Simulation and Analysis of Controlled Multi-Representational Reasoning Processes*

Tibor Bosse[1], Catholijn M. Jonker[1], and Jan Treur[1,2]
({tbosse, jonker treur}@cs.vu.nl)

[1] Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
URL: http://www.cs.vu.nl/~{tbosse, jonker, treur}

[2] Universiteit Utrecht, Department of Philosophy
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

Abstract

Multi-representation reasoning processes often show a variety of reasoning paths that can be followed. To analyse such reasoning processes with special attention for differences between individuals, it is required (1) to obtain an overview of the variety of possibilities and (2) to address navigation and control within the reasoning process. This paper presents a simulation model and analysis for the dynamics of a controlled reasoning process in which multiple representations play a role. Strategies to navigate through the space of possible reasoning states are modelled explicitly, and simulated. Simulation results are analysed by software tools on the basis of formalised dynamic properties.

Introduction

Human reasoning is often considered a process proceeding by accumulating a number of reasoning steps from start to end. An underlying assumption is that such a process can be analysed by studying each such step locally, in isolation from the rest of the reasoning process. Many reports of experimental research focus on one-trial-experiments where the number of reasoning steps is limited to one or, sometimes, at most two; e.g., (Rips, 1994; Johnson-Laird, 1983). However, a practical reasoning process often is not a straightforward accumulation of isolated steps. First, decisions to make a reasoning step may be not a local issue at the time point of the decision, but depend on the history and goals of the reasoning process as a whole. Second, often a multitude of

* In: Proc. of the 5th International Conference on Cognitive Modelling, ICCM'03, Bamberg, 2003, pp. 27-32.

reasoning paths is possible; only some of these actually reach the goal. Navigation and control in the sense of making a coherent set of choices at different time points to obtain one of the successful (and preferred according to one's own characteristics) paths is a nontrivial issue. Third, during the process steps may be taken that lead to a dead end, such that the reasoning process has to reconsider these steps, leading to revision of the reasoning path. These non-local aspects of a reasoning process require specific capabilities beyond, for example, the capability to locally apply modus ponens or modus tollens. Often some form of global reasoning planning and control is performed. Decisions to make or revise a specific reasoning step are made in the context of such a reasoning plan, which also has to be taken into account as part of a reasoning state.

In many cases the same information can be represented in different manners (e.g., in arithmetic, geometric or material form). Moreover, both internal (mental) and external (e.g., written or drawn) representations may play a role. The distinction between mental and external representations is also made in, e.g., (Hegarty, 2002). As the type of possible reasoning steps may be different for different forms of representation, these differences of representation have to be accounted for in different reasoning states. In such cases the number of possible reasoning states is not very small, and, as a consequence, the number of possible reasoning paths, may be quite large. Coherent controlled navigation involving non-local aspects of decisions for reasoning steps is of major importance to deal with such a large number of possibilities.

This paper reports analysis and simulation of controlled multi-representation reasoning processes, in which the issues put forward play an important role. An analysis method for the dynamics of reasoning (adopted from Jonker and Treur (2002)) is based on formal definitions of possible reasoning states and traces, and dynamic properties of these traces are specified in the Temporal Trace Language TTL. This analysis method is supported by a software environment that is able to check traces against specified dynamic properties. For simulation the component-based agent design method DESIRE is used. Traces generated by execution of a DESIRE model can be directly used as input of the analysis software environment.

Below, first the dynamic perspective on reasoning is discussed in some more detail. Next, an example reasoning pattern is introduced, and the first steps of an analysis are made. The example multi-representation reasoning process used to illustrate the approach put forward shows interaction between material, geometric and arithmetic reasoning. The example focuses on how to determine the outcome of multiplications such as 23 x 36, possibly using external arithmetic, geometric or material (based on Multi-base Arithmetic Blocks (MAB) material; e.g., Booker et al. 1997, English and Halford, 1995) representations. Third, the design of the simulation model is presented. Various

simulation traces have been generated, of which one example is briefly discussed. Fourth, a number of dynamic properties for this type of reasoning are identified. These properties have been checked for the generated simulation traces. Finally the approach is summarised and the contribution of the research presented in the paper is discussed.

