Simulating Spatial Reasoning Using ACT-R

Jona Boeddinghaus (jona@informatik.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee
79110 Freiburg, Germany
Marco Ragni (ragni@informatik.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee
79110 Freiburg, Germany
Markus Knauff (markus.knauff@cognition.iig.uni-freiburg.de)
Center for Cognitive Science, Friedrichstraße 50
79098 Freiburg, Germany
Bernhard Nebel (nebel@informatik.uni-freiburg.de)
Department of Computer Science, Georges-Köhler-Allee
79110 Freiburg, Germany

Abstract

We present an ACT-R model of spatial reasoning based on the SRM model (**S**patial **R**easoning by **M**odels). This model maps spatial working memory to a two-dimensional array and uses a spatial focus to place objects in the array, manipulate the position of objects, and inspect the array to find spatial relations that are not given in the premises. Since the SRM explains many experimental findings only on a qualitative level, we implemented it into an ACT-R model. Not only does the model show some well-known effects in spatial reasoning and offers a good insight into the processes in the SRM model, but in addition it also allows us to predict reasoning times. The Model is accessible through a Java interface, which can be found and run from the following website http://www.informatik.uni-freiburg.de/~srm.

Introduction

Spatial reasoning is fundamental for the human species. It is not only important for navigating through small or large-scale environments, but also for navigation through the internet. It describes the deduction process of individuals for a given set of premises consisting of spatial relations. Consider, for example, the binary spatial relations

The hammer is to the right of the pliers
The screwdriver is to the left of the pliers
The wrench is in front of the screwdriver
The saw is in front of the pliers

and the question "which relation holds between the wrench and the saw?", one can conclude that "the wrench is to the left of the saw". The latter sentence is accordingly called the *conclusion* while the former four sentences are the *premises*. These reasoning processes can be accomplished by applying formal rules of inference to the linguistic representation of the premises or – based on the mental model theory (MMT) (Johnson-Laird & Byrne, 1991; Johnson-Laird, 2001) – by constructing and inspecting a spatial array representing the relations of the objects to each other as described in the premises. Here an ACT-R 5.0 model is presented. It simulates spatial reasoning by using the MMT. While we explain the models architecture, the specific implementation of such a spatial array that acts as the spatial working memory will be the main focus.

The SRM Model

The SRM model (Spatial Reasoning by Models), which this ACT-R simulation is based on (Ragni, Knauff, & Nebel, 2005), provides a view of how mental models of spatial relations are constructed from premises and offers an easy to apply complexity measure that fits many experimental findings. There are two basic assumptions from which the grounds for the SRM model are derived.

First the spatial working memory containing the representation of the premises as objects with their relations is conceptualized as a spatial array. In this spatial array the spatial information is represented only in relational – not metrical – terms, thus following the line of spatial representations (Schlieder, 1995) and rejecting concepts of visual mental images. Binary spatial relations are defined as a triplet (X, r, Y) in which

X is the referent,
r is a binary relation, and
Y is the relatum.

X is called the "to be located object" (LO) and Y the "reference object" (RO) (Miller & Johnson-Laird, 1976). As relations "r" we use only the most parsimonious one: "left of", "right of", "in front of", or "behind". We interpret these relations uniquely, i.e. no two of these relations interfere with one another so that "left of", for instance, indicates that the considered objects (the RO and the LO) are in the same horizontal line with any number of (zero to many) empty or filled cells between them.

Second, all operations on the spatial array – namely the reasoning processes – are considered as moves of a spatial focus. This focus can place an element into the model or inspect the model to find new information (Schaeken et. al., 1996), or write annotations to objects in cases of ambiguity (Vandierendonck et al., 2004). We assume that the reasoning process proceeds in three steps. In the *construction phase*, reasoners construct a mental model that reflects the information from the premises. For the preceding example, they, for instance, construct the following model:

