RESEARCH ARTICLE

Modeling learner-controlled mental model learning processes by a second-order adaptive network model

Rajesh Bhalwankar[1], **Jan Treur**[2]*

[1] Work and Social Psychology Department, Maastricht University, Maastricht, Netherlands, [2] Social AI Group, Vrije Universiteit Amsterdam, Amsterdam, Netherlands

* j.treur@vu.nl

Abstract

Learning knowledge or skills usually is considered to be based on the formation of an adequate internal mental model as a specific type of mental network. The learning process for such a mental model conceptualised as a mental network, is a form of (first-order) mental network adaptation. Such learning often integrates learning by observation and learning by instruction. For an effective learning process, an appropriate timing of these different elements is crucial. By controlling the timing of them, the mental network adaptation process becomes adaptive itself, which is called second-order mental network adaptation. In this paper, a second-order adaptive mental network model is proposed addressing this. The first-order adaptation process models the learning process of mental models and the second-order adaptation process controls the timing of the elements of this learning process. It is illustrated by a case study for the learner-controlled mental model learning in the context of driving a car. Here the learner is in control of the integration of learning by observation and learning by instruction.

OPEN ACCESS

Citation: Bhalwankar R, Treur J (2021) Modeling learner-controlled mental model learning processes by a second-order adaptive network model. PLoS ONE 16(8): e0255503. https://doi.org/10.1371/journal.pone.0255503

Editor: Hocine Cherifi, Unviersity of Burgundy, FRANCE

Received: February 24, 2021

Accepted: July 16, 2021

Published: August 24, 2021

Data Availability Statement: In the Appendix within the paper.

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

1 Introduction

To describe the mental processes involving learning and problem solving in humans, often mental models are used, e.g., [1–16]. As a specific case, mental models of devices and their usage are formed to be able to adequately use these devices, e.g., [17, 18]. It is an interesting challenge to determine how mental models are formed or learnt, and how to control such learning processes. Computational models which represent such processes are almost absent, e.g., [19–21]. One exception is [8] in which a production rule modeling format is used to simulate students' construction of energy models for learning physics. In general, however, research into how mental models develop or are learnt and how that is controlled, is hard to find.

The current paper proposes such a computational model for mental model learning and its control, based on multi-order adaptive network-oriented modeling [22, 23]. It is illustrated by a case study for learning how a car works and how to drive it. A driver's mental model and how it can be learnt in an effective manner can be a basis for the designing virtual pedagogical agents, and for support of a driver by adaptive automation in a car.

Network-oriented modeling for adaptive networks [22–24] is an effective approach to model the adaptive mental processes as an adaptive interplay of mental states. Here the connections between the mental states change based on specific adaptation principles such as Hebbian learning [25]. Learning mental models involves such adaptation, but it also involves controlling this learning; the latter is a form of second-order adaptation. The network-oriented modeling approach from [22–24] covers such multi-order adaptive processes.

Then the picture is that a mental model can be modeled as a base network and learning the mental model can be modeled as (first-order) adaptation of this base network. In addition, the controlling of this learning process can be modeled as second-order adaptation, which adapts the first-order adaptation. In this way, a three-level second-order adaptive network architecture for mental model development is obtained. It is illustrated here for learning a mental model, by a learner-controlled interplay of observational and instructional learning in a case study for learning how a car works and how to drive it.

Part of this work was addressed in a preliminary form in [26]. However, the current paper extends this by more than 110%. In the current paper, in particular, (1) the description of the computational model and its background (Sections 2 and 4) has been extended much so that now a much more detailed design description is provided and (2) also an extensive analysis of equilibria of the model is now described (Section 6) which has been conducted to obtain a more solid basis for the implemented model by verification (Section 6 on equilibrium analysis of the model in the current paper is completely new). All this was not addressed in [26].

