The Character of Human Behavior Representation and its Impact on the Validation Issue *

Rainer Knauf and **Ilka Philippow**
Ilmenau Technical University, Faculty of Computer Science and Automation
PO Box 10 05 65, 98684 Ilmenau, Germany
rainer.knauf@theoInf.tu-ilmenau.de

Avelino J. Gonzalez
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2450, USA
gonzalez@pagasus.cc.ucf.edu

Klaus P. Jantke
German Research Center
for Artificial Intelligence Ltd.
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
jantke@dfki.de

Abstract

Human behavior representations (HBRs, for short) are a challange for system developers and validation technologies. Authors agree that classical technologies of knowledge processing can not adaequately be applied to HBRs and publish first ideas towards appropriate architectures. In fact, their suggestions are usually driven by a particular application field.
To illustrate some characteristics and their impact on validation technologies we introduce a tiny example and outline characteristic features. The upcoming exemplary insights nicly confirm the problems reported by other authors and raise at least two major insights: (1) First research steps should aim at a classification of HBR systems that need different architechtures as well as validation technonlogies. (2) The key to successful HBR systems is an integrated approach of developing and validating HBR systems at all stages.

Introduction

Here, we outline some basic features of knowledge bases that describe human behavior in tactical situations. HBR is widely used in simulation–based training, but also in entertainment and control.

Proceeding from a tiny exmaple, we collect some typical features of HBR. The basic objective is to work out the particulars of these models that have an impact to validation. More specific, this paper outlines some general insights on how to specify the general validation framework as described in (Knauf 2000) and (Knauf, Philippow, Gonzalez, and Jantke 2000), e.g.

The authors are aware, that any approach just created for a "toy example" is still far away from addressing the real challanges of HBR research. As claimed in (Banks and Stytz 2000), there is still "no architecture to house component technologies for experimentation to foster development of human behavior technologies their incorporation ... into operational ... systems." Thus, even a tiny example might be a first step to close that gap.

Today's operational HBRs adopt a large varity of AI representation technologies like logic, rules, frames, and objects, as well as a large variety of knowledge processing techniques like Intelligent Agents, Fuzzy Logic, Neuronal Networks, Case Based Reasoning, and Basian Networks applied in a various architectures like Blackboard Architectures and Distributed Agent Architectures (cf. (Banks and Stytz 2000)).

One can argue, that there is no need for a particular HBR, because there are some modern AI technologies and tools that seem to be applicable to HBR. Unfortunately, these technologies have inherent gaps with respect to the representation of human behavior. Objects, for example, usually have a non-reactive behavior or at least a very simplistic reactive behavior. Tools creating intelligent agents, on the other hand, focus on the total creation of a new agent that is desingned by some purpose.[1] and on the communications among multi-agents (cf. (Patrone and Nardo 2000)).

The huge amount of approaches that are not compatibel to each other makes clear that there is still no accepted standard. Thus, HBR research is aimed at the development of "unifying visions". This incorporates integrated environemts for the development of HBRs.

Unfortunately, topical literature does not care about quality management. The assumption that high quality HBRs are a result of using the right approach in a comprehensive environment is simply wrong. This is one of the lessons we should have learnt from the past. The development of HBRs has to be accompanied by validation and verification of all aspects at the different stages of development and implementation, starting at the high level design and going down to the operational details.

Here, we just try to conclude some general characteristic features of HBRs. For this purpose, an illustrating "toy example" still might be legal. Further research has to be aimed at classifying HBRs, idetifying their characteristic features, deriving appropriate

---

[1] This, hopefully, does not apply to humans.

architectures, and creating development - and validation environments.

An Example Tactical Situation

As a tiny example of a tactical situation we consider a driver who has to drive a vehicle through a labyrinth as illustrated in figure 1. A typical component of tactical knowledge is an adequate reaction to unexpected situations. Here, such situations are realized by some places within the labyrinth where an opponent is located. The job of the drive consists in driving the vehicle from the entrance to the exit of the labyrinth

1. without knowing the structure of the labyrinth in advance and
2. without being watched by the opponent.