Reasoning Dynamics

As in (Jonker and Treur, 2002), to formalise the dynamics of a reasoning process, traces are used. A difference here is that also material representations and for each representation form an internal and an external variant is considered. *Reasoning traces* are time-indexed sequences of *reasoning states* over a time frame: the set of natural numbers. The set of all possible reasoning states defines the space where the reasoning takes place. Reasoning traces can be viewed as trajectories in this space, for which every (reasoning) step from one reasoning state to the next one is based on an allowed *transition*. The set of proper reasoning traces can be defined as the set of all possible sequences of reasoning states consisting only of allowed transitions.

Reasoning States

A *reasoning state* formalises an intermediate state of a reasoning process. The content of such a reasoning state usually can be analysed according to different aspects or dimensions. A reasoning state can include both internal (e.g., specific mental representations) and external elements (e.g., written or drawn notes). For example, part of the state may contain an external material representation, another part an external arithmetic representation, and yet another part an internal geometric representation. Furthermore, as pointed out in the Introduction, also control information has to be taken into account in a reasoning state. Accordingly, the reasoning state is structured as a composition of (i.e., a tuple of) a number of parts, indexed by some set I. This index set includes different aspects or views taken on the state, e.g., I is the set

{control, extmaterial, extgeometric, extarithmetic, intmaterial, intgeometric, intarithmetic}.

The set of reasoning states RS can be characterised as a Cartesian product $RS = \Pi_{i \in I} RS_i$ where RS_i is the set of all states for the aspect indicated by i. For example, $RS_{extgeometric}$ may denote the set of all possible external (drawn) geometric representations. This Cartesian product formalises the multi-dimensional space where the reasoning takes place. For a reasoning state, which is a vector $S = (S_i)_{i \in I} \in RS$ in this space, the S_i are called its *parts*.

Reasoning Steps: Transitions of Reasoning States

A transition from one reasoning state to another reasoning state, i.e., an element $< S, S' >$ of $RS \times RS$, formalises one *reasoning step*; sometimes also denoted by $S \rightarrow S'$. Transitions differ in the set of parts that are involved. The most complex transitions change all parts of the state in one step. However, within stepwise reasoning processes, usually transitions only involve a limited number of parts of the state, e.g., one to three. In the current approach we concentrate on this class of transition types.

For example, when a modification in the reasoning state is made solely within an internal geometric representation, only the internal geometric part of the state changes (geometric reasoning step): intgeometric $\rightarrow$ intgeometric. An example of such a transition can be visualised as in Fig.1:

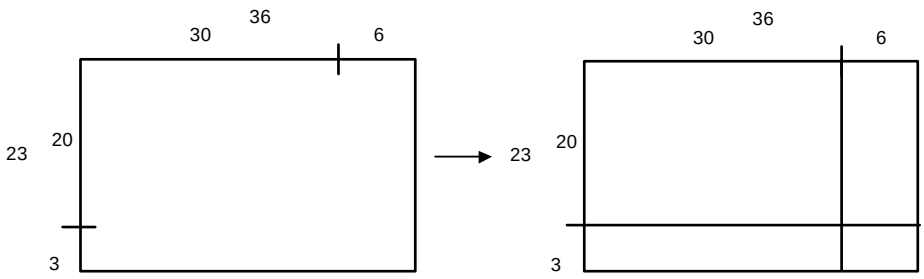

Figure 1 Visualised transition

Other types of transitions involve more than one part. For example, if an external geometric representation is extended on the basis of an internal geometric representation, then two parts of the state are involved: the external geometric arithmetic part and the internal geometric part:

extgeometric x intgeometric $\rightarrow$ extgeometric

E.g., the external geometric representation is extended or modified with results from the internal geometric representation. If control information is incorporated in the modelling approach the number of involved parts is even higher, since every transition involves the control part; e.g.: intarithmetic x control $\rightarrow$ intarithmetic or extgeometric x intgeometric x control $\rightarrow$ extgeometric.

