Reasoning by Assumption: Formalisation and Analysis of Human Reasoning Traces

Tibor Bosse[1], Catholijn M. Jonker[2], and Jan Treur[1]

[1] Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
`{tbosse, treur}@cs.vu.nl`
`http://www.cs.vu.nl/~{tbosse, treur}`
[2] Nijmegen Institute for Cognition and Information, Division Cognitive Engineering, Montessorilaan 3, 6525 HR
`C.Jonker@nici.ru.nl`

Abstract. This paper shows how empirical human reasoning traces can be formalised and automatically analysed against dynamic properties they fulfil. To this end, for the reasoning pattern called 'reasoning by assumption' a variety of dynamic properties have been specified, some of which are considered characteristic for the reasoning pattern, whereas some other properties can be used to discriminate between different approaches to the reasoning. These properties have been automatically checked for the traces acquired in experiments undertaken.

1 Introduction

Practical reasoning processes are often not limited to single reasoning steps, but extend to traces or trajectories of a number of interrelated reasoning steps over time. This paper presents experiments and an analysis for a pattern called 'reasoning by assumption'. This (non-deductive) practical reasoning pattern involves a number of interrelated reasoning steps, and uses in its reasoning states not only content information but also meta-information about the status of content information and about control. For this reasoning pattern human reasoning protocols have been acquired, analysed, formalised, checked on dynamic properties and compared. As a vehicle a temporal technique has been exploited which was already shown to be a useful analysis tool for reasoning processes in (Jonker and Treur, 2002).

Master Mind is a two player game of logic, which was invented in 1970-71 by Mordecai Meirowitz (Nelson). The goal of the game is to discover a secret code of three colored pegs, which can be obtained by making guesses and receiving information about the correctness of the guesses. Because of its protocol, the pattern of reasoning by assumption occurs frequently within this game. Therefore, the game of Master Mind (in a simplified version) will be the main case study within this paper.

Below, in Section 2 the underlying dynamic perspective on reasoning is discussed in some more detail, and focussed on the pattern 'reasoning by assumption'. Next, some more details of the temporal language used are described in

Section 3. In Section 4 it is shown how think-aloud protocols involving reasoning by assumption in the game of Master Mind can be formalised to reasoning traces. A number of the dynamic properties that have been identified for patterns of reasoning by assumption are shown in Section 5. For the acquired reasoning traces the identified dynamic properties have been (automatically) checked. The results of these checks are provided in Section 6. In addition, it is shown how logical relationships between dynamic properties at different abstraction levels can play a role in the analysis of empirical reasoning processes. Finally, Section 7 is a conclusion.

2 The Dynamics of Reasoning

Analysis of reasoning processes has been addressed from different areas and angles, for example, Cognitive Science, Philosophy and Logic, and AI. For reasoning processes in natural contexts, which are usually not restricted to simple deduction, dynamic aspects play an important role and have to be taken into account, such as dynamic focussing by posing goals for the reasoning, or making (additional) assumptions during the reasoning, thus using a dynamic set of premises within the reasoning process. Also dynamically initiated additional observations or tests to verify assumptions may be part of a reasoning process. Decisions made during the process, for example, on which reasoning goal to pursue, or which assumptions to make, are an inherent part of such a reasoning process. Such reasoning processes or their outcomes cannot be understood, justified or explained without taking into account these dynamic aspects. The approach to the semantical formalisation of the dynamics of reasoning exploited here is based on the concepts reasoning state, transitions and traces.

Reasoning state. A reasoning state formalises an intermediate state of a reasoning process. The set of all reasoning states is denoted by RS.

Transition of reasoning states. A transition of reasoning states or reasoning step is an element < S, S' > of RS x RS. A *reasoning transition relation* is a set of these transitions, or a relation on RS x RS that can be used to specify the allowed transitions.

Reasoning trace. Reasoning dynamics or reasoning behaviour is the result of successive transitions from one reasoning state to another. A time-indexed sequence of reasoning states is constructed over a given time frame (e.g., the natural numbers). Reasoning traces are sequences of reasoning states such that each pair of successive reasoning states in such a trace forms an allowed transition. A trace formalises one specific line of reasoning. A set of reasoning traces is a declarative description of the semantics of the behaviour of a reasoning process; each reasoning trace can be seen as one of the alternatives for the behaviour. In the next section a language is introduced in which it is possible to express dynamic properties of reasoning traces.