More specific, the driver has to develop and apply any strategy for systematically searching a path from the entrance $[1,1]$ to the exit $[12,1]$

- without knowing in advance the position of the exit (he will be "told" that the exit is reached as soon as this is the case) and
- without performing unnecessary loops.

As soon as there is a straight line (light ray) from an opponent's position (marked by ⊕) to the vehicle (marked by ◇), the driver has to move the vehicle immediately behind any wall to make sure that he/she can't be seen longer than one time unit Δt.[2]

The movement is performed in discrete steps within an $[x, y]$ array

- beginning at the time t_0 at the position $p(t_0) = [1,1]$ and ending after n steps with $p(t_n) = [12,1]$
- in steps to any of the directions *north*, *east*, *south*, and *west*
- that need one time unit $\Delta t = t_i - t_{i-1}$ each.

A Behavior Description

The Real Human Behavior

As with any interactive system, a HBR architcture development consists of

1. exploring the inputs the behavior depends on,
2. developing an inference engine that calculates appropriate reactions, i.e. outputs, and
3. performing the choosen reaction(s).

A particular feature of HBR is that the reactions do not just depend on inputs, they also depend on previous behaviors.[3]

Again, this might still be far from reality. This approach, for example, is based on the (in many cases unrealistic) assumtion that the environment is fix and does not attempt to determine the user's goal (as done with a cognitive model). The HBR determines a next user's action just by considering the actual status of the environment and the past user's actions.

More complex HBR include re-using concepts of a pre-developed behaviors ("learning from the past") and/or some "translater concepts" between a complex simulated enironment and the HBR. (cf. (Patrone and Nardo 2000)).

Here, human behavior, i.e. the steps performed by the driver, can be formally described by a n-elemental list of triples $[x, y, w]$ whose i-th element is $[x_i, y_i, w_i]$, where

- x_i and y_i indicate the position at time t_i and
- w_i ("watch - parameter") indicates whether or not the vehicle can be seen by an opponent at the time t_i, i.e.

$$w_i = \begin{cases} true & \text{, if the vehicle can be seen} \\ & \text{in direct line of sight } (los) \\ false & \text{, otherwise} \end{cases}$$

The Human Behavior Representation

One possible HBR that claims to describe all possible "correct" human behaviors and is the subject to validation could be characterized as follows.

At the time t_i, look through the list of previous positions $[x_j, y_j, w_j]$ $(j < i)$. Without limiting the generality of the approach we postulate the priorities of the four different directions of possible steps clockwise , i.e. *north* has the highest priority, followed by *east*, *south*, and *west*. At a time t_i, we look through the list of previous positions $[x_j, y_j, w_j]$:

1. If $w_i = false$,
   - For all $j < i$ with $[x_j, y_j, w_j] = [x_i, y_i, w_i]$ go through the priority list of directions and determine the one with the the highest priority which does **not** lead to a (former) position $[x_{j+1}, y_{j+1}, w_{j+1}]$. In case there is such a possible step, perform it.
   - Otherwise, perform a step in the direction *north*.
2. Otherwise,
   - For all $j < i$ with $[x_j, y_j, w_j] = [x_i, y_i, w_i]$ go through the priority list of direction and determine the one with the the highest priority which does **not** lead to a position $[x_{j+1}, y_{j+1}, w_{j+1}]$ and does **not** lead to any position with $w_{i+1} = true$. In case there is such a possible step, perform it.
   - Otherwise, perform a step in the highest priority direction that does **not** lead to a position with $w_{i+1} = true$.

Note, that

---

[2] In the borderline case that the light ray is tangential to a wall's end, the vehicle is also considered to be seen by the opponent.

[3] This quality is comparable to the one of sequential automata with respect to combinatorical automata in automata theory.