Reasoning Traces

Reasoning dynamics results from successive reasoning steps, i.e., successive transitions from one reasoning state to another. Thus a *reasoning trace* or *trajectory* is constructed: a time-indexed sequence of reasoning states $(\gamma_t)_{t \in T}$, where T is the time frame used (the natural numbers). A reasoning trace can be viewed as a trajectory in the multi-dimensional space $RS = \Pi_{i \in I} RS_i$ of reasoning states. Reasoning traces are sequences of reasoning states subject to the constraint that each pair of successive reasoning states in this trace forms an allowed transition. A trace formalises one specific line of reasoning.

Multi-Representation Reasoning Process

Experiences on using multi-representational reasoning processes with children (8-9 years old) in classrooms have been reported, e.g., by (Dekker et al., 1982), see also (Hutton, 1977). Also teaching quadratic equations can be supported by such materialisations and visualisations; e.g., by (Bruner, 1968), pp. 59-63; see also (Koedinger and Terao, 2002) for further explorations of the idea to use visualisations in pre-algebraic reasoning. The example pattern may show a large number of transition types involving one to three parts. The idea is that only simple arithmetical steps are required. The more complicated steps are performed via the external material or geometrical representation. A number of basic skills are assumed. These basic skills can be defined in the form of transitions as described above. A variety of (part of the) possible reasoning paths determined by such transitions is depicted in a simplified manner in Figure 2; here the focus lies on transitions involving (both internal and external) arithmetic representations, and switches between geometric and arithmetic representations. For the sake of simplicity material steps, and transitions between geometric and material representations have been left out. The numbers refer to basic skills. E.g., the transition labelled "4" refers to skill bs4, i.e., partitioning a rectangle in non-overlapping areas, based on a partitioning of its sides, and the transitions labelled "7" refer to skill bs7, i.e., splitting a number in tens and digits.

Simulation Model

The simulation model[1] is based on agent modelling techniques, in particular the component-based agent modelling approach DESIRE; cf. (Brazier et al., 2002). At the highest level of abstraction, two components play a role in the system, i.e., the reasoning agent (called *Alan*) and the *External*

[1] A complete specification of the model (with clickable components) can be found at www.cs.vu.nl/~wai/GTM/rmr/

World. Alan can perform actions and observations, executed in the external world, and receive observation results as input from the external world. After Alan generates a certain action to be performed (e.g., draw a rectangle with sides 23 x 36), this action is transferred to the external world and executed there.