screwdriver	pliers	hammer
wrench	saw	

In agreement with many experimental findings, we assume that if new information is encountered during the reading of the premises, it is immediately used in the construction of the model (Johnson-Laird & Byrne, 1991). In the *inspection phase*, this model is inspected to find new information that is not explicitly given in the premises. From this model it follows: *the wrench is to the left of the saw*. In the *variation phase,* alternative models are constructed from the premises that refute this putative conclusion. In our example no such model exists and thus the conclusion is valid. The formal reason for this phase is that a conclusion "follows" from a set of premises if the conclusion is true in all models of the premises. There are two concepts that explain how all these models can be checked – by a repeated iteration of the first two phases without using the prior constructed model, (Johnson-Laird & Byrne, 1991), or following our own account by saying that there is no iteration process but rather a process that starts from the preferred mental model (PMM) and then *varies* this model to find alternative interpretations of the premises (Rauh, Hagen, Knauff, Kuß, Schlieder, & Strube, 2005). There is a great number of arguments favoring the latter approach, the most important argument is of representational economy, i.e. most relational aspects can be reused – there is no need for them to be generated from scratch. The term PMM refers to a phenomenon encountered during reasoning with multiple-model problems in which reasoners often construct only one single model – the PMM. This model is the one that is easier to construct and to maintain in working memory than other possible models (Knauff, Rauh, Schlieder, & Strube, 1998). As it is known from many studies indeterminate problems are more difficult than determinate ones, and the PMM may frequently lead to incorrect conclusions because other possible models are ignored (Rauh et al., 2005).

The SRM model works on an input in the following way:

(1)Initially the SRM receives the first premise.

(2)The SRM model inserts the first object of the first premise in cell (0, 0). Then it uses this object as RO and adds the second object into the next adjacent cell according to the relation.

(3)The "parser" reads the next premise

(4)The SRM model decides on the type of premise:

- If an object of the premise is already in the spatial array, the focus moves to the RO and inserts into the next cell according to the relation the LO. If there is already an object, the focus moves to the next free cell according to the relation and inserts the object there. It also adds an annotation to this object, indicating that more than one position is possible.
- If none of the objects of the premise exists a new spatial array is generated and both objects are inserted as in step 2 (Schaeken et al., 1996).
- If both objects of the premise exist in the spatial array, the focus groups one model and inserts it into the other model (Bara et al., 2001).

When the model construction is finished, the *inspection phase* works for our example in the following way: a conclusion must be generated that defines the relation that holds between the wrench and the saw. So the focus moves to the wrench (RO) and then inspects the model to find the saw (LO). In previous studies, we were able to determine how this inspection process works (Knauff et al., 1998). After constructing the mental model, the focus is positioned on the last end-term of the last premise which should also be the starting point for the scanning of the RO. In our model, then the scanning for the LO proceeds in the same direction as before when it found the RO. This saves the costs of re-focusing (see below). If the LO cannot be found in this direction the focus changes its direction and proceeds until it has found the LO. It is important that in our model, the focus only checks the cells of the array in which an object is. Empty cells are not scanned. In other words, the system "knows" which cells are occupied but not which object is in the cell. If the LO is found from the scan direction the relation between the two objects is known (the meaning is again provided by the external module).

What happens if a possible conclusion must be verified? This is the case when the question for the relation is replaced by a conclusion that must be verified. Assume that the model must check whether the conclusion "The wrench is to the left of the saw" is valid. In this case, the focus moves to the saw (RO) and then scans the array to the left to find the wrench (LO). Since the conclusion is valid the model generates the output "valid conclusion".

It is important to notice that in the SRM model no variation of the model is assumed if a conclusion is generated. The SRM model stops when it has found just one model – which often leads to errors. Model variation only comes into play if a conclusion must be verified, or if more than one model can be constructed from the premises. We are still working on the exact details of the *variation phase*, but we definitely assume that there is no iteration of the first two phases in which alternative models are generated and inspected in turn (Johnson-Laird & Byrne, 1991). Instead, the current version of the SRM model starts from the PMM and then successively generates alternative models by modifying the PMM with minimal changes (Rauh et al. 2005). The minimal changes follow the principle of "conceptual neighborhood" which we have empirically determined in recent studies (Rauh et al. 2005). The principle says that alternative models are generated by local transformations, i.e. moving one object in the model. To find the next alternative model, the SRM model starts from the RO of the conclusion and first checks if the next objects have annotations with respect to the LO. As already mentioned, this annotation basically stores the relation that must hold between RO and LO. If so, (this is always the case in indeterminate problems because the premises itself are forgotten) the SRM model starts to change the position of the objects as long as the constraint from the annotation is satisfied. This leads stepwise to alternative models. As a consequence models which are difficult to reach are thus more likely to be neglected than models which are only minor revisions of the PMM. This phenomenon was reported in recent experiments (Rauh et al., 2005).

