic knowledge is mediated by associated quantitative values.[7] In constructing an ACT-R model of balance scale behavior, we distinguish between three aspects of the model: mechanisms, task-specific concepts, and capacity constraints. Most important are the mechanisms. The mechanisms involve several ACT-R-specific processes, which will be explained in detail below. However, the key idea is a very general strategy, which underpins balance scale behavior in all phases. This strategy is: "solve balance scale items by searching for differences between the left and right side of the balance scale". The performance of this strategy is constrained by the presence or absence of task-specific concepts and by the available capacity. With respect to the task-specific concepts, the availability of a particular concept or property determines whether it can be used in the reasoning process. Examples are the facts that multiplication is not available initially, and that children notice the weights earlier than the distances. The capacity constraint, expressed as uni- versus multidimensional reasoning, influences the number of concepts (i.e., weight and distance) that can be incorporated simultaneously in the general strategy. Initially, children can only accommodate a difference on one dimension, which constrains the general strategy to "search for a difference between the left and right side of the balance scale". We will demonstrate how the interplay between these three aspects of the model explains the four empirical phenomena. First, we give an overview of the properties of the ACT-R architecture that are important in the present model. Then, we discuss the task-specific concepts and the capacity constraints. Finally, we present simulations of the developmental process including all phase transitions, the transition patterns and the Torque Difference Effect.

3.1. Key structures and mechanisms of the ACT-R architecture

3.1.1. Chunks & production rules

ACT-R stores knowledge in two types of memory. Declarative memory contains chunks that represent descriptive knowledge, whereas procedural memory contains production rules that represent procedural knowledge in the form of IF–THEN actions.

Each chunk is of a specific type that defines its function in the model. At each point in time, one of the chunks in the model denotes the goal of current processing.

At each processing step, ACT-R examines all production rules to test which production rules match the current goal. A production rule is considered to match if the conditions of that production are satisfied by the active goal. Besides conditions on the active goal, a production rule can also have conditions on the availability of declarative information. To retrieve information from declarative memory, a production rule initiates a retrieval request. This request specifies the conditions that determine the type of information that is required. Subsequent production rules can test for the availability of that knowledge, that is, they can test whether the given conditions returned a retrievable chunk. In addition to initiating a retrieval, the THEN-side of production rules can change the current focus to a new goal, modify the current goal or initiate other external actions.

3.1.2. Activation & utility

Instances of both declarative and procedural knowledge have associated quantitative parameters that express their usefulness. If the chunk or production rule has proven to be successful in achieving the goal, the associated parameter is increased in value. As a consequence, the chunk or production rule will be used more often, and vice versa. For declarative memory chunks, this quantitative parameter is represented as the activation level of the chunk. The activation level depends on a base-level activation and on activation sources originating in the model's context. The base-level activation (B) is determined by the number and recency of past retrievals of that chunk, i.e.,

$$B = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right).$$

The summation is over all previous retrievals (n) of this chunk, with t_j being the time in seconds between retrieval j and present, and d being the decay rate (by default fixed at 0.5). The default ACT-R context activation, called associative activation does not play an important role in the model presented here. However, in the second set of simulations presented in this paper, a different type of context activation is introduced that reflects perceptual saliency. When a model initiates a retrieval request, and multiple chunks with activation levels above the retrieval threshold satisfy the conditions of that retrieval request, the chunk with the highest activation is retrieved.

For production rules, the main parameter is the expected utility. The utility of production rules is determined by a combination of the proportion of successful completions of goals and the costs associated with that production rule: Utility $= U = P \times G - C + \sigma$. In this equation, P is the proportion of successful applications of the production rule. This proportion is multiplied by a constant G (20, by default). C reflects the cost associated with the execution of the production rule, expressed in units of time. The last variable in the equation, σ, represents normally distributed noise that may optionally be added to the utility.

The value of the parameters P and C depend on the performance of the current production rule *and* the performance of the production rules that follow the current one. With respect to the P parameter, a production rule is considered to be executed successfully if the goal to which it was applied is tagged as successfully solved. The value of the C parameter depends on both the costs of the current production rule and the costs of following production rules used for the same goal. Each production rule has a fixed cost of 50 ms. Besides these fixed costs, a production rule might take more time if it has to wait for a retrieval from declarative memory, or if it involves perceptual or motor actions. During the run of a model, the P and C parameters get updated on the basis of new experiences. For example, if a large number of goals are solved successfully, P will increase as the overall proportion of correctly answered problems increases. Similarly, if the costs associated with solving a problem decreases (e.g., because of acquiring a more efficient strategy), the C parameter will decrease.

As with chunks, if more than one production rule satisfies the current constraints, the production rule with the highest utility value is selected. In the simulations reported below, normally distributed noise was added to the utility. This results in the model occasionally trying alternative production rules. If an alternative production rule performs better than the original production rule, this is reflected in its utility parameters. Eventually, this alternative production rule will supersede the original.

3.1.3. Learning & production composition

ACT-R models are able to expand their knowledge by adding new chunks and production rules to their memories. The source of new declarative knowledge can be internal or external.

The internal source of new chunks are production rules. New chunks are added if the production rule modifies the goal-chunk to which it was matched, or if it constructs a new chunk in its THEN-side. An example of an external source of new chunks is perception.

A recent addition[8] to ACT-R (ACT-R v5.0, 2002b, production rule learning; Taatgen, 2000; Taatgen & Anderson, 2002) is production composition. Production composition creates new production rules by joining two production rules that occur in succession into one single rule. This process eliminates intermediate retrievals from declarative memory. For example, assume that two production rules are executed in succession. If the first production rule initiates a retrieval request and the second production rule processes the retrieved chunk by storing information from that chunk in the current goal, production composition constructs a new production rule that modifies the goal chunk in the same fashion as before, but without requiring a retrieval. In other words, production composition specializes a general procedure to solve a problem into a problem-specific procedure by eliminating the retrieval.

This composition mechanism also applies to learning new behavior from declarative descriptions of the involved actions. These representations (declarative actions) are available as chunks (cf., Anderson, 2000), for example as the results of observing others perform actions, or as the result of explicit instruction. The declarative actions can be matched and executed by "interpretive production rules". These production rules retrieve the declarative actions and modify the goal based on the contents of the declarative actions. Because of these modifications, other production rules now match the goal, giving rise to new behavior. By means of this process of alternating execution of interpreting and "normal" production rules, new sequences of behavior emerge on the basis of already available production rules.

The utilities of newly composed production rules are inherited from the parent rules. However, as the composed production rule often removed the necessity of a retrieval and combined two production rules into one, the costs associated with the new production rule are lower than the old combination of production rules. That is, the costs of the newly composed production rules does not involve the costs associated with extra retrievals (a variable amount of time) *and* the costs of executing the second production rule (a fixed 50 ms). As the success rate of the newly composed production rule is equal to the success rate of the non-composed production rules, the difference in costs will eventually favor the composed production rules.[9]

3.2. Explanation of development in the model

Fig. 3 shows three main factors underlying the behavior and development of the ACT-R model. The first factor, the mechanisms, is a combination of the ACT-R architecture and task-general knowledge. ACT-R provides the basis for behavior (e.g., the execution of production rules, the selection of chunks) and development (e.g., composition of new production rules, updating of the quantitative parameters that express the utility of production rules). The task-independent part of the mechanisms contains the interpretive production rules and the declarative representations of actions. These actions represent the strategy "answer a balance scale problem by searching for differences".