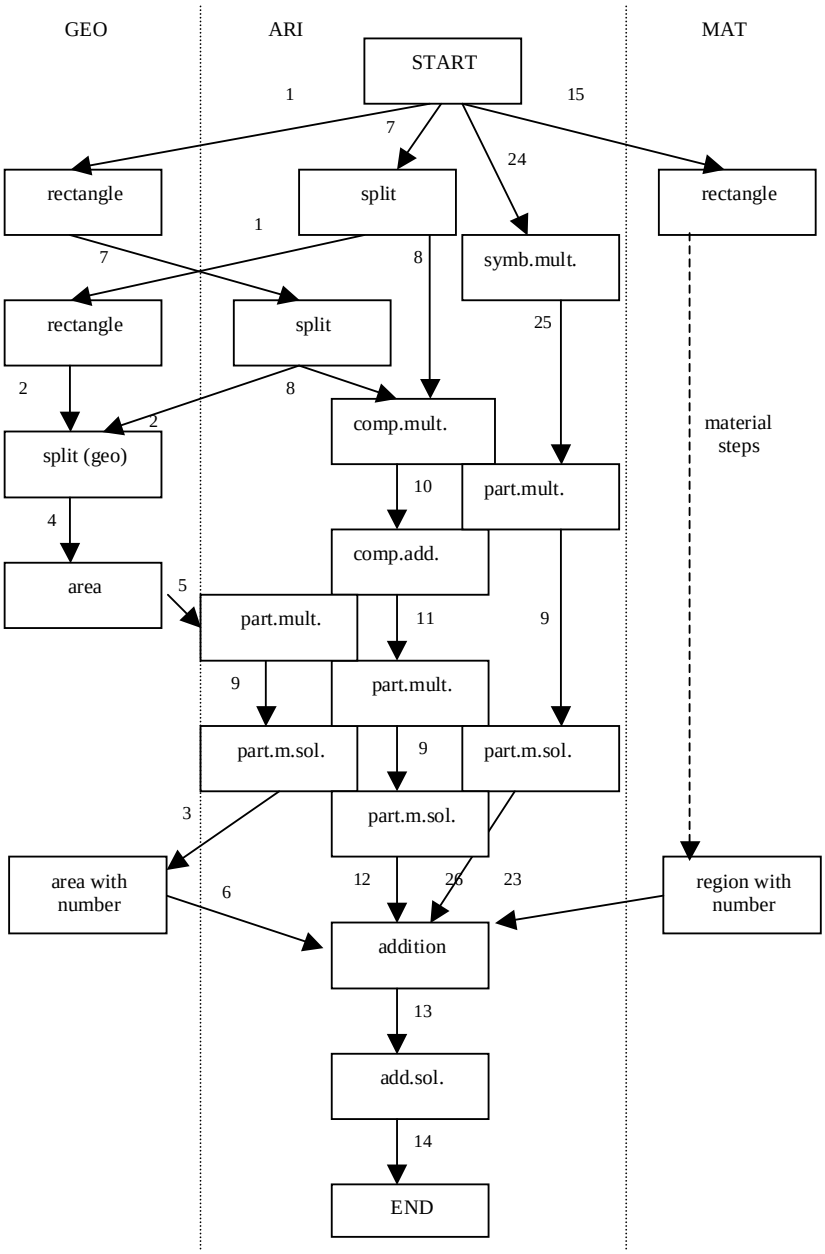

Figure 2 Variety of reasoning paths

The result of the action (e.g., a rectangle with corners A, B, C, D and sides 23 x 36 drawn on a piece of paper) will occur, with a certain delay, within the external world. Besides performing actions, Alan can pro-actively observe the world. The agent does this by explicitly determining what aspects of the world it is interested in: its observation focus. This focus is then transferred to the external world, which in return provides the corresponding observation result. Figure 3 depicts an overview of the components of the simulation model.

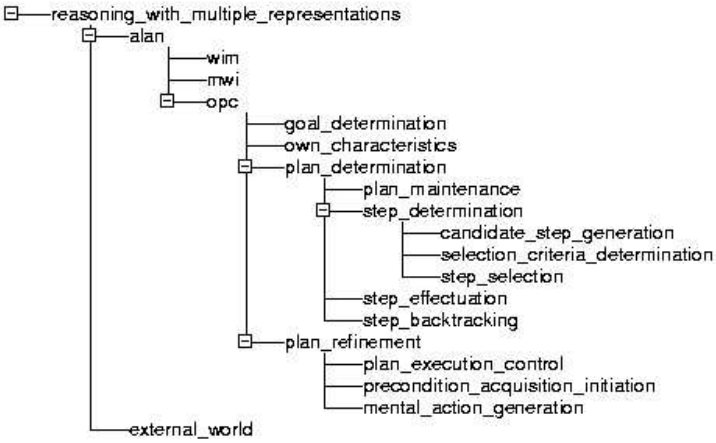

Figure 3 Overview of the components of the simulation model

Reasoning Agent

The approach used in this paper assumes that for every action a mental and a physical part can be distinguished and modelled (e.g., imagining a rectangle with sides 23 x 36 vs. actually drawing such a rectangle). Whilst the external world is concerned with the physical parts of the actions, everything that is represented within the agent is mental. To be able to make a clear distinction between the two concepts, a different notation is used for both types of information, e.g., rectangle(A, B, C, D, 23, 36) denotes a specific rectangle in the world, whereas entity(shape([]), parameters(23, 36)) denotes the internal representation. Internal representations can be created on the basis of an observation, but also on the basis of internal reasoning.

The composition of the reasoning agent Alan is based on the generic agent model as described in (Brazier et al., 2002). Three of the generic agent components are used in our model, namely *World*

Interaction Management, *Maintenance of World Information* and *Own Process Control*. The other generic agent components were not needed within this model.