The SRM model also implies a complexity measure which could be described in short as a function of the number of relations to handle and the number of operations in the array. Abstract units are introduced for all operations on the model. This cost function reflects qualitatively the different difficulty of tasks, but there is no prediction of response times. For this reason we have implemented the SRM into an ACT-R model.

The ACT-R Model

The modeling task consisted in how to translate the formal, symbolic SRM of spatial reasoning to the production system ACT-R that comes as its own architecture of cognition. If this task is done properly, the model would allow statements about how close the complexity measures of the SRM model of spatial reasoning tasks are to experiments. Effects like premise- and term-order, indeterminacy, and verification-errors (all described below, see results) should be observable by means of processing times.

Like every production system, ACT-R offers a theory for the organization of knowledge. The knowledgebase is divided into a declarative and a procedural component. In the declarative part passive knowledge about facts is stored in so called chunks, sets of a given type with a given number of attribute-value pairs, the slots. Procedural knowledge comes as productions, condition-action pairs that usually modify chunks if their conditions are satisfied (Anderson et al., 2004). The problem to solve was how to model the elements of the SRM model in terms of these structures. This means how, i.e. with which of those components, should the spatial array that the SRM model describes as the spatial working memory (and that holds the representation of the mental model) be modeled; how the objects; how the relations; and how the focus which does all the operations in the array?

The presented ACT-R model works with the focus stored in the goal buffer that moves around in a virtual spatial array and manipulates object (and annotation) chunks. The spatial array is made up by a two-dimensional space in which both the focus and each object has its position. The coordinate values exist only for accessibility as there is no plane with fixed cells or the like. The focus has got slots for the object it is currently on, the direction it points to, and for position values. Every time an object is inserted into the spatial array a new object chunk is created and added to the declarative memory. These chunks have slots for the name of the object, for position values, and for annotation references. The spatial array is spun implicitly by these object chunks, empty cells do not exist. The implementation of the semantics of the relations between objects is described below. First let us see how the model works in the different phases – in other words how the productions are modeled.

At the beginning of the construction phase when the first premise has been read[1] the first object is inserted. This object is then used as the RO, and the focus sets its direction according to the relation given by the premise. The focus then moves one step in this direction and inserts the second object (the LO). As soon as the next premise has been read, the focus first has to check whether none, one, or both objects of this premise already exist in the spatial array. Usually one of the two objects can be retrieved. In this case the focus first moves to the existing object that is now the RO. Next it turns to the indicated direction and searches for a free cell for the LO in that direction. Here the implementation follows the theory of the PMM explicitly and applies the *first free fit* principle (FFF). As soon as this first free cell is found the LO is inserted. The alternative would be the *first fit* (FF) principle which would place the LO in the first cell next to the RO and if this cell is filled a more expensive shifting operation would be necessary. The fff principle, in contrast, means that the object is placed in the first free cell and if this cell is not the adjacent one to the RO, then the LO also gets an annotation containing the relational information of the current premise, i.e., an annotation chunk for which object is added to the declarative memory. If none of the objects of the premise could be retrieved, then the focus jumps to another level and starts a new spatial array. If both objects could be retrieved the focus searches the outermost objects in both spatial arrays of the objects in the direction according to the indicated relation and then groups these two spatial arrays into one.

In the inspection phase the focus first moves from its most recent position (which is the last position of the construction phase) to the first object to inspect. This is accomplished by a search process that includes all objects in the worst case. When the desired object is reached it acts as the RO and the second object – the LO – is searched. In this second search process the steps that the focus takes are stored for every direction so that afterwards when the LO is found the model can tell the relation between the two objects.