However, this does not specify the property to which the model should attend. This is specified by the second factor: the task-specific concepts. The availability of a concept is

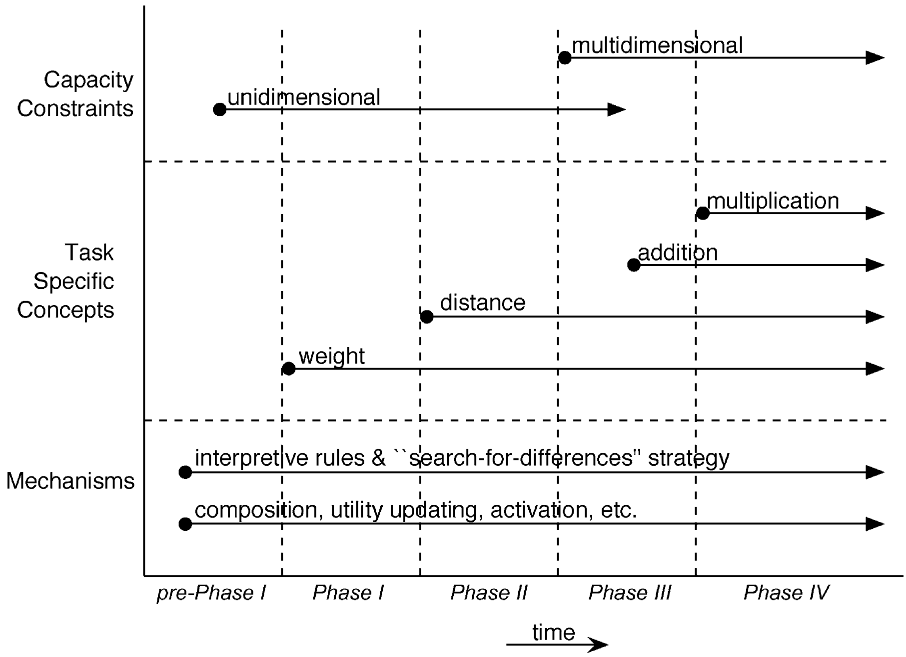

Fig. 3. Availability of components of the model over time.

mediated by the activation of the declarative chunk representing that concept. If this chunk is above the retrieval threshold, the related concept can be incorporated in the decision process. We make two, rather uncontroversial, assumptions about the second factor. The first is based on the empirical observation (Siegler, 1976; Siegler & Chen, 1998) that children initially prefer weight above distance. We simply assume an initially higher activation for weight than distance[10] on the basis of the encoding studies of Siegler (1976). The second assumption states that the concepts of addition and multiplication appear later in development. The exact timing of the appearance is of less importance since the emergence of the Addition Rule and Rule IV also depend on the capacity constraints.

Capacity constraints form the third factor. In many developmental theories, notably the Neo-Piagetian theories, the increase in capacity plays a central role in the development of domain-specific strategies (Case, 1985; Pascual-Leone, 1970). Although it has proven to be difficult to quantify the precise effects of cognitive capacity (Chapman, 1990), it is quite clear that its increase plays an important role in the transition from uni- to multidimensional thinking (Siegler, 1996). It is this specific consequence of the capacity constraint that we use to explain the generalization of the "search for differences" strategy. Initially, capacity limitations constrain the strategy to "search for a difference," and only later is the strategy extended to "search for more than one difference".

The development of performance depends on these three factors. The mechanisms explain how development occurs, the other two factors explain the order and timing of development.

From its initialization, the model has access to declarative knowledge about how "forced choice problems" (e.g., answer either "tip to left", "tip to right" or "remain in balance") are to be solved. This declarative knowledge represents the strategy that, in order to solve a force-choice problem, a property has to be found with respect to which the alternatives differ. In the context of the balance scale this means that it should look for a difference between the left- and right-hand side of the balance scale. If a difference is found based on the interpretation of the declarative actions, the model uses this difference to determine an answer. However, if no difference is found, the model searches for a new property.

Initially, neither the weight nor the distance property is encoded because the activation levels of the chunks representing the property are below threshold. Therefore, the model generates an answer by guessing. After each guess, feedback is given about the correctness of the given answer. As the proportion of correct answers based on guessing is low, the production rules representing this strategy have a low utility. Therefore, as soon as the weight property becomes available, the model will start to incorporate this concept in the decision process, yielding Rule I behavior. As this increases the proportion of correct answers, the production rules associated with Rule I are preferred over the pre-Rule I production rules. As the proportion of successful answers is still relatively low, distance is incorporated shortly after it becomes available to the model. However, as the available capacity is insufficient to incorporate both weight and distance in the decision process at the same time, the model switches its attention to distance, discarding the weight information. As determining the answers based solely on weights is less successful if the weights are equal than if the weights are unequal, the shift from weights to distances only occurs if the weights are equal. When the capacity has increased sufficiently to make it possible to use a strategy that incorporates both weights and distances, the model progresses to Rule III behavior. In this Phase, the weights and distances are examined regardless of the weights being equal or unequal. However, as knowledge to combine the weights and distances is as yet unavailable, the model can only guess the answer if the weights and distances are both unequal. Only when the concept of addition or multiplication becomes available is the model able to progress to the Addition Rule or Rule IV.

3.3. Simulation I: Phase-by-phase developmental progression

In this section we present the model in more detail and discuss how it simulates development. In discussing the model, we will refer to Table 2 and Fig. 4. Table 2 presents the numbers and proportions of correctly answered problems for all possible balance scale configurations. The numbers presented in this Table are specific to a balance scale with a maximum of five weights and five distances on either arm. For Rule I and Rule II, Table 2 contains both the overall statistics and statistics broken down into the two different branches of these Rules (i.e., “weights equal” and “weights unequal”, see Fig. 1). As all items of a particular item type are answered using the same branch, the other branch does not apply for that item type. This is denoted by a full stop in Table 2.

Fig. 4 plots the utilities of key production rules as function of the number of balance scale problems presented to the model. As the random selection of training items influences the exact course of the utilities, the graph shows some variation over runs. However, the values shown are illustrative for a typical run of the model. The production rules in Fig. 4 are all constructed by the model on the basis of the declarative actions.

Table 2
Number of correct responses per combination of Rule and balance scale item type for a five weights times five distances balance scale configuration

	Balance scale types							Overall correct	
	B	*D*	*W*	CB	CD	CW	WD	Absolute	Proportion
Pre-Rule I (guess)	8.3[a]	33.3[a]	33.3[a]	8[a]	29.3[a]	29.3[a]	66.7[a]	208.3	0.33
Rule I	25	0	100	0	0	88	200	413	0.66
Weights equal[b]	25	0	–	–	–	–	–	25	0.20
Weights unequal[b]	–	–	100	0	0	88	200	388	0.78
Rule II	25	100	100	0	0	88	200	513	0.82
Weights equal[b]	25	100	–	–	–	–	–	125	1
Weights unequal[b]	–	–	100	0	0	88	200	388	0.78
Rule III[c]	25	100	100	8[a]	29.3[a]	29.3[a]	200	491.6	0.79
Addition Rule[d]	25	100	100	20	64	64	200	573	0.92
Rule IV	25	100	100	24	88	88	200	625	1
Number of items	25	100	100	24	88	88	200	625	

Note. Different balance scale configurations lead to different absolute *and* relative values. *B*: balance item, *D*: distance item, *W*: weight item, CB: conflict balance item, CD: conflict distance item, CW: conflict weight item, WD: weight/distance item. Divisions by three indicate guessing the answer.
[a] Assuming a 1/3 probability of a correct guess.
[b] 125 items have equal weights, 500 items have unequal weights.
[c] As described in Siegler (1976), e.g., “Muddle Through”.
[d] As described in Normandeau et al. (1989).