World Interaction Management handles the interaction with the external world, i.e., observation and action performance. It interprets information from the world and makes it available for relevant other components. Also it prepares information on actions to be performed.

The task of the component *Maintenance of World Information* is to maintain a (partial) world model, i.e., a snapshot of the present world state. In this domain, this world model is restricted to the observed information about objects that the agent has manipulated itself, such as the numbers it has written down. Moreover, since the agent does not necessarily have to perform each intermediate step physically, some imaginary world model must be maintained as well. This model describes the world like it would be after the physical execution of some steps, without these steps actually being performed. As both models contain information about a (possible) state of the world, both are maintained by Maintenance of World Information.

According to the generic agent model, tasks of the component *Own Process Control* are the processes the agent uses to control its own activities (e.g., determining, monitoring and evaluating its own goals and plans), but also the processes of maintaining a self model. The way the tasks are performed is described in detail in the next section.

Own Process Control

Own Process Control consists of four sub-components: *Goal Determination*, *Own Characteristics*, *Plan Determination* and *Plan Refinement*.

For the application in question, *Goal Determination* is a relatively simple component. It contains information about the initial multiplication problem the agent desires to solve.

The component *Own Characteristics* contains a self-model, which includes several aspects. In the first place, it includes (self-)information on the basic skills that the reasoning agent thinks to possess. Note that this does not necessarily mean that the agent indeed has all these skills. For instance, it is perfectly possible that the agent believes to be able to apply the distribution law of arithmetic, whilst during execution it turns out that it does not. Whenever a certain skill has failed (i.e., the agent planned to use it, but at the end, it could not), Own Characteristics revises the self-knowledge of the agent by asserting that it does not have the skill after all. Third, Own Characteristics is used to store the agent's profile with respect to its problem solving strategy for the multiplication problem. Two aspects are represented: (1) a list of priorities among the different representations that can be used while solving the problem, e.g., ari-geo-mat, and (2) to what extent

steps in the reasoning process have to be performed physically. This way, several types of agents can be modelled, varying from those that write down every step to those that write down nothing. As a final remark, notice that, although DESIRE offers the opportunity to dynamically add changes in the specification (and thereby realise an open state space), this has not been done within the current model.

Before actually solving the problem, the reasoning agent makes an abstract plan (e.g., a particular navigation route through Figure 2). *Plan Determination* is responsible for this planning process. Its input consists of the agent's own goal and characteristics. Based on this information, and knowledge about pre- and postconditions of the basic skills, Plan Determination explores the entire reasoning process at an abstract level. The pre- and postconditions are expressed in an abstract way; e.g., they do not contain any numbers. While planning, Plan Determination continuously matches the current state of the explored plan against the preconditions of all basic skills, in order to determine which skills are applicable. It then uses its strategy profile in order to select one of the applicable skills. Subsequently, the skill is evaluated by adding its (abstract) postcondition to the current state of the explored plan. This way, the component constructs a complete list of steps to be performed, that would solve the multiplication problem. Furthermore, the component uses backtracking in situations where no more basic skills are applicable. Finally, if no solution can be found at all, this is also indicated. The sub-components of Plan Determination will be described in the next section.

Abstract plans, generated by Plan Determination, are transferred to the component *Plan Refinement*. This component, which consists of the sub-components *Plan Execution Control*, *Precondition Acquisition Initiation* and *Mental Action Execution*, is responsible for the refinement of the basic steps, i.e., it determines the specific mental and physical actions associated to a basic step of the abstract plan (e.g., it refines bs4 to bs4m). Moreover, it executes the detailed mental actions associated to the basic steps. This is done by repeating the following activities. First, Plan Execution Control selects the first step of the (remaining) plan to be executed. Second, Precondition Acquisition Initiation determines what observations have to be made to provide the agent with the necessary information for the application of the selected step. For instance, if the selected step is to draw a rectangle, it is important to know the dimensions. Third, as soon as this information has been obtained, Mental Action Execution creates a mental image of the result of the application of the mental action (with instantiated variables, e.g., 'a rectangle with sides 23 x 36', denoted by entity(shape([]), parameters(23, 36))). This mental image is then stored within Maintenance of World Information. After that, Plan Execution Control decides whether to perform the associated physical