The variation phase comes into play if the premises allow the construction of multiple models – that is if there are any annotations – i.e. annotation chunks – in the constructed model. Moreover the variation phase is started only if there is a valid conclusion to be verified. The variation then acts very similarly to the inspection but with the search going solely in the direction indicated by the relation in the conclusion. If the LO is found in this direction, the conclusion is verified; otherwise it is falsified.

Maybe the biggest question during the implementation process was how to model the relations. On the one hand they should not be represented in any near-linguistic terms as this would contradict the main assumption of our model theory. On the other hand it was very important not to model them in any way close to visual imagery. Knauff and Johnson-Laird (2002) have shown that models have a different structure than images. Models represent the spatial

[1]The process of reading a premise itself is not modeled. Premises are read from a file and accordingly premise chunks are created at model initialization.

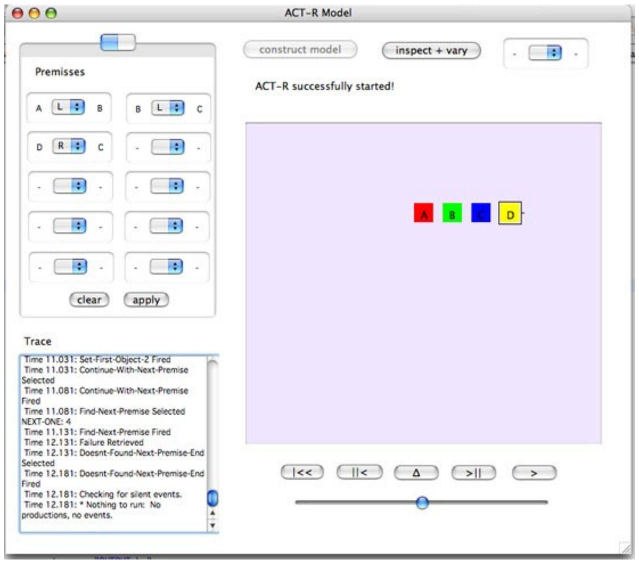
relations among entities, i.e., they represent what things are where, but in inferential tasks, they are likely to exclude visual detail, to represent only the information relevant to inference.

We first developed an account to model the relations as relation-chunks, all of one of four chunk types – left-of, right-of, in-front-of, or behind. The advantage was that there was no need for coordinates anymore. The actual positions of the objects could be left undiscovered while all objects were accessible simply by retrieving according relation-chunks. However, as there was an almost exact representation of the read premises and, worse, the model tended to be a simple rule-based system, this account was soon put aside. The current solution works with coordinates that specify the position of the objects and the focus. But rather than representing the relations as declarative units as in the previous account they are now modeled in a procedural way. Whenever the focus requests to move into one of the four possible directions a production fires to retrieve an object that lies somewhere in that direction. If the focus is to move to the left, for instance, all objects that are defined to be left of the focus are candidates for the next retrieval. Which object is retrieved depends on the association strengths of the focus (or the object the focus is currently on) and the objects to the left. Because the association strengths are higher for neighboring objects the adjacent object to the left of the focus will probably be the one that is retrieved.

Another interesting feature of the model is its limitation of the maximum number of objects and annotation chunks. The values come as parameters that can be set in the Java Interface – the default is a maximum of eight object chunks and two annotation chunks. The parameters can be used to adjust ACT-Rs retrieval threshold and the initial activations of object and annotation chunks. The base level activation of all chunks decays over time and increases whenever the chunk is retrieved. Also all chunks get activation from related chunks. If an activation of an object or annotation chunk falls below the retrieval threshold the according object or annotation will – at a higher level of description - not be present in the spatial array anymore. Thus, of course, the parameters mentioned above only give approximations of the actual limits. Chunks which fall below the threshold are typically those that are retrieved less often, that is, those that the focus visited less often. In limiting the capacity of those chunks the model is able to simulate errors. First of all (with the limited number of annotations) the common verification errors that arise when the validation of a generated conclusion in alternative models – by varying the constructed PMM – is omitted. The variation phase cannot start if there are no more (retrievable) annotations to resolve.