The specific reasoning pattern used in this paper to illustrate the approach is ‘reasoning by assumption’. This type of reasoning often occurs in practical reasoning; for example, in everyday reasoning, diagnostic reasoning based on causal knowledge, and reasoning based on natural deduction. An example of everyday reasoning by assumption is ‘Suppose I do not take my umbrella with me. Then, if it starts raining, I will get wet, which I don’t want. Therefore I’d better take my umbrella with me’. An example of reasoning by assumption in the context of a game of Master Mind is: ‘Suppose there is a red pin at position 1. Then, guessing the code [red-blue-white] would at least provide one ”correct” point. But if I try, it turns out I do not receive any ”correct” points. Therefore there is no red pin at position 1.’ Examples of reasoning by assumption in natural deduction are as follows. Method of indirect proof: ‘If I assume A, then I can derive a contradiction. Therefore I can derive not A’. Reasoning by cases: ‘If I assume A, I can derive C. If I assume B, I can also derive C. Therefore I can derive C from A or B’. Notice that in all of these examples, first a reasoning state is entered in which some fact is *assumed*. Next (possibly after some intermediate steps) a reasoning state is entered where *consequences* of this assumption have been *predicted*. Finally, a reasoning state is entered in which an *evaluation* has taken place; possibly in the next state the assumption is retracted, and conclusions of the whole process are added.

3 A Temporal Trace Language

In recent literature on Computer Science and Artificial Intelligence, temporal languages to specify dynamic properties of processes have been put forward; for example, (Dardenne, Lamsweerde and Fickas, 1993; Dubois, Du Bois and Zeipen, 1995; Herlea, Jonker, Treur, and Wijngaards, 1999). To specify properties on the dynamics of *reasoning* processes in particular, the temporal trace language TTL used in (Herlea et al., 1999; Jonker and Treur, 1998) is adopted. This is a language in the family of languages to which also situation calculus (Reiter, 2001) and event calculus (Kowalski and Sergot, 1986) belong, and was also succesfully used to analyse multi-representational reasoning processes in (Jonker and Treur, 2002).

Ontology. An ontology is a specification (in order-sorted logic) of a vocabulary. For the example reasoning pattern ‘reasoning by assumption’ in a game of Master Mind the state ontology includes unary relations such as assumed and rejected_code on sort INFO_ELEMENT and binary relations such as prediction_for, observation_result_for and holds_in_world_for on INFO_ELEMENT x INFO_ELEMENT. The sort INFO_ELEMENT includes specific domain statements such as at(red, 1), code(red, white, blue), answer(black, black, black).

Reasoning state. A (reasoning) state for ontology Ont is an assignment of truth-values {true, false} to the set of ground atoms At(Ont). The set of all possible states for ontology Ont is denoted by STATES(Ont). A part of the description of an example reasoning state S is:

assumed(code(red, white, blue)) : true
prediction_for(answer(black, empty, empty), code(red, white, blue)) : true
observation_result_for(answer(white), code(red, white, blue)) : true
rejected_code(code(red, white, blue)) : false

RS is the sort of all reasoning states of the agent. For simplicity in the formulation of properties WS is the set of all substates of elements of RS, thus WS is the set of all world states. The standard satisfaction relation $\models$ between states and state properties is used: S $\models$ p means that state property p holds in state S. For example, in the reasoning state S above it holds S $\models$ assumed(code(red, white, blue)).

Reasoning trace. To describe dynamics, explicit reference is made to time in a formal manner. A fixed time frame T is assumed which is linearly ordered. Depending on the application, for example, it may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers). A trace γ over an ontology Ont and time frame T is a mapping γ : T $\rightarrow$ STATES(Ont), i.e., a sequence of reasoning states γ_t (t $\in$ T)in STATES(Ont). The set of all traces over ontology Ont is denoted by $\mathit{\Gamma}$(Ont), i.e., $\mathit{\Gamma}$(Ont) = STATES(Ont)^T. The set $\mathit{\Gamma}$(Ont) is also denoted by $\mathit{\Gamma}$ if no confusion is expected.