action as well, depending on the agent's own characteristics. Then, the physical action either is or is not executed (within the External World component), after which the next step of the plan is treated by Plan Execution Control. Finally, if the agent is unable to perform an action that it had planned to do because it lacks either the mental or the physical skill for that action, notification with the name of the skill that failed is transferred to Own Characteristics. As a consequence, this latter component will revise its self-model, so that Plan Determination can construct a new plan more adequately.

Plan Determination

Plan Determination consists of the components Plan Maintenance, Step Determination, Step Effectuation and Step Backtracking. *Plan Maintenance* keeps track of all kinds of information concerning the 'current' state of the explored reasoning process, such as the (abstract) steps that have been applied successfully, those that have failed and those that have not been applied yet. *Step Determination* determines the next step to be added to the current plan in three phases. First, it determines which steps are currently applicable, by matching the preconditions of abstract steps against the current state of the exploration. Second, based on the applicable steps and the agent's strategy profile, it decides whether it will make an arithmetic, geometric, or material step. And third, based on the chosen representation, it will select one single step. The components responsible for the three phases are called, respectively, *Candidate Step Generation*, *Selection Criteria Determination*, and *Step Selection*. Finally, the selected step is passed to Step Effectuation. However, if, independently of the representation, no steps are applicable, this failure is indicated, so that the backtracking component can become active. *Step Effectuation* explores the execution of the selected abstract step by adding the postcondition of the step to the current state of the simulation. *Step Backtracking* becomes active whenever no more steps are applicable and uses a standard backtracking algorithm.

Example Simulation Trace

Although many simulations have been performed, there is only room to present (part of) one trace. The trace selected uses geometric and arithmetic skills to solve the problem.

```
strategy profile:   geo-ari-mat
available abstract skills:      all skills
available mental skills:        all skills
available physical skills:      all skills
represent physically:           all steps
```

```
mental_representation(arithmetic, entity(shape("X*Y"), parameters(23, 36)))
plan([bs1, bs7, bs2, bs4, bs5, bs9, bs3, bs6, bs13, bs14])
is_represented_in_world(arithmetic, multiplication(23, 36))
mental_representation(geometric, entity(shape("[]"), parameters(23, 36)))
is_represented_in_world(geometric, rectangle('A', 'B', 'C', 'D', 23, 36))
mental_representation(arithmetic, entity(shape("X=X1+X2"), parameters(36, 30, 6)))
mental_representation(arithmetic, entity(shape("X=X1+X2"), parameters(23, 20, 3)))
...             /* split 23 into 20 and 3; split 36 into 30 and 6 */
mental_representation(geometric, entity(shape("[]"), name('A11'), dim_parameters(20, 30)))
mental_representation(geometric, entity(shape("[]"), name('A12'), dim_parameters(20, 6)))
mental_representation(geometric, entity(shape("[]"), name('A21'), dim_parameters(3, 30)))
mental_representation(geometric, entity(shape("[]"), name('A22'), dim_parameters(3, 6)))
...             /* define four area's with sides 20x30, 20x6, 3x30, 3x6 */
mental_representation(geometric, entity(shape("[]"), name('A11'), area_parameter(600)))
mental_representation(geometric, entity(shape("[]"), name('A12'), area_parameter(120)))
mental_representation(geometric, entity(shape("[]"), name('A21'), area_parameter(90)))
mental_representation(geometric, entity(shape("[]"), name('A22'), area_parameter(18)))
...             /* assign a number to each area */
mental_representation(arithmetic, entity(shape("V+W+X+Y=Z"), parameters(600, 120, 90, 18, 828)))
is_represented_in_world(arithmetic, addition_solution(600, 120, 90, 18, 828))
mental_representation(arithmetic, entity(shape("XX*YY=ZZ"), parameters(23, 36, 828)))
is_represented_in_world(arithmetic, multiplication_solution(23, 36, 828))
```

The trace first contains a description of the characteristics of the agent, then the arithmetic problem is mentally represented and the plan is produced. Due to the strategy profile of the agent, the plan shows as many basic geometric skills as possible (this corresponds to the left part of Figure 2). Every step is represented both mentally and physically, corresponding to the agent's characteristics. Since the agent has all skills both in abstracto and in concreto, no backtracking was necessary either during plan determination or plan execution.