3.3.1. *Phase 0*

At the start of each presentation, a problem is randomly sampled from the set of all possible problems. For each problem, the model is presented the number of weights and the distance to the fulcrum for both sides. Initially, the model resorts to guessing because it cannot retrieve any properties (i.e., the activation level of the weight property is still below threshold) to use in the problem solving process. As can be seen in Table 2, the proportion of correct answers for the guess-strategy is low. Therefore, the utility of the production rule that implements the guessing (see Fig. 4, labeled A(g)) drops rapidly. Because of the low utility of this guess-rule, the interpretive production rules will often be selected.[11] Consequently, as soon as a new strategy to solve the problems becomes available, the model will solve balance scale problems using the new strategy.

3.3.2. *Phase I*

The general strategy is represented in five steps:

1. Retrieve a property on which the two sides of the balance can differ.
2. Encode the values of the specified property.
3. If the encoded values are unequal, then base the answer on the observed difference.
4. If the encoded values are equal, then search for a new property and if found, return to Step 2.

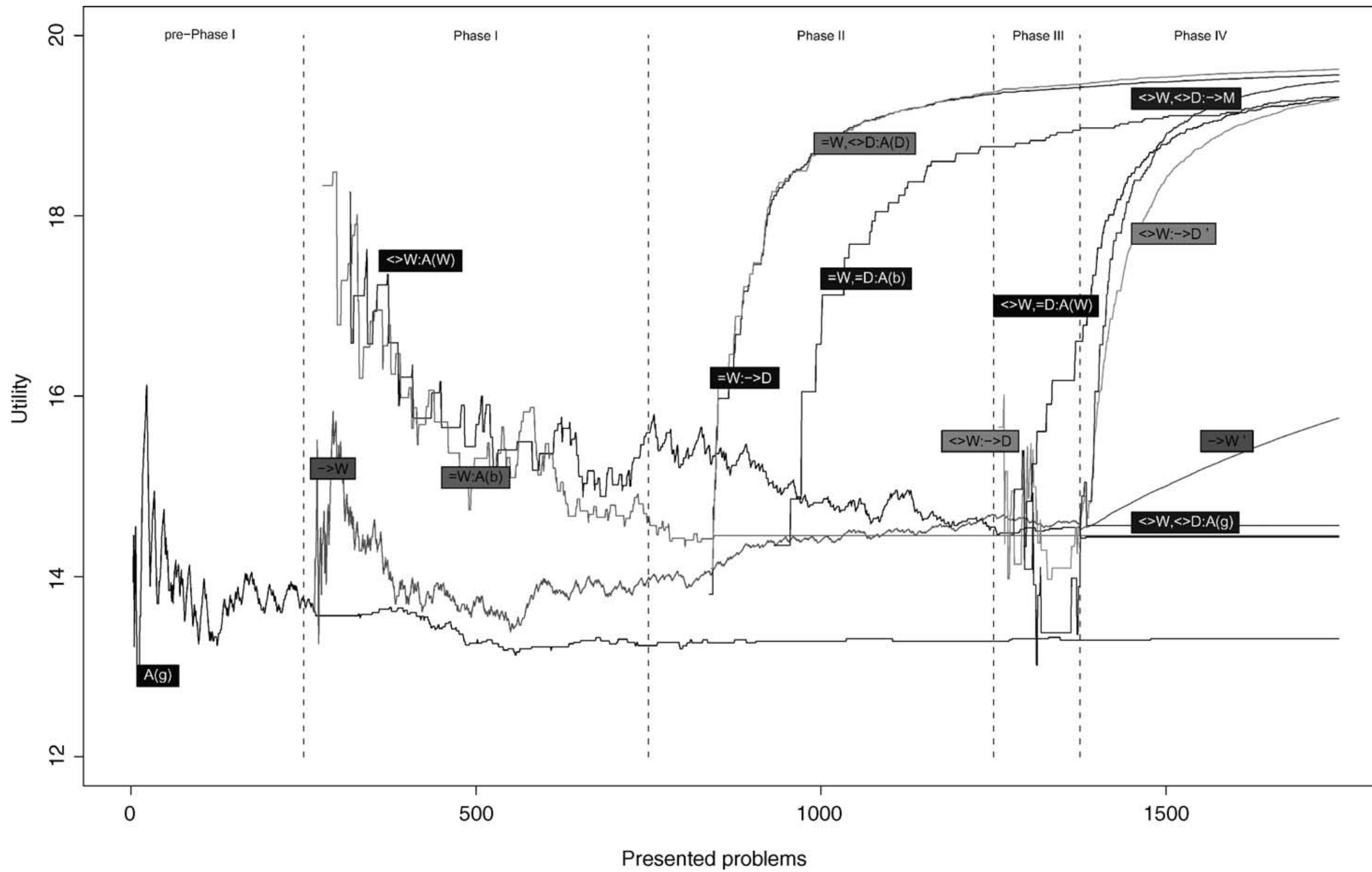

Fig. 4. Typical course of utility changes over time of selected production rules. The content of the production rules is represented in the labels, see the text for explanation. *Note*: To help track the production rules “->W” and “<>W->D” their labels have been repeated twice.

5. If no new property can be found, then render an answer based on the encoded values.

This general strategy first leads to Rule I. Step 1 determines the property on which the model will focus its attention. As initially the weight property is most active, the model will encode the weight values (Step 2) and use those to solve the balance scale problem. If the perceived values differ, Step 3 applies and an answer is determined on the basis of this difference (predicting a tilt to the side with the greater number of weights). If the perceived values are equal, the interpretation of the declarative knowledge leads to an attempt to retrieve a new property (Step 4). However, because the activation level of the distance property is too low, the model cannot retrieve it and consequently applies Step 5. As a result of the weights being equal, Step 5 renders "balance" as answer.

When the corresponding declarative actions are interpreted, the composition mechanism constructs new production rules. As discussed earlier, the composition process removes most retrievals from the constructed production rules, resulting in hard-coded values in the new production rules. Because of this, Step 1 is replaced by a production rule that immediately initiates a retrieval of the weight *values*, removing the requirement of having to retrieve a *property* first (same holds for Step 3). As the encoding of the values is based on perception instead of on a retrieval, this step is not removed by the composition mechanism. Because in this phase Step 4 always results in a failure to retrieve a new property, the composition mechanism will remove the retrieval request for a different property and, will instead directly answer "balance" (i.e., use Step 5) if the weights are equal.

The resulting production rules[12] are presented below. Before the model reaches the stage in which it only uses these production rules, it will have constructed numerous production rules that were only used once before being replaced by a new production rule. These replaced production rules remain available to the model. However, because these production rules (and the initial interpretive production rules) require more steps and more retrievals before arriving at an answer, these production rules are more expensive and therefore have a lower utility. As a result, the production rules associated with Rule I, as shown below, will prevail.