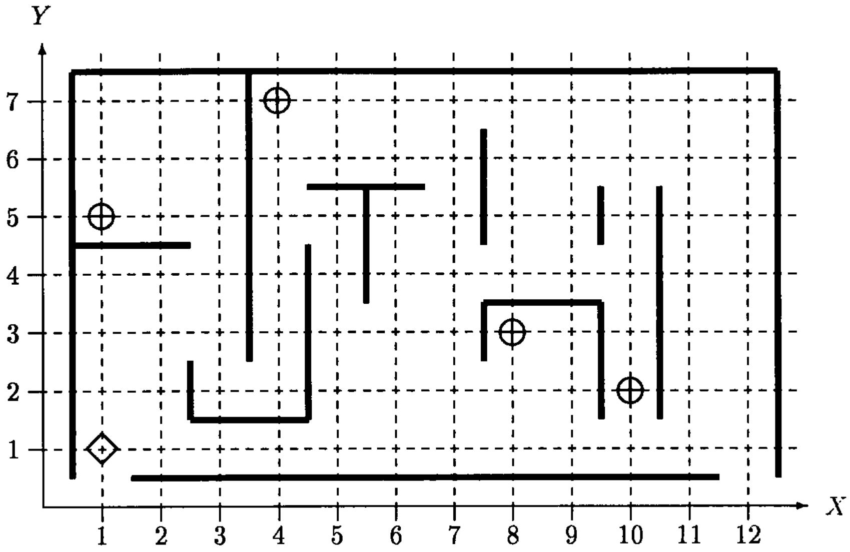

Figure 1: The Example Labyrinth

- even the authors are not sure, whether or not this behavior representation really describes all possible "correct" behaviors[4] but, on the other hand,
- this is not an issue here since it's not the objective of our research.

Characteristic Features

Essential characteristic features of any tactical behavior description that are the issues of specifying general validation technologies are as follows.

1. Time plays an important role. A valid activity at a certain time t_i depends on former situations and activities.

   In our example HBR, a valid step at a particular time t_i depends on the history of previous positions at times t_j $(j < i)$, i.e. whether or not the vehicle already has been at the same position and, if this is true, whether or not it has been watched by an opponent at this previous position.
2. This leads to the second typical feature: The "input of human behavior" contains an particular component "unexpected events" that is represented as the "watch parameter" w in our example.
3. There is an inherent non-determinism, i.e. there may be several valid activities at a certain time t_i.

   In our example, we defined a priority list. The priority of directions is defined by a sequence of directions that is somehow arbitrary. It could have been defined by any other sequence as well. The only necessity is to have any priority regulation to perform the search systematically, i.e.
4. there may be several valid HBRs.

Impacts on the Validation Technology

In our tiny example problem there might be some "meta-model" which is able to check the validity of an HBR. In this case validation turns to become a verification issue. In practice, on the other hand, there is a large range of problem classes that do not have a (formally) well-defined description of validity requirements the HBR has to meet. These are the "interesting" problems in general and for the validation research in particular.

This insight makes clear that the general idea of our former TURING Test approach as described in (Knauf 2000) e.g., does not have to be changed.

Since (Knauf 2000) aims at the validation of "static" knowledge bases and, more specific, at rule sets, it has to be adapted for using in HBR validation, naturally.

The most essential issue is the inclusion of a parameter that describes the time issue. Some first ideas to formalize a time dependence by representing a historical sequence of input/output pairs is sketched in (Knauf 2000), but just as a "launching pad for future research".

The second issue, unexpected events, charactrizes just a certain kind of system inputs. Generally, it might be useful to represent this part of the system input explicitely (in the test cases for the TURING Test, e.g.).

The third of the insights above, the inherent non-determinism, does not really raise a necessity to ad-

---

[4] In particular, a behavior like that might be bad in case of escaping from a "watched" position behind a wall. In this case, usually, escaping to a new position might always be better than escaping back. But this strategy is not explicitely implemented in the behavior representation.

apt the general approach. Since Artificial Intelligence problems are usually characterized by their non-determinism, this issue has also been included in the (relational) approach of (Knauf 2000).

The problem that there is usually not "the one and only one" valid HBR (cf. fourth of the issues listed above) does not appear in our validation approach. The "knowledge base" (the HBR) is not compared with some "meta-model", i.e. some general requirements. Instead of this, it is compared with real human behaviors of the experts in a panel. Their knowledge (their model) is not represented formally.