In the paper, Section 2 presents a brief literature overview. In Section 3, the design of the proposed second-order adaptive network architecture is presented, addressing the controlled interplay between observational and instructional learning of mental models. Section 4 presents a refinement of this architecture, addressing the case study of learner-controlled integration of observational and instructional learning. Simulation results for an example scenario can be found in Section 5. In Section 6, a detailed analysis of equilibria is addressed by which verification of the model was performed. Section 7 is for discussion.

2 Overview of background knowledge on mental models

Mental models are being studied in Cognitive and Social Sciences, as well as in Educational Sciences for a long time, e.g., [1–16, 27–35]. A famous quote of Craik from his 1943 book mentions:

> ‘If the organism carries a “small-scale model” of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilise the knowledge of past events in dealing with the present and future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it.’
>
> (Craik [2], p. 61)

He writes that such internal models use a certain *relation-structure* that makes the mental model work in a way similar to how the real world works:

> ‘By “relation-structure” I do not mean some obscure non-physical entity which attends the model, but the fact that it is a physical working model which works in the same way as the process it parallels. . .
>
> (Craik [2], p. 51).

Within educational science, the term model-based learning is used for learning based on constructing coherent mental models [1, 9, 36–38]. Buckley formulates this as:

> 'Model-based learning is a dynamic, recursive process of learning by building mental models.'
>
> (Buckley [1])

More specifically, the following elements play an important role in such learning.

2.1 Learning by observation

Observational learning occurs when observation of others is the main source for the formation of a mental model. For example, the trainees see someone else perform a target behavior and then attempt to imitate or reenact it; e.g., [36, 39]. Demonstration is an often applied method to let others learn a specific motor task. This type is called observational motor learning. Empirical research has found that observational motor learning improves action perception and motor execution. From a neuroscientific perspective, mirror neurons are considered responsible for the ability to learn by observing and imitating others, e.g., [40–42].

2.2 Learning by instruction

Instructional learning assumes that instructions from an expert instructor are supportive for learning. For a beginner, only learning by discovery or observation often involves much trial and error; e.g., [14, 15]. Hence, instructions from an expert are considered a useful addition to build effective mental models. This is supported by a format of scaffolded model-based learning in which many supporting actions such as prompts, questions, hints, stories, conceptual models, visualizations are performed to facilitate a learner's progress during learning tasks, e.g., [43].

2.3 Learner-controlled learning

For the integration of observational and instructional learning, control is a crucial element. It is discussed, for example, by Gibbons and Gray [44] that instructions serve learning processes best when the learner has control over them. The scaffolded model-based learning format mentioned above supports this. Kozma [28] suggested that individuals actively use external information sources for mental model formation. Learners are sensitive to characteristics of the learning environment such as the availability of certain information at a given time, the structure of the information and the ease with which it can be accessed. Thus, the learner's need for instruction and the ease acquiring it are crucial for building effective mental models. In learning methods based on guided discovery, the learner seeks for information to complete the initial mental model. This requires the learner to be proactive and in control of the learning process. In contrast, in expository teaching methods, an instructor aims at directing the mental model formation by providing adequate information according to some temporal sequence [33]. Meela, and Yuenyong [29] demonstrated in their study that Model-Based Inquiry (MBI) can support a student's mental model formation in scientific learning. MBI focuses on students' formulations of questions and procedures [31]. Feedback on performance are a significant factor in learning [1, 45]; many studies support that feedback is crucial in skill acquisition [46].

Thus, in the adaptive network model introduced in the current paper, the learner can seek for instructions whenever it is useful or needed or as a feedback about what she/he has learnt by observation. The control for this was modeled by control states for instructions on a separate level within the adaptive network model. Using this, the learner controls timing and content of incoming information by seeking it only when it seems appropriate to her/him. More specifically, in Section 3 it is shown how a learning process based on mental models can be modeled by a generic three-level adaptive network architecture. In this architecture, the mental models themselves can be modeled at the base level as networks. In addition, during learning the mental models change; this can be modeled by (first-order) network adaptation at a second level. Control of the learning process is a form of adaptation of the learning process; this can be modeled at the third level addressing adaptation of the first-order adaptive network for the learning process: *second-order network adaptation*.