Production Rule: **Initialize** (->W)
IF the goal is to solve a balance scale problem
THEN request the encoding of the values of the weight property

Production Rule: **Initiate-Encoding**
IF the goal contains a request to encode the values of a property
THEN encode those values

Production Rule: **Test-Encoded-Values-a** (<>W:A(W))
IF the values of the weight property have been encoded
AND the values are unequal
THEN render as response the side with the largest value

Production Rule: **Test-Encoded-Values-b** (=W:A(b))
IF the values of a property have been encoded
AND the values are equal
THEN render as response "balance"

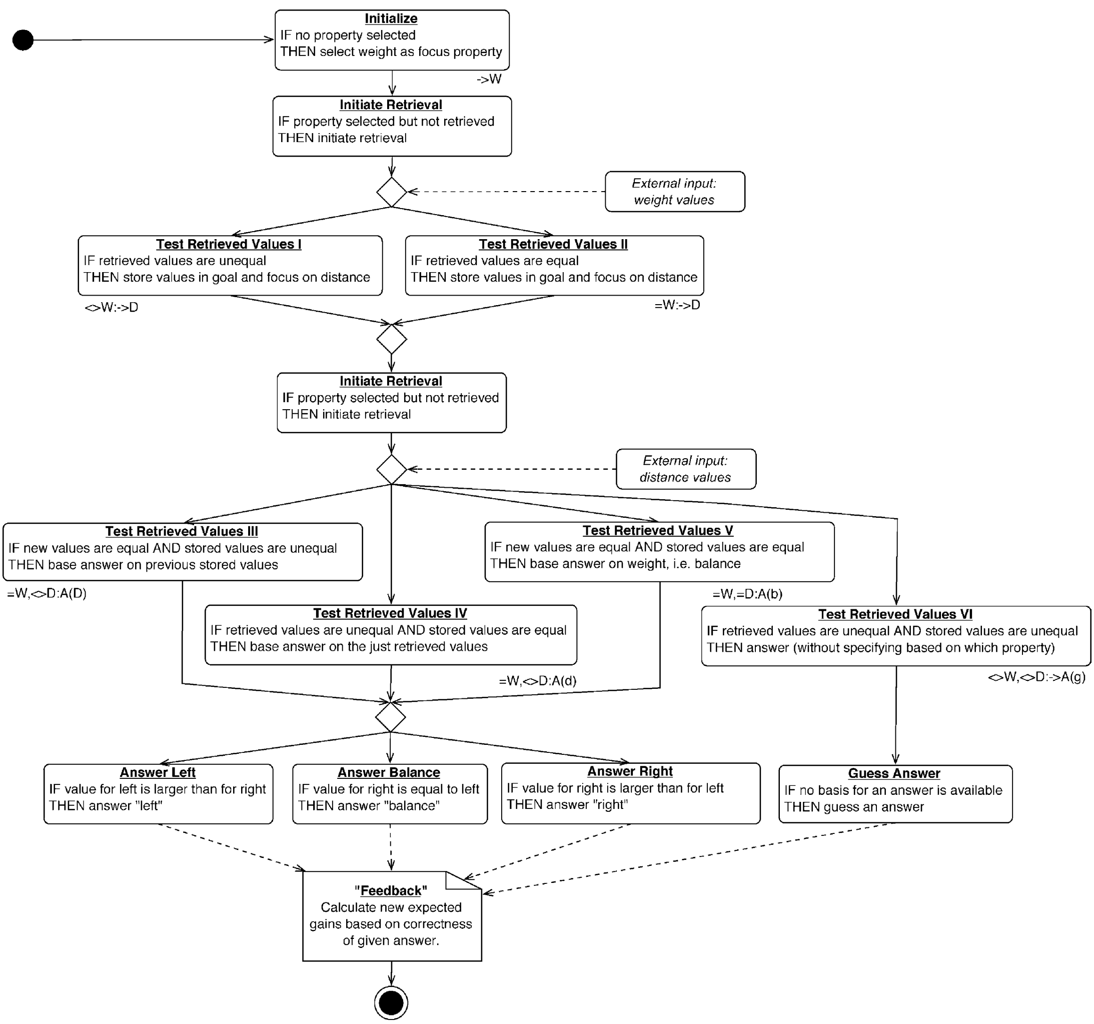

Fig. 5. Tree-like representation of the ACT-R model in Phase III. Below the boxes are the labels as used in Fig. 3 and the text.

Each production rule has been assigned a shorthand notation which is mentioned directly after the name of the production rule (e.g., "->W"). This shorthand notation is used both in the text and in Figs. 4 and 5. In this notation, "W" stands for weights, "D" for distances, "g" for guess, "b" for balance and "m" for multiplication. The operator "->" refers to "switch attention to", "=" to a condition requiring equal values for the property mentioned after the equal sign, "<>" to requiring unequal values and "A(x)" refers to "answer based on x". For example, "<>W,<>D:A(g)" refers to the production rule reflecting the following knowledge: if the weight and distance values are unequal, guess an answer.

The first production rule (->W) causes the model to consider the weight property. The second production rule initiates the encoding of the values of the weight property. These two steps are not merged because the perception based encoding step is not removed by the composition mechanism. Based on the encoded values, one out of two production rules matches. If the weights are equal, the production rule =W:A(b) renders "balance" as response, if the weights are unequal, the production rule <>W:A(W) renders as response the side with the largest number of weights. As can be seen in Table 2, if the answer is based on Rule I "weights equal" (which corresponds to the =W:A(b) production rule), the proportion of correct answers is lower than if the answer is based on Rule I "weights unequal" (<>W:A(W)). Therefore, the utility of the weights equal production rule (=W:A(b)) decreases faster than the utility of the weights unequal production rule (<>W:A(W)), as is visible in Fig. 4.

As the utility of a constructed production rule is based on the utilities of its parent production rules, newly composed production rules have different initial utilities. As the ->W production rule is based on production rules used in Phase 0, it starts with a low utility, whereas <>W:A(W) and =W:A(b) are not directly based on production rules with decreased utilities. Therefore, the utilities of these production rules are initially higher.

3.3.3. Phase II

Because of the decrease in utility of the =W:A(b) production rule, the interpretive production rules will eventually try to construct new strategies based on the declarative "search for a difference" action. The number of problems required for this to happen depends on the distribution of the balance problems. For example, if the proportion of weight items is large, as in the simulations of McClelland (McClelland, 1989, 1995), the production rules used in Rule I will encounter fewer failures and will remain active longer.

To find a new strategy, the model has to include a new property in the reasoning process. As long as the activation of the distance property is below the retrieval threshold, the model is in Phase I. Only when the distance property reaches a sufficient level of activation, is it incorporated in the answering strategy.[13] As soon as this level is reached, the composition mechanism starts to construct new production rules as it did at the start of Phase I. The resulting two most significant production rules are =W,<>D:A(D) and =W,=D:A(b). The first production rule determines the solution based on the distance values if the weights are equal. The second production rule renders the solution "remains in balance" if the values for both properties are equal. Since these production rules always yield the correct answer (see Table 2, Rule II, weights equal), the utility of these production rules increases asymptotically as the P value in $U = P \times G - C$ approaches 1 and the costs are constant. This increase in P also influences the utility of the ->W production rule, as this rule now gives rise to a greater proportion of correct answers.

3.3.4. Phase III

During Rule II behavior, the utility of <>W:A(W) decreases. This decrease will cause the interpretive production rules to be used occasionally. However, the available declarative knowledge in Phases I and II limits the "search for a difference" strategy to one unequal dimension. Therefore, as long as more advanced versions of this knowledge are unavailable, the model cannot progress to Rule III.