Formalizing Time Depency

A crucial drawback of the approach developed so far is its time independency and its ahistorical character.

A simple way to handle the time issue as suggested in (Knauf 2000) is to introduce a (non-empty) sequence of pairs $(I \times O)^+$ that contain an element of some input set I and some output set O each.

Our example also includes a particular part of the input describing the unexpected events that has been represented explicitely by the parameter w. Consequently, the behavior (as well as its representation) can be formally described as a sequence of triples $(X \times Y \times W)^+$.

In practice, especially in simulation and training applications, some more qualified ways to handle the time issue are necessary.

In (von der Lippe, McCormack and Kalphat 2000), e.g., the authors list some temporal information that has to be represented as well as processed within their application field:

- There is a need to represent some requested and/or provided time duration between performing some action and enjoying it's result.
- Based on this issue, there must me some "time processor" that composes a time schedule of actions which aims at a certain goal.

  This suggests itself to adopt temporal logic approaches as used for planning systems.
- The HBR system has to be able to conclude in advance reactions of other agents (opponents, e.g.) to the particular possible actions of the HBR.

Their analysis of existing (and well reseached) time processing (planning) approaches like the *Situation Calculus* (cf. (Gerevini 1997)) and state-based approaches like STRIPS (cf. (Shoham and McDermott 1988)), e.g., that are both based on a temporal logic called *Intervall Algebra* reveals their "limited abilities to handle external and simultanious events." Consequently, they started some research on application oriented approaches that overcome these drawbacks and present some first results in (von der Lippe, McCormack and Kalphat 2000).

However, to draw conclusions to the validation for HBR systems in general , HBR developments for particluar application fields like the one above does not have to be considered – at least for now.

Later, we should develop qualitative classes of HBR systems that need to be distinguished concerning both their development and validation. Moreover, both system development and system validation should be performed within one development environment. We are convinced that such an integrated environment is the key to the success of HBR systems.

Overcoming the Combinatorical Explosion of the Turing Test

A first step to adapt the TURING Test technology developed so far is to define what test cases are. One approach is to use initial segments of $(X \times Y \times W)^+$ as test cases.

As a result of former research we learnt that even in the static approach without considering the time issue the number of test cases in the so called *reasonable set of test cases* $ReST \subseteq (I \times O)$ is still pretty large for interesting application domains.

Of course, this problem gets worse for the dynamic approach that considers not just single input-output pairs as test cases, but *sequences* of *triples* $ReST \subseteq (X \times Y \times W)^+$.

One idea to overcome this problem is to define *equivalence classes* of such initial segments. This gives us the opportunity to consider just one representant of each equivalence class as a potential test case.

Of course, a useful definition of whether or not 2 (formally different) test cases can be considered as equivalent heavely depends on the application domain.

In our tiny example, the human behavior, i.e. the next step to be performed by the vehicle at a time t_i, depends on whether or not the vehicle has been at the same position before. If yes, it does not matter, at which time t_j $(j < i)$ this has been the case. Therefore, in this application the "secret" of limiting $ReST$ consits in considering the initial segments of $(X \times Y \times W)^+$ not as a sequence of triples, but as a set of triples. Therefore, two test cases, i.e. two initial segments, can be considered as equivalent, iff they contain the same triples, independently on the postions and the frequency these triples occure within the segments.

Summary and Conclusions

HBR rearch is a very recent subject and there are just some first approaches for particular application fields like simumlation and training, e.g.

Authors with some experience in developing HBR point out that all these components should be developed independently from each other and using an appropreate technology each. In (Patrone and Nardo 2000), e.g., the authors point out some "lessons learnt" that include the following:

1. "Do not tie the logical definition of a behavior to a specific simulation implementation."

2. "Do not hard-code the reasoning terms of the behavior creation language."

Since 1 is the (old) idea of representing knowledge explicitely in an (as far as possible) domain independent manner, and 2 is the (also old) idea of doing that at a high level that makes modifications easy, even for non-system developers, HBR has to make use of AI technologies, of course.

There are two essential insights concerning HBR compared with "classical knowledge representations":

1. Human behavior is comparable to the model of sequential automata, i.e. its output does not just depend on its input, but also on its history.