The Java Interface

While the model itself runs solely in ACT-R (i.e. Lisp) there is a comfortable user interface written in Java. On the one hand it offers some easy to use input mechanisms for premises and conclusions as well as parameter settings, and on the other hand it provides a controllable graphical representation of the constructed mental model at each time of the three phases. The communication with the ACT-R / Lisp processes was kept as simple and reliable as possible: There are script files and temporary data files which are called, or read, or written, from both systems as needed. The program can be downloaded and executed via the website http://www.informatik.uni-freiburg.de/~srm. Versions for MacOSX, Linux, and Windows are available. The program gives the opportunity to load preprocessed model files and thus to run those models likewise offline. But if the user wants to run models with their own set of premises the program depends on a working ACT-R 5.0 installation of course. The path to the Lisp interpreter has to be set according to the configuration of the client. The figure below gives an impression of the interface:



ACT-R successfully started!

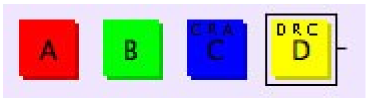
```
Time 11.031: Set-First-Object-2 Fired
 Time 11.031: Continue-With-Next-Premise
Selected
 Time 11.081: Continue-With-Next-Premise
Fired
 Time 11.081: Find-Next-Premise Selected
NEXT-ONE: 4
 Time 11.131: Find-Next-Premise Fired
 Time 12.131: Failure Retrieved
 Time 12.131: Doesnt-Found-Next-Premise-End
Selected
 Time 12.181: Doesnt-Found-Next-Premise-End
Fired
 Time 12.181: Checking for silent events.
 Time 12.181: * Nothing to run: No
productions, no events.
```

A Processing Example

To get a better insight to the ACT-R model we provide in the following a description of an exemplary run. Three premises shall be given:

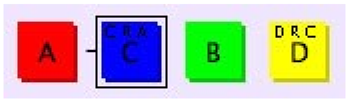
A is to the left of B
C is to the right of A
D is to the right of C

The construction phase starts by processing the first premise "A is to the left of B". Since the spatial array is empty the first object, "A", is just placed by the focus at the current position. Then the focus turns into the direction the relation of the premise indicates by using the first object as the RO so the desired direction is the opposite of the relation, namely "right". Now the focus moves one step to the left and inserts the LO, "B", at position (1,0). After the second premise, "C is to the right of A" has been read, it first has to check whether one, none, or both objects already exist in the spatial array. In this case only the second object, "A", exists, known by a successful retrieval of its chunk. The focus now moves to this object. To prevent (in some cases infinitive) loops the position of the searched object in relation to the focus is always known by an approximation of the horizontal or vertical direction. Thus the focus sets its direction to “left” and moves from object to object (empty cells are ignored by the activation matrix function) as long as its object slot does not match the searched object. While now object number one of the premise, “C”, is the LO the focus then changes its direction to “right” - in accordance with the relation of the premise. Now an empty cell for the LO is searched in this direction. As the next cell is occupied by an object, “B”, the focus continues moving till it finds a free cell following the named principle of first free fit (FFF) rather than that of first fit (FF) where the LO would have been inserted right next to “A” shifting “B” aside. Because at this point it is clear that the premises apparently allow the construction of more than one model the LO, “C”, gets an annotation holding the premise information. Therefore an annotation chunk for object “C” is created which later can be retrieved to create one or more alternative models. The third premise is processed much as the second one, but now the adjacent cell to the RO, “C”, is empty. Nevertheless the new object “D” gets an annotation because its neighbor, “C”, is annotated. This method of handing down annotations is necessary to be able to resolve all alternative models (see the variation phase below). The figure below shows the representation of the constructed model:[2]

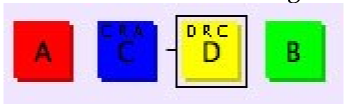


Now consider the question “Which relation holds between B and D?” shall be answered. The model starts the inspection phase. The focus moves around the spatial field (it inspects the spatial field) to find a relation between “B” and “D”. First the reference object (RO), “B”, has to be found. The focus starts at its last position at object “D”. It turns to the left and moves object to object till “B” is reached. Now the LO, “D”, is searched. While performing this search all steps in both dimensions (horizontal and vertical) of the focus are remembered. So when the focus finally reaches “D”, the relation between “B” and “D” can be generated on the basis of these steps. In this example the move from “B” to “C” results in a positive value for steps in the horizontal dimension and a zero value for those in the vertical dimension. With this setting a production fires that resolves the movement as a move to the right. Thus the relation “B is to the left of D” is given as the answer.