As the progression from Rule II to Rule III is associated with a shift from unidimensional to multidimensional reasoning (Case, 1985; Chapman, 1990; Pascual-Leone, 1970; Siegler, 1996), the declarative actions are modified to reflect this change. At about problem presentation 1250 in Fig. 4, the declarative representations are extended to allow for the retrieval of a new property even if the values of the first encoded property are unequal. This extension involves the modification of the original Step 3, as is shown below:

1. Retrieve a property on which the two sides of the balance can differ.
2. Encode the values of the specified property.
3a. If the first encoded values are unequal, then store the encoded values and search for a new property and return to Step 2.
3b. If both the first set of encoded values and the second set are unequal then search for and apply a method that uses both sets to render an answer.
4. If the encoded values are equal, then search for a new property and if found, return to Step 2.
5. If no new property can be found, then render an answer based on the encoded values.

Now, in case the weights are unequal, applying the interpretive production rules will lead the model to apply Step 3a. This results in the retrieval of distance as an additional property to take into account. If the encoded distance values are equal, the model will base the answer solely on the weight values as these are already known to be unequal. However, if the distance values are unequal, the model searches for a method to combine both weights and distances. If a method cannot be found, the model will resort to guessing. When the composition process is complete, i.e., when no new production rules can be created, the model solves balance scale problems using the production rules as shown in Fig. 5.

The guessing production rule is incorrect in two thirds of all cases. As a consequence, its utility decreases to a relatively low level. Therefore, this phase is relatively unstable: if too many incorrect answers are registered in succession, behavior will temporarily regress to Rule II (cf., Jansen & Van der Maas, in press, Table 7). In Fig. 4, this is reflected in the continuation of updating of the utility of the <>W:A(W) production rule. This decrease in utilities also causes the model to regularly test for other methods to combine the unequal values of weights and distances.

3.3.5. Phase IV

As soon as the concepts of addition or multiplication become available, these are incorporated in the decision process (Step 3b), thereby replacing the guess production rules. In Fig. 4, the multiplication method (<>W,<>D:A(m)) is used from about problem 1,400 onward. As a side effect, it is only after the multiplication rule has become available, that the <>W->D production rule wins the utility competition over <>W:A(W). This is visible in Fig. 4 by comparing the utilities of <>W:A(W) and <>W->D. During Phase III, their utilities are relatively equal, only after the transition to Rule IV, the utility of <>W->D increases sharply. A similar effect is visible for ->W (label: ->W′). As this production rule is never involved in any incorrect answer in Phase IV, its utility increases consistently. However, because of the relative large number of incorrect answers ->W was involved in previously, its utility increases more gradually.

As can be seen in Table 2, the Addition Rule gives a correct answer to most balance scale problems. Therefore, its utility will not decrease below the utility level of the interpretive production rules. Only if the model is presented a sequence of items that are answered incorrectly using the Addition Rule, the utility of this production rule drops below the utility of the interpretive production rules. Only in this situation, the model can progress to Rule IV. This is consistent with the infrequency of a spontaneous improvement from the Addition Rule to Rule IV.

Note that even when the multiplication method is available, problems that can be solved with simpler comparisons of values will still be solved by searching for differences. This is due to the high utility of the associated production rules. Therefore, the interpretive production rules cannot interfere because the interpretive production rule's utility is too low compared to the utility of the comparison production rules. Although a different explanation of the sparse use of the multiplication method could refer to the relative high cost of multiplication, this model does not depend on this assumption.

3.4. Simulation II: Transitions without feedback and the Torque Difference Effect

As was shown by Jansen and Van der Maas (2002), transitions from Rule I to Rule II may also take place without feedback. They describe three different types of transition patterns (see Fig. 2). The Maxwell convention pattern obtains when the improvement to Rule II occurs at the same distance difference as the regression to Rule I. The delay convention pattern obtains when the improvement to Rule II occurs at a higher distance difference than the regression to Rule I. Finally, the sudden jump pattern obtains when after the progress to Rule II, the child never regresses to Rule I. In the present section, we will illustrate how transitions without feedback as well as the observed transition patterns can be explained by the model.

We presented the model with the same sequence of problems as in the experiments of Jansen and Van der Maas. Like Jansen and Van der Maas, we provided no feedback about the correctness of the given answers. Without feedback, the utilities of production rules do not change. As the activation of chunks is based solely on the number of retrievals, irrespective of the success or failure of the retrieving production rules, the activations *are* updated. Therefore, the only source of changes during the presentations of problems without feedback is the activation of declarative knowledge.

The manipulated variable in the experiments of Jansen and Van der Maas is the difference in distances from the fulcrum between the left and right sides of the balance scale. A number of consecutive balance problems with equal weight configurations, but with different distance configurations, were presented to children. In the first half of the problems, the difference in distance between left and right was increased by one distance unit per problem. In the second half, the distance was decreased by one distance unit per problem. We assumed that this step-by-step increase in difference would increase the perceptual saliency of that property. This additional saliency can be represented in the ACT-R model as an increased activation of the distance property. To achieve this, the activation formula

$$\text{Activation} = B = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$$

was extended to: Activation $= B +$ saliency. The saliency is calculated by simply scaling the distance difference (saliency $= \Delta\text{distance}/c$, where c is the scaling constant). Note that the default ACT-R activation formula contains a term reflecting noise. This simulation is run without noise as noise would not provide different results in this simulation.

To progress to Rule II, the model has to apply the interpretive production rules at least occasionally. As long as the distance property remains below the retrieval threshold, performance is consistent with Rule I. However, the saliency associated with extreme distance differences can increase the activation of the distance property to a level above the retrieval threshold. If the distance becomes sufficiently activated, the distance property is retrieved and the distance values are used in determining a solution. If this happens before the maximum distance difference is reached, the next problem is also solved using the distance property, as the previous presentation increased the base level of the distance chunk's activation (B) and the increased distance difference also increased the activation. When the maximum difference is reached, the next presented problem has a decreased distance difference and therefore a lower additional saliency activation. What happens at this point depends on the interplay between increase in base-level activation and the decrease in saliency activation. If the base level was increased more than the decrease in saliency activation, the model is still able to retrieve the distance property. This results in continued Rule II based behavior. However, if the increase in B is smaller than the current decrease in saliency activation, the model is not able to retrieve the distance property, and therefore regresses to Rule I.

The results of the simulation are presented in Fig. 6. The solid lines represent the activation of the distance chunk. When the total activation of this chunk is above the threshold which is depicted by the horizontal dotted line at $Y = 0$, performance is in accordance with Rule II. The performance, given the current activation, is shown as small circles, either at the bottom of the graph (denoting Rule I use), or at the top of the graph (denoting Rule II use).

If during Rule II use the increase in the base level is small compared to the differences in saliency activation, the regression to Rule I behavior takes place at the same distance difference as the change to Rule II behavior. The phenomenon that the change takes place at the same point is referred to as the Maxwell convention (see Fig. 6a). If the increase in the base level is large, the activation without the saliency component might be large enough to ensure the activation is above the retrieval threshold (the sudden jump pattern, Fig. 6c). If the base-level activation is raised slightly, but not sufficiently to be retrieved in the absence of additional saliency activation, the delay convention pattern appears (Fig. 6b). These differences in activation effects can be explained by, for example, differences in original base levels or by differences in levels of elaboration of declarative chunks.