Expressing dynamic properties. States of a trace can be related to state properties via the formally defined satisfaction relation $\models$ between states and formulae. Comparable to the approach in situation calculus, the sorted predicate logic temporal trace language TTL is built on atoms such as state(γ, t) $\models$ p, referring to traces, time and state properties. This expression denotes that state property p is true in the state of trace γ at time point t. Here $\models$ is a predicate symbol in the language (in infix notation), comparable to the Holds-predicate in situation calculus. Temporal formulae are built using the usual logical connectives and quantification (for example, over traces, time and state properties). The set TFOR(Ont) is the set of all temporal formulae that only make use of ontology Ont. We allow additional language elements as abbreviations of formulae of the temporal trace language. The fact that this language is formal allows for precise specification of dynamic properties. Moreover, editors have been developed to support specification of properties. Specified properties can be checked automatically against example traces to detect whether they hold.

4 The Experiment

Participants. Thirty subjects participated in the experiment. They were divided into two groups of 15. Group 1 consisted of 'AI-scientists', all working at the Department of Artificial Intelligence at the Vrije Universiteit Amsterdam. Group 2 consisted of 'non-scientists', a random set of friends and relatives of the authors. Some of them were students, but none of them had any background related to AI. Group 1 included 10 males and 5 females. Group 2 included 9 males and 6 females. The average age of both groups was approximately 28 years.

Method. The subjects were asked to solve a simplified game of Master Mind. Before starting the experiment, they were given the following instructions:

> *The opponent picks a secret code consisting of three pegs, each peg being one of eight colors. Your goal is to guess the exact positions of the colors in the code in as few guesses as possible. After each guess, the opponent gives you a score of exact and partial matches. For each of the*

pegs in your guess that is the correct color in the correct position, the opponent will give you an 'exact' point (represented by a black pin). If you score 3 black pins on a guess, you have guessed the code. For each of the pegs in the guess that is a correct color in an incorrect position, the opponent will give you an 'other' point (represented by a white pin). Together, the black and white pins will add up to no more than 3. Notice that the positions of the black and white pins do not necessarily relate to the positions of the colors. Within this specific experiment, ***one initial guess has already been done for you.*** *While doing the experiment, please think aloud, explaining each step you perform.*

For each participant, the *solution code* was the same, namely the combination [blue-white-red]. The *initial guess* mentioned above was always the combination [red-white-blue]. Hence, the provided answer corresponding to the initial guess was [black-white-white].

Table 1. Example human reasoning trace

Human transcript	**Formalisation**
Right? Okay. So, what I'm going to do now. I'm going to... I'm trying to find out which of the colors is in a good place, first. So, let's say I say it's the red one. Maybe.	focus_assumed(at(red, 1))
So, I'm going to put the red here. And then, change these two.	code_extention_for(code(red, blue, white), at(red, 1)) assumed(code(red, blue, white)) prediction_for(answer(black, black, black), code(red, blue, white))
[red-blue-white] *Okay, so this is your guess?* This is my guess.	to_be_observed_for(answer, code(red, blue, white))
Then my answer is like this... *[white-white-white]* *... two, and three.*	observation_result_for(answer(white, white, white), code(red, blue, white))
Okay, so it wasn't the red. Okay.	rejected_code(code(red, blue, white)) rejected_focus(at(red, 1))
I will always use these ones, apparently. Then, keep the white and exchange red and blue.	focus_assumed(at(white, 2)) code_extention_for(code(blue, white, red), at(white, 2)) assumed(code(blue, white, red)) prediction_for(answer(black, black, black), code(blue, white, red))
[blue-white-red] *Okay, so why do you do this?* I'm testing now if the white one is in the good position.	to_be_observed_for(answer, code(blue, white, red))
Okay. So then my answer is this. Congratulations! *[black-black-black]*	observation_result_for(answer(black, black, black), code(blue, white, red))

In Table 1 an example trace is shown, and the way in which it was formalised in order to automatically check their properties. The left column contains the human transcript, the right column contains the formal counterpart. The transcripts of all human reasoning traces can be found at:
http://www.cs.vu.nl/∼tbosse/mastermind/human-traces.doc.