Analysis in Terms of Dynamic Properties

As in (Jonker and Treur, 2002), to specify properties on the dynamics of a reasoning process, the temporal trace language TTL is used. In short, in TTL it is possible to express that in a given trace at a certain point in time the reasoning state has a certain (state) property. Moreover, it is possible to relate such state properties at different points in time. For the case at hand, more than 70 of such dynamic properties have been specified, varying from global properties for the overall reasoning process to more local properties. The idea is that part of these properties are of a general nature (i.e., they can be used to assess whether a trace qualifies as a proper reasoning trace), whereas the other properties are used to characterise the different types of possible traces (i.e., they are used to

identify individual differences). A large number of automated checks have been performed, to reveal which properties hold for which traces.

Global properties

As an example, the following (global) property of a reasoning trace γ is considered, which expresses that in a trace all multiplication problems in two digits eventually will be solved without using any external geometric and material representations.

GP1(extarithmetic)

at any point in time t

if in the reasoning state in trace γ at t an external arithmetic representation of a multiplication problem for numbers x and y < 100 is present,

then a time point t' ≥ t exists such that in the reasoning state in γ at t' an external arithmetic representation of a solution z of this multiplication problem with z = x∗y is included

and for all t'' with t ≤ t'' ≤ t' it holds that in the reasoning state in γ at t'' no external geometric or material representation is included.

The formalisation of this property in TTL is as follows.

∀t ∀x, y < 100 state(γ, t, extarithmetic) |== multiplication_problem(x, y)

⇒ ∃t'≥t ∃z z = x∗y & state(γ, t', extarithmetic) |== multiplication_solution(x, y, z)

& ∀t'' [t ≤ t'' ≤ t' ⇒ [∀a state(γ, t'', extgeometric) |=/= a &

∀a state(γ, t'', extmaterial) |=/= a &]]

Note that for simplicity no maximal allowed response time has been specified. If desired, this can be simply added by putting a condition t' ≤ r in the consequent with r the maximal response time. Similarly, other variants of overall properties can be specified, for example expressing that within the trace all three types of external representations have been used, or no external representations at all.

Milestone Properties

Within the overall reasoning process a number of dynamic properties can be defined that express whether the process has reached a certain milestone; for example, to indicate that a reasoning state was reached in which an external material representation full with blocks in the areas occurs. Another form of milestone property expresses that internally an abstract plan was reached.

Local Properties

A large number of properties have been specified that each characterise one reasoning step. For the sake of simplicity, for the example reasoning process persistence of representations in reasoning states over time is assumed. As examples, two local properties are represented; one representing a transition from intarithmetic to intgeometric and one from intgeometric to intgeometric.

LP1 (intarithmetic-intgeometric)

at any point in time t

if in the reasoning state in trace γ at t an internal arithmetic representation of a multiplication problem for numbers x and y between 10 and 100 is present,

then a time point $t' \geq t$ exists such that in the reasoning state in γ at t' an internal geometric representation of a rectangle ABCD with |AB| = x and |AD| = y is present.

LP4 (intgeometric-intgeometric)

at any point in time t

if in the reasoning state in trace γ at t an internal geometric representation of a rectangle ABCD is present with points P on AB and Q on AD,

then a time point $t' \geq t$ exists such that in the reasoning state in γ at t' the rectangle ABCD is partitioned into four areas A11, A12, A21, A22 by two lines PP'//AD and QQ'//AB with P' on CD and Q' on BC.

Relationships between Properties

During the analysis also relationships between properties are identified. For example, analysis shows that if the agent's strategy profile prefers arithmetic and the agent has the necessary skills to solve the problem arithmetically, then GP1(extarithmetic) holds.