Although these explanations of the transition patterns are formulated in terms of the activation of the distance property, the composition process can also explain the sudden jump pattern. During the increased activation period, the composition process starts to construct new production rules. If the composition process is complete, the resulting production rules are completely specialized. That is, all retrievals (except for the encoding of the actual values of weights and distances) are removed from these production rules. Because the explicit search for the distance property is removed from the production rules, the activation of the distance property chunk does not determine the behavior of the model. Thus, giving rise to a sudden jump.

The saliency of the distance property, modeled as an additional activation of the corresponding chunk, also explains the Torque Difference Effect. As torque difference is highly correlated with distance difference, items with large torque differences will on average cause an increased

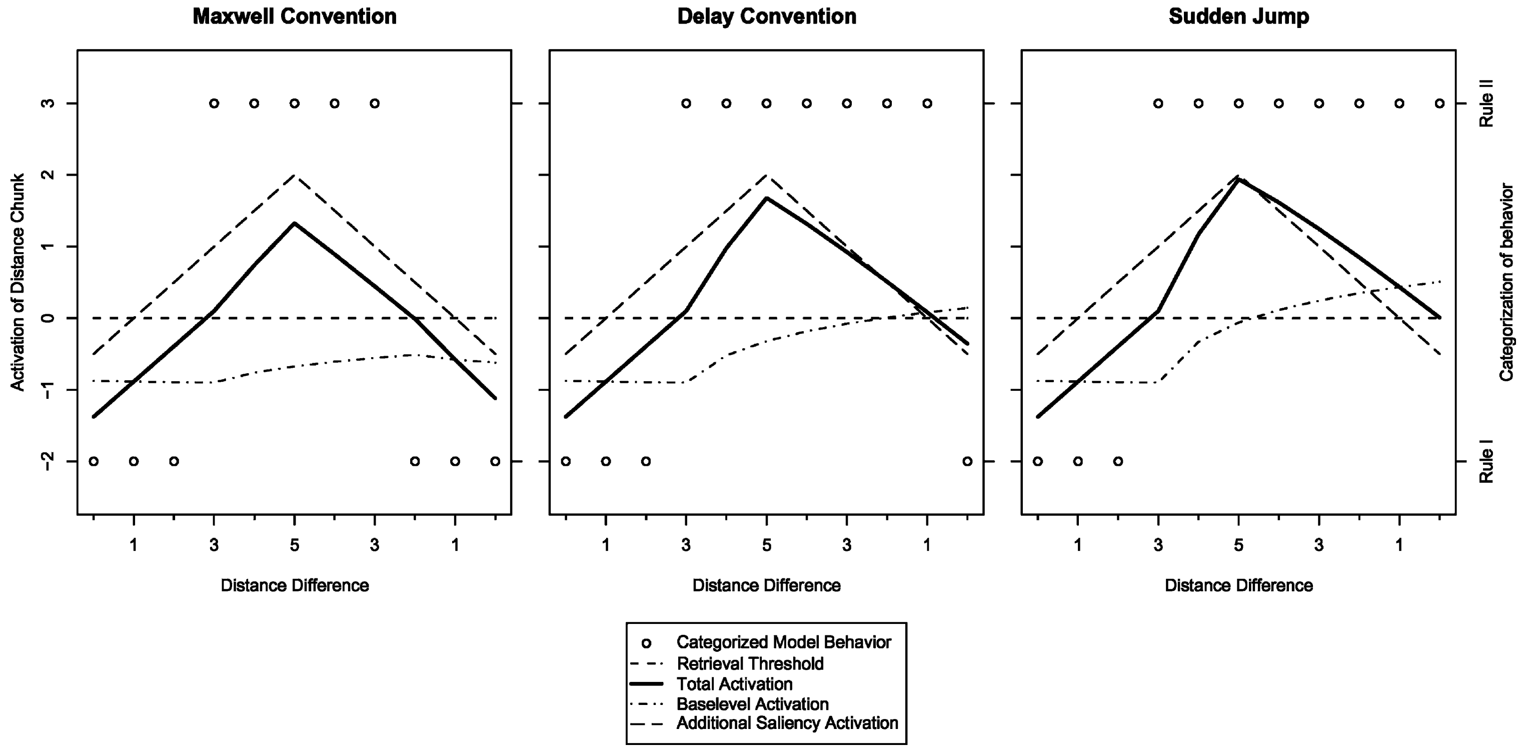

Fig. 6. Simulation of no-feedback transition patterns.

activation for the distance chunk. This increases the probability that distance will be included when the declarative actions initiate a new search for a difference. However, this will *only* apply if the utility of the interpretive production rules and the utility of the rules that currently solve the problem are similar. In other words, the torque difference only influences the model's behavior in the instable period just before transitions. This more restricted Torque Difference Effect is in accordance with the reanalysis of the Torque Difference Effect (Jansen & Van der Maas, 1997).

In this second set of simulations, the explanation of developmental phenomena is based on the influence of the additional saliency based activation. In the first set of simulations, saliency activation was disabled. However, no qualitative change in model performance would occur if saliency was added to the model used in the first set of simulations. The effect would be limited to minor modifications in the timing of Rule transitions. As this would complicate matters unnecessarily, we decided not to incorporate the notion of saliency in the first set of simulations.

4. Discussion

In this paper we presented a model of balance scale behavior that is based on three components: architectural and task-independent mechanisms, task-specific concepts, and constraints related to cognitive capacity. Earlier, we discussed the merits of previous models of balance scale behavior in terms of four empirical criteria. We now turn to a discussion of the new model in terms of the empirical criteria and subsequently we outline the relation between our model and the previous models.

4.1. Empirical criteria

4.1.1. EC1: Rule-like behavior

The first empirical criterion is the presence of Rules (i.e., stable performance) and transitions between these Rules. As demonstrated in Simulation 1, the ACT-R model shows stable performance when the composition process has constructed a fixed set of production rules to solve balance scale problems. As long as the utility of the production rules is higher than that of the interpretive production rules, no new production rules are constructed. But even if the interpretive production rules are applied, no new rules can be constructed in the absence of new knowledge for the construction of better rules. In both situations, behavior is stable. However, as soon as the utility of the composed production rules drops below the utility of the interpretive production rules *and* sufficient knowledge is available, new production rules are composed. This changes the way in which the model responds. At this point several rules have comparable utilities: the old production rules, the interpretive production rules, and the newly composed production rules. This makes behavior at this point highly dependent on factors like noise, the distribution and type of presented balance scale problems, and the visual saliency of task dimensions.

In short, the Rule-like behavior is explained by architectural mechanisms: The stable periods associated with Rule-like behavior are explained by the stabilizing effects of the adjustments of the utility and activation of available knowledge as specified by ACT-R. Whereas the transitions to new Rules depend on the ACT-R mechanisms to construct new production rules. With different task-specific knowledge, the framework of the model can be applied to different reasoning tasks that are associated with Rule-like behavior.

4.1.2. EC2: Rule sets

A model of balance scale behavior should reproduce the four distinct Rules as originally presented by Siegler (1976), and the later identified Addition Rule (e.g., Normandeau et al., 1989). As discussed earlier, our model reproduces the Rules identified by Siegler and is also able to perform according to the Addition Rule. The Addition Rule is not easily replaced by Rule IV because the Addition Rule results in the correct responses to most items. The model progresses to Rule IV only if it is presented with a carefully selected set of balance scale problems that are answered incorrectly with the Addition Rule. This is consistent with the observation that very few children progress to Rule IV spontaneously.