5 Dynamic Properties

In this section a number of the most relevant of the dynamic properties that have been identified as relevant for patterns of reasoning by assumption are presented. Two categories of dynamic properties exist. The first category is specified by *characterising properties*. These are properties that are expected to hold for all reasoning traces. In contrast, the second category contains *discriminating properties*, properties that distinguish several types of traces from each other. Within each category, *global properties* (GP's, addressing the overall reasoning behaviour) as well as *executable properties* (EP's, addressing the step by step reasoning process) are given.

Characterising Properties

GP2 Correctness of Rejection
Everything that has been rejected does not hold in the world situation.
$\forall\gamma{:}\Gamma \quad \forall$t:T $\forall$A:INFO_ELEMENT
state(γ,t) $\models$ rejected_code(A) $\Rightarrow$
state(γ,t) $\not\models$ holds_in_world_for(answer(black, black, black), A)
This property holds for all traces, leading to the conclusion that none of the participants makes the error of rejecting something that is true.

EP5 Observation Initiation Effectiveness
For each prediction an observation will be made.
$\forall\gamma{:}\Gamma \quad \forall$t:T $\forall$A,B:INFO_ELEMENT
state(γ,t) $\models$ prediction_for(B,A)
$\Rightarrow$ [$\exists$t':T $\geq$ t:T state(γ,t') $\models$ to_be_observed_for(answer, A)]
This property holds for all traces, leading to the conclusion that in every case that a prediction was made, this was followed by a corresponding observation.

EP6 Observation Result Effectiveness
If an observation is made the appropriate observation result will be received.
$\forall\gamma{:}\Gamma \quad \forall$t:T $\forall$A,B:INFO_ELEMENT
state(γ,t) $\models$ to_be_observed_for(answer, A) $\wedge$
state(γ,t) $\models$ holds_in_world_for(B,A)
$\Rightarrow$ [$\exists$t':T $\geq$ t:T state(γ,t') $\models$ observation_result_for(B,A)]
This property holds for all traces. Thus, in all traces, the opponent provided the correct answers.

Discriminating Properties

GP5 Correctness of Assumption
Everything that has been assumed holds in the world situation.
$\forall\gamma{:}\Gamma \quad \forall$t:T $\forall$A:INFO_ELEMENT
state(γ,t) $\models$ assumed(A) $\Rightarrow$
state(γ,t) $\models$ holds_in_world_for(answer(black, black, black), A)

This property only holds in four of the 30 cases. By checking it, the subjects that made only correct assumptions can be distinguished from those that made some incorrect assumptions during the experiment. Put differently, the subjects that immediately make the right guess are distinguished from those that need more than one guess.

GP7 Observation Effectiveness

For each assumption, the agent eventually obtains the appropriate observation result.

$\forall \gamma{:}\Gamma \quad \forall \mathrm{t{:}T}\ \forall \mathrm{A,B{:}INFO\_ELEMENT}$

$\mathrm{state}(\gamma,\mathrm{t}) \models \mathrm{assumed(A)} \wedge \mathrm{state}(\gamma,\mathrm{t})\ |{==}\ \mathrm{holds\_in\_world\_for(B,A)}$

$\Rightarrow [\ \exists \mathrm{t'{:}T} \geq \mathrm{t{:}T}\ \mathrm{state}(\gamma,\mathrm{t'}) \models \mathrm{observation\_result\_for(B,A)}\ ]$

This property holds for all but three of the traces. In these three cases people make an assumption that cannot be right, according to the information they have. However, they correct themselves before they decide to observe the answer to this wrong assumption. Thus, the answer to the incorrect assumption is never obtained.

GP9 Initial Assumption

The first focus assumption made was at(red, 1).

$\forall \gamma{:}\Gamma \quad \exists \mathrm{t{:}T}$

$\mathrm{state}(\gamma,\mathrm{t}) \models \mathrm{focus\_assumed(at(red,\ 1))}$

$\wedge\ [\ \forall \mathrm{t'{:}T} < \mathrm{t{:}T}\ \forall \mathrm{A{:}INFO\_ELEMENT}$

$\mathrm{state}(\gamma,\mathrm{t'}) \models \mathrm{focus\_assumed(A)} \Rightarrow \mathrm{A} = \mathrm{at(red,\ 1)}\ ]$

This property holds in 18 of the 30 cases. Thus, 18 participants started reasoning by assuming that the red pin was at position 1. Given the fact that they wanted to keep one of the colors at its initial position, and all three options have an equal probability to be the solution, this seems a logical choice, because it is the first pin they encounter when looking from left to right. Nevertheless, there were still 12 participants that started in a different way.