Discussion

Analysis of the cognitive capability to perform reasoning has been addressed from different areas and angles. Within Cognitive Science, the two dominant streams are the syntactic approach (based on inference rules applied to syntactic expressions, as common in logic), e.g., (Rips, 1994), and the semantic approach (based on construction of mental models); e.g., (Johnson-Laird, 1983). In experimental work for these approaches reasoning processes usually are studied by focussing on reasoning steps in isolation, by means of one-trial experiments. More extensive reasoning processes involving a number of steps that are tuned to each other require coherent controlled navigation. The current paper reports analysis and simulation of such a reasoning process.

The analysis method for the dynamics of reasoning processes used in this paper was adopted from (Jonker and Treur, 2002) and validated on the basis of reports from experiments with 8-9 year old children in classrooms in the Netherlands (Dekker et al., 1982). A similar report has been made by (Hutton, 1977). The current paper shows how an analysis of these dynamics can be made using traces consisting of sequences of reasoning states including control information over time to describe controlled reasoning processes. It is shown for the example reasoning pattern, how characterising dynamic properties can be identified. Furthermore, the agent modelling approach DESIRE has been used to implement a simulation model, and other software tools have been used to automatically check which dynamic properties hold for which simulated traces. In addition, these software tools can be used to check which properties hold for empirical data, thereby supporting the comparison of human reasoning with simulated reasoning. The variety of dynamic properties specified and the variety of traces simulated provides an overview for the individual differences between subjects that have been observed while solving multiplication problems. For example, using our formalisation those with an emphasis on external arithmetic representations are neatly distinguished from those who use external material representations where possible. In the analysis the notion of reasoning strategy was addressed, incorporating such differences in skill and preference. Due to the compositional structure of reasoning state it was not difficult to extend a reasoning state with a component for control information.

Further experiments will be conducted, in which also a focus is more explicitly on the control of the reasoning. For example, are subjects able to explain why at a point in time a translation to a geometric representation is made? Are think-aloud protocols involving control information a reliable source of further analysis? In addition, future work will explore the possibility to reuse the current simulation model in other cognitive domains.

References

Booker, G., Bond, D., Briggs, J. & Davey, G. (1997). Teaching Primary Mathematics, Melbourne: Longman.

Brazier, F. M. T., Jonker, C.M., and Treur, J., (2002). Principles of Component-Based Design of Intelligent Agents. Data and Knowledge Engineering, vol. 41, pp. 1-28.

Bruner, J.S. (1968). Toward a Theory of Instruction. Norton & Company, Inc. New York.

Dekker, A., Heege, H. ter, and Treffers, A. (1982). Cijferend vermenigvuldigen en delen volgens Wiskobas. Universiteit Utrecht, Freudenthal Institute.

English, L. & Halford, G. (1995) Mathematics Education: Models and Processes. Mahwah, NJ: Lawrence Erlbaum

Hegarty, M. (2002). Mental Visualizations and External Visualizations. In: W.D. Gray and C.D. Schunn (eds.), Proceedings of the 24th Annual Conference of the Cognitive Science Society, CogSci'02. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2002, p. 40.

Hutton, J. (1977). Memoirs of a Maths Teacher 5: Logical Reasoning. In: Mathematics Teaching, vol. 81, pp. 8-12.

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

Jonker, C.M., and Treur, J. (2002). Analysis of the Dynamics of Reasoning Using Multiple Representations. In: W.D. Gray and C.D. Schunn (eds.), Proceedings of the 24th Annual Conference of the Cognitive Science Society, CogSci 2002. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2002, pp. 512-517.

Koedinger, K.R., and Terao, A. (2002). A Cognitive Task Analysis of Using Pictures To Support Pre-Algebraic Reasoning. In: W.D. Gray and C.D. Schunn (eds.), Proceedings of the 24th Annual Conference of the Cognitive Science Society, CogSci'02. Mahwah, NJ: Lawrence Erlbaum Associates, 2002, pp. 542-547.

Rips, L.J. (1994). The Psychology of Proof: Deductive reasoning in human thinking. MIT Pres, Cambridge, Mass.