This conclusion is only valid if it holds in all possible models which can be created from the premises. So the verification and variation phase becomes important. The constructed model – the PMM – is varied step by step and in each alternative model the conclusion is verified. The annotations mark objects to be possibly placed on another cell. So the variation process works by searching annotated objects and then resolving these annotations to get to alternative models. The focus starts at the last position of the inspection phase, this is object “D” or cell (3,0). First it checks whether any annotations, namely annotation chunks, exist in the model or not. Here two annotation chunks exist so that the retrieval is successful. Now an annotated object is searched. Starting at object “D” the first attempt is a success. The annotation of object “D”, “D is to the right of C” (in short “D R C”), is retrieved and then the focus switches “D” and its adjacent object – in the direction the annotated relation indicates – “C”. But because “C” is the second object of the annotation this variation is not possible in the current model. So while “D” is the first annotated object the first annotation to resolve is that of “C” for “A”, the second object of “C’s” annotation is not adjacent to object “C”. The actual variation is done by changing the cells of the two objects – in cases where there are adherent objects in the orthogonal directions whole object structures have to be moved. Thus the positions of objects “C” and “B” are switched, meaning their according slots are modified. The model now looks as shown below:



Now the conclusion “B is to the left of D” is verified in the new model. For this the focus first moves to object “B” and then searches object “D” somewhere to the right, the direction the conclusion gives. Here “D” is to the right of “B” so the verification of the conclusion succeeds. Next the model is varied further considering the annotation of “D”. Since now the second object of “D’s” annotation, namely “C”, is not adjacent to “D”, “D” can switch positions with “B”. The figure below shows the resulting model:



In this model the found conclusion “B is to the left of D” apparently does not hold. The verification process ends with a negative result, thus the conclusion is falsified for the model given by the premises. No more verification or variation is necessary.

Comparison with the performance of human subjects

The model shows at least three different effects that are typical in spatial deduction processes. First, the premise order effects the processing times significantly. If the premises are ordered in a continuous way the model takes less time to run than in cases where the premises are ordered discontinuously. A continuous order such as “A is to the left of B”, “B is to the left of C”, and “C is to the left of D”, for instance, takes a total of 3,709 seconds – with the first premise at 1,303 s, the second one at 1,204 s, and the third one at 1,202 s. In contrast, the discontinuous order “A is to the left of B”, “C is to the left of D”, and “B is to the left of C” of the same model takes a total of 5,488 seconds – with the first premise at 1,303 s, the second one at 1,523 s, and the third one at 2,662 s. In the latter case, the third premise took longer because in processing the second premise two separate spatial arrays had been created which then had to

[2]All figures showing the representation of the model are taken from the Java interface clipped with the screen part to the right.

be integrated into one again. These times agree with findings from Knauff, et al. (1998).

Second, indeterminate problems are harder (they take longer) than determinate ones. Indeterminate problems are those that allow the construction of more than one model from the premises (Byrne & Johnson-Laird, 1998). The model of the processing example in the last section ("A is to the left of B", "C is to the right of A", "D is to the right of C"), apparently an indeterminate problem, for instance, has a time for the construction phase of 5,573 seconds – more than those times of the determinate continuous and semi-continuous problems above. While this difference is due to the creation of annotations this model also has verification/variation phases joining the instruction phase.

Third, people often make errors when they have to draw a conclusion and omit to verify it in all alternative models. They only take into account the PMM (Knauff et al., 1998). This effect is simulated by the limited amount of annotation (and object) chunks available at each time during the processes. The approximate maximum number of object and annotation chunks can each be specified in the settings dialog of the Java interface. These parameters are used to adjust the retrieval threshold of ACT-R. If an object or annotation chunk falls below this threshold the chunk can not be retrieved anymore. ACT-Rs chunk activation is calculated as a function of the chunk's creation time and the number (and points in time) of references of the chunk and the spreading activation of related chunks. Therefore, those chunks will probably fall below the retrieval threshold that are retrieved the less and are far away from the focus. These chunks will not be available in the variation phase which means that the alternative models that could have been created from a lost annotation will not be considered at all.