EP4 Prediction Effectiveness

For each assumption that is made a prediction will be made.

$\forall \gamma{:}\Gamma \quad \forall \mathrm{t{:}T}\ \forall \mathrm{A{:}INFO\_ELEMENT}$

$\mathrm{state}(\gamma,\mathrm{t}) \models \mathrm{assumed(A)}$

$\Rightarrow [\ \exists \mathrm{t'{:}T} \geq \mathrm{t{:}T}\ \exists \mathrm{B{:}INFO\_ELEMENT}\ \mathrm{state}(\gamma,\mathrm{t'}) \models \mathrm{prediction\_for(B,A)}\ ]$

This property holds in 26 of the 30 cases. So in four cases the subjects make an assumption for which no prediction is made. Three of these four traces have already been discussed at GP7. The fourth trace involves a situation where a person has the following reasoning pattern: "...Let's use one of the colors twice. What would happen in that case? Well, I don't know. Let's just see what happens..." Hence, the subject tries a code of which he intuitively thinks that it is an intelligent guess, without really understanding why. Therefore, he does not make a prediction.

6 Results

A special piece of software has been developed that takes a formally specified property and a set of traces as input, and verifies whether the property holds for the traces (see Bosse, Jonker, Schut, and Treur, 2004). By means of this checking software, all specified properties have been checked automatically against all traces to find out whether they hold. In Table 2 an overview of the results is shown. In this table, an X indicates that the property holds for that particular trace. The final row provides the number of guesses needed by each subject to solve the problem.

Table 2. Overview of the results: traces against properties

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
GP2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
GP5	X	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	X	-	-	-	-	-	-	X	-	-	-	-
GP7	X	X	X	X	X	-	X	X	X	X	X	X	X	X	-	X	X	-	X	X	X	X	X	X	X	X	X	X	X	X
GP9	-	-	-	-	-	X	X	X	X	-	X	X	X	-	-	X	X	X	-	X	X	X	X	-	X	-	-	X	X	X
EP5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
EP6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
EP4	X	X	X	X	X	-	X	X	X	-	X	X	X	X	-	X	X	-	X	X	X	X	X	X	X	X	X	X	X	X
steps	1	2	3	3	3	3	3	2	3	3	2	3	2	1	3	3	3	3	1	3	2	2	3	3	2	1	3	3	3	2

As can be seen in the table, all characterising properties indeed hold for all traces. The discriminating properties only hold for some of the traces, which allows making a distinction between different classes of reasoners.

In addition to the above, logical relationships have been identified between properties at different abstraction levels. An overview of the logical relationships relevant for overall property GP7 is depicted as an AND-tree in Figure 1.

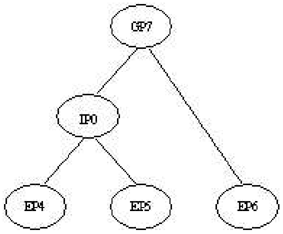

Fig. 1. Logical relationships between dynamic properties

For example, the relationship at the highest level expresses that IP0& EP6 ⇒ GP7 holds. Here, IP0 is an *intermediate property*, expressing the dynamics of the reasoning between two milestones:

IP0 Assumptions lead to Observation Initiation

For each assumption that is made a prediction will be made.

$\forall \gamma{:}\Gamma \quad \forall t{:}T\ \forall A{:}\mathrm{INFO\_ELEMENT}$

$\mathrm{state}(\gamma,t) \models \mathrm{assumed}(A)$

$\Rightarrow [\ \exists t'{:}T \geq t{:}T\ \mathrm{state}(\gamma,t') \models \mathrm{to\_be\_observed\_for}(\mathrm{answer}, A)\ ]$

Intermediate properties address smaller steps than global properties do, but bigger steps than executable properties do. At a lower level, Figure 1 depicts the relationship EP4 & EP5 ⇒ IP0.