   This is due to the fact that humans are able to leran from history. They do have experiences and they use it for their interaction with the environment.

2. The role of experience might be different depending on the class of problem a human has to solve. There are problems for which the time any experience has been made matters and there are problems for which this issue doesn't matter. For HBR this means, the "history" of (former) behavior

   (a) has to be hold
      - either as a sequence of pairs [*input*, *outpunt*]
      - or as a set of triples [*input*, *output*, *time*], where each output is associated to an input and to a time or

   (b) can be hold just as a set of (former) input/output pairs.

The second insight leads to another possible systematization criterion: There may be behaviors where time can be considered going on in discrete steps of a certain time unit and, on the other hand, there are behaviors where time runs countinously. For HBR, this means to represent the time

1. just by introducing the sequence of input/out pairs (instaed of a set) or
2. by introducing triples [*input*, *output*, *time*]

An approach to represent human behavior is based on the belief that a user's intention is a result of events occuring in the environment, the experiences he/she made, and the goals he/she is trying to obtain (cf. (Banks and Stytz 2000)). Thus, the HBR is a collection of sequences of actions that the human can perform.

In fact, all authors claim some drawbacks with existing approaches to handle typical HBR problems like time processing, e.g. To overcome these drawbacks they developed some application oriented solutions. This raises the insight that a very first step towards HBR development and validation approaches is distinguishing some different classes of HBRs.

As an example for the usefulness of a systematization is the fact that the combinatorical explosion of test cases can be limited by classifying the history as a one where sequences don't matter.

We completely agree with most of the authors who claim that we do need some integrated approach, i.e. integrated methodology, integrated tools, and integrated system development.

In the past, we considerd the validation issue of (monolithic) systems that are already in use – some of them even many years. Here, on the other hand, we ask for validation technologies of (obviously hybrid) systems that are still a subject of fundamental research and development. This opens the opportunity to accompany all aspects at all different stages of this process by validation and verfication, starting at the high level design and going down to the operational details.

References

Banks, S.B.; Stytz, M.R. 2000. Advancing the State of Human Behavior Representation for Modeling and Simulation: Technologies and Techniques. In: *Proc. 9th Conf. on Computer Generated Forces*, Orlando, FL, USA, University of Central Florida, Division of Sponsored Research, May 2000.

Gerevini, A. 1997. Reasoning About Time and Actions in AI: Major Issues. Stock, O. (ed): *Spatial and Temporal Reasoning*, Kluwer Academic Publishers, 1997.

Jantke, K.P.; Knauf, R.; Abel, T. 1997. The Turing test approach to validation. In: *Proc. Workshop on Validation, Verification & Refinement of AI Systems and Subsystems (W32)* of the *Int. Joint Conf. on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, pp. 35-45. Aug. 1997.

Knauf, R. 2000. *Towards Validation of Rule-Based systems - A Complete Methodology.* Habilitation Thesis, Ilmenau Technical University, Faculty of Computer Science and Automation, 2000.

Knauf, R.; Philippow, I.; Gonzalez, A.J.; Jantke, K.P. 2000. Towards Validation of Rule-Based Systems - The Loop is Closed. In: *Proc. 13th Int. Florida Artificial Intelligence Research Society Conf. 2000* (FLAIRS-2000), Orlando, FL, USA, pp. 331-335. May 2000.

PatroneD.M.; Nardo, A. 2000. A Domain– and Simulation Independent Architecture for Creating Simulates Object Behaviors. In: *Proc. 9th Conf. on Computer Generated Forces*, Orlando, FL, USA, University of Central Florida, Division of Sponsored Research, May 2000.

Shoham, Y.; McDermott, D. 1988. Problems in Formal Temporal Reasoning. In: *Artificial Intelligence*, 36, 1988.

von der Lippe, S.; McCormack, J.S.; Kalphat M. 2000. Embracing Temporal Relations and Command and Control in Composable Behavior Technologies. In: *Proc. 9th Conf. on Computer Generated Forces*, Orlando, FL, USA, University of Central Florida, Division of Sponsored Research, May 2000.