Discussion

The ACT-R model works fine for the named effects. It shows that the SRM model is able to predict the complexity of a given problem in terms of processing times (or units of cost measures) and error patterns. The presented times of the exemplary runs quantitatively match times of experiments except for an almost constant factor – the times for each premise of the experiment are higher than those of the ACT-R model. In the following table the processing times are compared to experimental data from Knauff et al. (1998):

Premise order	Premise 1	Premise 2	Premise 3
continuous	13.0 (1.303)	11.2 (1.204)	10.9 (1.202)
semi-continuous	13.6 (1.303)	11.0 (1.204)	14.4 (1.794)
discontinuous	12.4 (1.303)	13.9 (1.523)	19.5 (2.662)

Mean reading times of premises in seconds from Knauff et al., 1998. The values in parenthesis are the processing times of the ACT-R model.

To some extent this may be due to the fact that the default ACT-R mechanisms are set too fast for such complex spatial reasoning tasks. Furthermore, an important reason may be that the model contains no simulation of the reading and interpreting process of the premises. The premises are created as chunks at initialization of the model. The time gap could be closed by adjusting according ACT-R parameters. While the current implementation indeed uses no such parameter adjustments the processing times are qualitatively comparable to experimental results.

Further work on the model could concentrate on an implementation of the reading and encoding processes of the premises and an integration of the phonological loop (Baddeley & Hitch, 1974) in which the annotations could be remembered to get an even better simulation of the verification errors.

Acknowledgements

The research was supported by grants from the DFG in the Transregional Collaborative Research Center (SFB/TR 8) to MK and BN. M.K. is also supported by a Heisenberg Award from the DFG.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review 111,* (4). 1036-1060.

Baddeley, A. D., & Hitch, G. J. (1974). Working Memory. In G. A. Bower (Ed.), *Recent advances in learning and motivation* (Vol. 8, pp. 47-90). New York, Academic Press.

Johnson-Laird, P. N. (2001). Mental models and deduction. *Trends in Cognitive Science*, 5, 424-442.

Johnson-Laird, P. N., & Byrne, R. M. J. (1991). Deduction. Hove, UK: Lawrence Erlbaum Associates.

Knauff, M. & Johnson-Laird, P. N. (2002). Visual imagery can impede reasoning. *Memory & Cognition*, 30, 363-371.

Knauff, M., Rauh, R., Schlieder, C., & Strube, G. (1998). Mental models in spatial reasoning. In C. Freska, C. Habel & K. F. Wender (Eds.), *Spatial Cognition I - An Interdisciplinary Approach to Representing and Processing Spatial Knowledge.* Berlin: Springer.

Miller, G. A., & Johnson-Laird, P. M. (1976). Language and Perception. Cambridge, Cambridge University Press.

Ragni, M., Knauff, M., Nebel, B. (2005). A Computational Model for Spatial Reasoning with Mental Models. Lawrence Erlbaum Associates. *Proceedings of the 27th Annual Cognitive Science Conference.* Bruno G. Bara, Lawrence Barsalou, & Monica Bucciarelli, pp. 1797.

Rauh, R., Hagen, C., Knauff, M., Kuß, T., Schlieder, C., & Strube, G. (2005). Preferred and alternative mental models in spatial reasoning. *Spatial Cognition and Computation*, 5, 239-269. .

Schaeken, W., Johnson-Laird, P. M., & d'Ydewalle, G. (1996). Mental models and temporal reasoning. *Cognition* 60, 205-304.

Schlieder, C. (1995). The construction of preferred mental models in reasoning with interval relations. In G. Rickheit & C. Habel (Eds.), *Mental models in discourse processing and reasoning*, pp. 333-357. Amsterdam, Elsevier.

Vandierendonck, A., Dierckx, V., & Vooght, G. D. (2004). Mental model construction in linear reasoning: Evidence for the construction of initial annotated models. *The Quarterly Journal Of Experimental Psychology*. 57A. 1369-1391.