Notice that the results given in Table 2 validate these logical relationships. For instance, in all traces where EP4, EP5 and EP6 hold, also GP7 holds. Such logical relationships between properties can be very useful in the analysis of empirical reasoning processes. For example, if a given person does not obtain the appropriate observation result for her assumption (i.e. property GP7 is not satisfied by the reasoning trace), then by a refutation process it can be concluded that either property IP0, or property EP6 fails (or both). If, after checking these properties, it turns out that IP0 does not hold, then either EP4 or EP5 does not hold. Thus, by this example refutation analysis it can be concluded that the cause of the unsatisfactory reasoning process can be found in either EP4 or EP5. In other words, either the *Observation Initiation* mechanism fails (EP5), or the *Prediction* mechanism fails (EP4).

In this section, only one logical relationship is shown. However, many more global, intermediate, and executable properties for the pattern of reasoning by assumption, as well as the relationships between them can be found at the following URL: http://www.cs.vu.nl/~tbosse/mastermind/properties-and-relationships.doc.

7 Conclusion

This paper shows how given instances of empirical human reasoning traces can be formalised and automatically analysed against dynamic properties they fulfil. To this end a variety of dynamic properties have been specified, some of which are considered characteristic for the reasoning pattern ‘reasoning by assumption’, whereas some other properties can be used to discriminate between different approaches to the reasoning. For the Master Mind experiments undertaken, properties of the first, characteristic, type indeed hold for the acquired reasoning traces. Properties of the latter, discriminating type hold for some of the traces and do not hold for other traces: they define subsets of traces that collect similar reasoning approaches.

In addition to empirical traces, the analysis method can be applied to traces generated by simulation models. Dynamic properties found relevant for human traces can be used to validate a simulation model, by generating a number of simulation runs and checking the dynamic properties for the resulting traces. This type of validation has been exploited to validate a simulation model for reasoning by assumption to solve the wise men puzzle in (Jonker and Treur, 2003). Moreover, in (Bosse, Jonker and Treur, 2003) a similar analysis approach has been used to validate a simulation model for controlled multi-representational reasoning involving arithmetic, geometric and material representations.

References

1. Bosse, T., Jonker, C.M., and Treur, J., Simulation and analysis of controlled multi-representational reasoning processes. *Proc. of the Fifth International Conference on Cognitive Modelling, ICCM'03.* Universitats-Verlag Bamberg, 2003, pp. 27-32.
2. Bosse, T., Jonker, C.M., Schut, M.C., and Treur, J., Modelling Shared Extended Mind and Collective Representational Content. In: *Proc. of the 24th International Conference on Innovative Techniques and Applications of Artificial Intelligence.* Lecture Notes in AI, Springer Verlag. To appear, 2004..
3. Dardenne, A., Lamsweerde, A. van, and Fickas, S. (1993). Goal-directed Requirements Acquisition. *Science in Computer Programming*, vol. 20, pp. 3-50.
4. Dubois, E., Du Bois, P., and Zeippen, J.M. (1995). A Formal Requirements Engineering Method for Real-Time, Concurrent, and Distributed Systems. In: *Proceedings of the Real-Time Systems Conference, RTS'95.*
5. Herlea, D.E., Jonker, C.M., Treur, J., and Wijngaards, N.J.E. (1999). Specification of Behavioural Requirements within Compositional Multi-Agent System Design. In: F.J. Garijo, M. Boman (eds.), *Multi-Agent System Engineering, Proc. of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99.* Lecture Notes in AI, vol. 1647, Springer Verlag, 1999, pp. 8-27.
6. Jonker, C.M., and Treur, J. (1998). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. In: W.P. de Roever, H. Langmaack, A. Pnueli (eds.), *Proceedings of the International Workshop on Compositionality, COMPOS'97.* Lecture Notes in Computer Science, vol. 1536, Springer Verlag, 1998, pp. 350-380. Extended version in: *International Journal of Cooperative Information Systems*, vol. 11, 2002, pp. 51-92.
7. 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.
8. Jonker, C.M., and Treur, J. (2003). Modelling the Dynamics of Reasoning Processes: Reasoning by Assumption. *Cognitive Systems Research Journal.* In press, 2003.
9. Kowalski, R., and Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing*, 4:67-95, 1986.
10. Nelson, T. *A Brief History of the Master MindTM Board Game.* http://www.tnelson.demon.co.uk/mastermind/history.html
11. Reiter, R. (2001). Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, 2001.