The order of transitions matches the empirical observed sequence. This is based on constraining the availability of task-specific concepts and the assumption of processing limitations related to cognitive capacity. For example, only when the weight concept is available, is the model able to progress from pre-Rule I to Rule I; only when the concept of multiplication becomes available, can the model progress to Rule IV, etc. With respect to the cognitive capacity, as long as the available capacity constrains the model to the use of a single property (i.e., either weight or distance), the model cannot proceed to Rule III.

As discussed earlier, the assumptions with respect to the initial unavailability of task-specific concepts and the capacity constraints are in line with empirical evidence.

4.1.3. EC3: Transition patterns without feedback

Transitions from Rule I to Rule II can occur when children are presented with a sequence of distance problems with increasing distance differences, even in the absence of any feedback about the correctness of their responses. As we have shown in Simulation 2, the ACT-R model explains this phenomenon by incorporating visual saliency in the activation of chunks. The larger the difference between the values of a property for the left and right side of the balance scale, the larger the additional activation of the chunk that represents that property. When the distance difference is decreased again, the activation of a chunk might drop below the retrieval threshold, making that property unavailable to the answering process. Given individual differences in both activation updating and production rule composition, the three types of transition patterns emerge (i.e., the Maxwell convention, the delay pattern, and the sudden jump pattern).

However, feedback still plays an important role in the presented model. That is, only by the feedback-driven updating of the production rules utilities, the model can develop to a situation in which a transition without feedback can take place. Although feedback is not necessary for the actual transitions, a model would not progress from the beginning of a phase to the end of that phase without feedback.

4.1.4. EC4: Torque Difference Effect for large torque differences

As the transition patterns are directly related to the distance difference and therefore to the torque difference, our model shows torque difference effects. However, our model does predict that the Torque Difference Effect will not be strong enough to give rise to an improvement at all times. That is, the Torque Difference Effect can only influence the behavior of the model in the vicinity of transitions.

Summarizing, we have presented a model that passes the four empirical phenomena associated with balance scale learning. In the construction and description of the model, care was taken to clearly state all necessary assumptions and to have all assumptions supported by empirical evidence.

4.2. Comparisons and predictions

The successful reproduction of the empirical phenomena by the presented model was partly realized with features that were also used in previous models. Like the symbolic models, behavior in the ACT-R model is based on the application of production rules, which results in rule-like behavior. As in the previous models, new production rules are learned by extending the already present knowledge. However, instead of containing a few complex production rules (like the complete trees presented in Fig. 1), our model consists of a larger number of smaller production rules. Each of these production rules performs only a small part of the complete answering process. Therefore, newly constructed production rules can simply replace older production rules instead of requiring a complex mechanism to modify existing production rules.

As in the neural net models of balance scale behavior, quantitative information plays an important role in the ACT-R model (i.e., utility and activation). The dynamics of these quantitative variables are important for the description of the empirical phenomena.

The combination of features from ACT-R and the symbolic and neural net type of models provides the basis for a number of achievements specific for this model:

(1) The model produces the relatively abrupt transitions, which are problematic in the non-symbolic models.
(2) It explains transitions without feedback and the related transition patterns, which cannot be explained from the learning methods used in the previous models.
(3) The model demonstrates that, given a non-biased training set, Rule IV will not easily be learned because of the high success-rate of the Addition Rule.
(4) The presented model is able to explain both phenomena related to long-term development and phenomena which are only observable during short time-spans (cf., Anderson, 2002).
(5) The model makes explicit that the notion of "search for differences" combined with a gradual increase in capacity and knowledge is sufficient to explain development on the balance scale task.

However, the most distinct feature of our model is the parsimony of its main assumption: Children who are solving balance scale problems are explicitly looking for differences between the left and right side of the balance scale.

The model also makes novel predictions about behavior on the balance scale task. During pre-Rule III behavior, the model is not sensitive to (multiplication) instruction on the balance scale task. Only when the model's behavior is in accordance with Rule III, the model is able to benefit from multiplication instruction. Similar reasoning holds during other Phases, for example, the model predicts that instruction emphasizing the importance of distance during stable Rule I usage does not have an effect on behavior.

The model predicts an important role for visual saliency, especially during transitions. An example of which is the explanation of the Torque Difference Effect as this effects are explained on the basis of visual saliency. Because effects of visual saliency can only influence the behavior of the model during the less stable periods around transitions, the model predicts that the Torque Difference Effect is limited to these periods.

Although not discussed in this paper, the model can also be used to predict reaction time patterns. As ACT-R specifies the amount of time necessary for the steps in the answering process, reaction time patterns can be easily derived from the model. For example, during and just after transitions, the model predicts increased reaction times as the model is still in the process of constructing the optimal set of production rules for the new phase. During the process of constructing this optimal set, additional production rules are used, causing an increase in reaction times. Moreover, previous work on arithmetic skills (e.g., Lebiere & Anderson, 1998) might be incorporated into the model to use analyses of reaction times to shed light on what strategies are used during the unstable Phase III.

Notes

1. Given a five weights and five distances at each side balance scale, 68% of all possible items fall in these six items types, see Table 2. The remaining items have more weights at the side with the larger distance, and are therefore less suitable to categorize behavior. We will refer to these items as weight/distance items.
2. To ease presentation, the Rules empirically identified by Siegler and others, see EP2, are written with a uppercase “R”, whereas ACT-R production rules, to be introduced later, are written with a lowercase “r”. A single Rule is always implemented as a set of production rules.
3. Latent class models consist of classes that are defined by response patterns associated with the item types. Latent class models are fitted by determining the relative size of each of the classes (McCutcheon, 1987). Main advantages over the standard rule assessment method are (a) statistical methods to evaluate the fit of the Rules to the data are available, (b) it is not necessary (although possible) to specify the Rules *a priori*, so that new Rules can be detected, and (c) different balance scale tests with different numbers of items can be compared.
4. Some models directly implement the Rules as specified in Fig. 1. Classification by latent class analysis is superfluous in this case.
5. An often-cited production rule model of balance scale behavior is a model in SOAR (Newell, 1990, p. 465 ff). However, the description of this model does not provide enough information to evaluate it.
6. Latent class analysis has yet to be applied to the data generated by this model.
7. ACT-R has previously been applied to cognitive development. Taatgen (1999) focussed on the compilation and usage of different production rules, Lebiere and Anderson (1998) focussed on the acquisition and tuning of declarative knowledge, Jones, Ritter, and

Wood (2000) focussed on capacity related constraints and Taatgen and Anderson (2002) focused on past-tense learning.

8. This addition is not part of the ACT-R version as described in Anderson and Lebiere (1998), but is incorporated in a new version (ACT-R v5.0, 2002a). Although the version of ACT-R as described in Anderson and Lebiere (1998) is also able to construct new production rules, the new composition mechanism constructs new rules using a more principled method.
9. For links to an extended discussion of composition and learning from instruction, see the ACT-R v5.0 website: http://act-r.psy.cmu.edu/software/.
10. Note that the ACT-R model would also work given an initially higher activation for distances, i.e., it does not explain the preference of weight above distance.
11. Given the general nature of the interpretive production rules, the usage of these production rules is not limited to the balance scale domain. Because of this extensive "external" usage, the parameters will not be influenced significantly by the application of the production rules in the balance scale domain. Therefore, we assigned a constant utility of approximately 13.3 (+noise) to these production rules. This value is based on a P value of 2/3 and a relative low C value, but other values would not influence the general behavior of the model.
12. The production rules presented here have been edited for reasons of readability. All code necessary to run the model is available at http://www.van-rijn.org/hedderik/bstm/.
13. A discussion of how this threshold is reached is outside the scope of this paper, but both developmental changes in activation or threshold parameters, and exposure to tasks in which distance is an important predictor might play a role.

Acknowledgments

This research was conducted while the first author was supported by a grant of the Social Science Research Council of The Netherlands Organization for Scientific Research (MaG-NWO: 575-22-004). The authors would like to thank Conor Dolan, Casper Hulshof, Brenda Jansen, Pat Langley, Maartje Raijmakers, Niels Taatgen, and an anonymous reviewer for valuable comments on earlier versions of this paper.

References

ACT-R v5.0 (2002a) (n.d.). Retrieved October 5, 2002, from "ACT-R 5.0 Beta" Web site: http://act.psy.cmu.edu/ACT-R_5.0/.

ACT-R 5.0 (2002b). Unit 9: Production Rule Learning (n.d.). Retrieved January 5, 2002, from "ACT-R 5.0 Beta" Web site: http://act.psy.cmu.edu/ACT-R_5.0/unit9/.

Anderson, J. R. (1987). Methodologies for studying human knowledge. *Behavioral and Brain Sciences*, *10*(3), 467–505.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (2000). Learning from instructions. In *Proceedings of the Seventh Annual ACT-R Workshop*, Cargnegy Mellon University, Pittsburgh, PA.

Anderson, J. R. (2002). Spanning seven orders of magnitude: A challenge for cognitive development. *Cognitive Science*, *26*, 85–112.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Boden, M. A. (1979). *Piaget*. Glasgow, UK: William Collins Sons & Co. Ltd.

Case, R. (1985). *Intellectual development: Birth to adulthood*. London: Academic Press.

Chapman, M. (1990). Cognitive development and the growth of capacity: Issues in Neo-Piagetian theory. In J. T. Enns (Ed.), *The development of attention: Research and theory* (pp. 289–334). Amsterdam, The Netherlands: Elsevier.

Chlestos, P., De Lisi, R., Turner, G., & McGillicuddy-De Lisi, A. V. (1989). Cognitive assessment of proportional reasoning strategies. *Journal of Research and Developmental Education*, *23*, 18–27.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, *120*, 235–253.

Ferretti, R. P., & Butterfield, E. C. (1992). Intelligence-related differences in the learning, maintenance, and transfer of problem-solving strategies. *Intelligence*, *16*(2), 207–223.

Ferretti, R. P., Butterfield, E. C., Cah, A., & Kerkman, D. (1985). The classification of children's knowledge: Development on the balance-scale and inclined-plane tasks. *Journal of Experimental Child Psychology*, *39*(1), 131–160.

Jansen, B. R. J., & Van der Maas, H. L. J. (1997). Statistical test of the rule assessment methodology by latent class analysis. *Developmental Review*, *17*, 321–357.

Jansen, B. R. J., & Van der Maas, H. L. J. (2001). Evidence for the phase transition from Rule I to Rule II on the balance scale task. *Developmental Review*, *21*, 450–494.

Jansen, B. R. J., & Van der Maas, H. L. J. (2002). The development of childrens' rule use on the balance scale task. *Journal of Experimental Child Psychology*, *81*(4), 383–416.

Jones, G., Ritter, F. E., & Wood, D. J. (2000). Using a cognitive architecture to examine what develops. *Psychological Science*, *11*, 1–8.

Klahr, D., & Siegler, R. (1978). The representation of children's knowledge. In H. Reese & L. Lipsitt (Eds.), *Advances in child development and behavior* (pp. 61–116). New York, NY: Academic Press.

Langley, P. (1987). A general theory of discrimination learning. In D. Klahr, P. Langley, & R. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.

Lebiere, C., & Anderson, J. R. (1998). Cognitive arithmetic. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 297–342). Mahwah, NJ: Lawrence Erlbaum Associates.

McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In M. G. M. Morris (Ed.), *Parallel distributed processing, implications for psychology and neurobiology* (pp. 8–45). Oxford: Clarendon Press.

McClelland, J. L. (1995). A connectionist perspective on knowledge and development. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 157–204). Hillsdale, NJ: Lawrence Erlbaum Associates.

McCutcheon, A. L. (1987). *Latent class analysis*. Beverly Hills, CA: Sage Publications.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Normandeau, S., Larivee, S., Roulin, J. L., & Longeot, F. (1989). The balance-scale dilemma: Either the subject or the experimenter muddles through. *Journal of Genetic Psychology*, *150*(3), 237–250.

Pascual-Leone, J. (1970). A mathematical model for the transition rule in Piaget's development stages. *Acta Psychologica*, *32*, 301–345.

Pinker, S. (1984). *Language learnability and language development*. Cambridge, MA: Harvard University Press.

Raijmakers, M. E. J., Van Koten, S., & Molenaar, P. C. M. (1996). On the validity of simulating stagewise development by means of PDP networks: Application of catastrophe analysis and an experimental test of rule-like network performance. *Cognitive Science*, *20*(1), 101–136.

Reese, H. W. (1989). Rules and rule-governance: Cognitive and behavioristic views. In S. C. Hayes (Eds.), *Rule governed behavior: Cognition, contingencies, and instructional control* (pp. 3–84). New York: Plenum Press.

Sage, S., & Langley, P. (1983). Modeling cognitive development on the balance scale task. In A. Bundy (Ed.), *Proceedings of the Eight International Joint Conference on Artificial Intelligence* (Vol. 1, pp. 94–96), Karlsruhe, Germany.

Schmidt, W. C., & Ling, C. X. (1996). A decision-tree model of balance scale development. *Machine Learning*, *24*, 203–230.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning*, *16*, 57–86.

Shultz, T. R., & Schmidt, W. C. (1991). A cascade-correlation model of balance scale phenomena. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 635–640). Hillsdale, NJ: Erlbaum.

Shultz, T. R., Schmidt, W. C., Buckingham, D. & Mareschal, D. (1995). Modeling cognitive development with a generative connectionist algorithm. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 205–261). Hillsdale, NJ: Lawrence Erlbaum Associates.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology*, *8*, 481–520.

Siegler, R. S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development*, *46*(2), 1–74. (With commentary and reply from Sidney Strauss and Iris Levin.)

Siegler, R. S. (1989). Mechanisms of cognitive change. *Annual Review of Psychology*, *40*, 353–379.

Siegler, R. S. (1996). Unidimensional thinking, multidimensional thinking, and characteristic tendencies of thought. In A. Sameroff & M. Haith (Eds.), *Reason and responsibility: The passage through childhood* (pp. 63–84). Chicago, IL: University of Chicago Press.

Siegler, R. S., & Chen, Z. (1998). Developmental differences in rule learning: A microgenetic analysis. *Cognitive Psychology*, *36*(3), 273–310.

Taatgen, N. A. (1999). Explicit learning in ACT-R. In U. Schmid, J. Krems, & F. Wysotki (Eds.), *Mind modelling: A cognitive science approach to reasoning, learning and discovery* (pp. 233–252). Berlin: Science Publishers.

Taatgen, N. A. (2000). *Proceduralization in ACT-R* (Tech. Rep.). Groningen University, The Netherlands. (Retrieved June 13, 2001, from World Wide Web: http://tcw2.ppsw.rug.nl/niels/proceduralization/proceduralization.doc.)

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? A model of learning the past tense without feedback. *Cognition*, *86*(2), 123–155.