3 Network architecture for controlled mental model learning

In this section a global view on the architecture of the introduced network model for learner-controlled mental model learning is discussed. Following with what was concluded in Section 2, this architecture must cover the following three types of processes in an integrated manner:

1. The mental models themselves described by base networks
2. Learning as change of mental models described by first-order network adaptation
3. Control of learning processes described by second-order network adaptation.

Using the notion of self-modeling network (also called reified network) [22–24], these three description levels indeed can be modeled adequately by a three-level second-order adaptive network architecture as depicted in Fig 1.

Here, for any specific application each plane contains a specific network and the specific upward and downward connections define the interactions between the different levels.

More specifically, adding a self-model to a network model is done in the way that for some of the network structure characteristics additional network states (self-model states) are added. In the network-oriented modeling approach [23] applied here, in particular for nodes (also called states) X and Y, the following network structure characteristics are used:

- $\boldsymbol{\omega}_{X,Y}$ for *connectivity* (connections $X \rightarrow Y$ with their connection weights)
- $\boldsymbol{\gamma}_{i,Y}$ and $\boldsymbol{\pi}_{i,j,Y}$ for *aggregation* (combination function choices and their parameters for each node Y)
- $\boldsymbol{\eta}_Y$ for *timing* (speed factors for each node Y)

Then to obtain adaptive networks, self-model nodes can be added to the network for any of these characteristics to make it adaptive:

- **Connectivity self-model**
  Self-model nodes $\mathbf{W}_{X,Y}$ are added representing connection weights $\boldsymbol{\omega}_{X,Y}$
- **Aggregation self-model**
  Self-model nodes $\mathbf{C}_{j,Y}$ are added representing combination function weights $\boldsymbol{\gamma}_{i,Y}$ and/or self-model states $\mathbf{P}_{i,j,Y}$ representing combination function parameters $\boldsymbol{\pi}_{i,j,Y}$
- **Timing self-model**
  Self-model nodes $\mathbf{H}_Y$ are added representing speed factors $\boldsymbol{\eta}_Y$

The notations $\mathbf{W}_{X,Y}$, $\mathbf{C}_{i,Y}$, $\mathbf{P}_{i,j,Y}$, $\mathbf{H}_Y$ for the self-model states indicate the referencing relation with respect to the characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{i,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$: here **W** refers to $\boldsymbol{\omega}$, **C** refers to $\boldsymbol{\gamma}$, **P** refers to $\boldsymbol{\pi}$, and **H** refers to $\boldsymbol{\eta}$, respectively. These **W**, **C**, **P** and **H** notations are considered to indicate

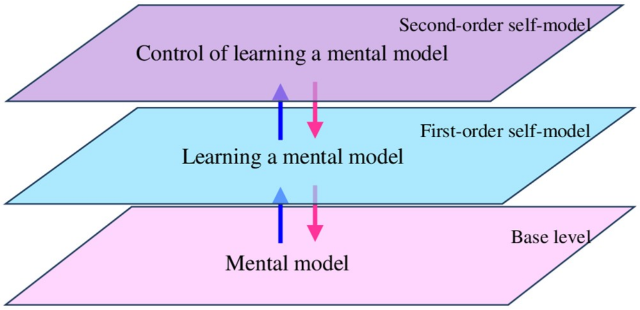

Fig 1. Overview of the introduced second-order adaptive network architecture.

https://doi.org/10.1371/journal.pone.0255503.g001

the roles these **W**-, **C**-, **P**- and **H**-states play in the network, so that at the base level the values of them are used for the intended characteristics. Sometimes slightly different notations are used, for example, by adding the letter **R** for representation to emphasize that it represents some characteristic of the network: $\mathbf{RW}_{X,Y}$, $\mathbf{RC}_{i,Y}$, $\mathbf{RP}_{i,j,Y}$, $\mathbf{RH}_{Y}$. This construction can easily be applied iteratively to obtain multiple levels of self-models. For example, by adding a second-order self-model state $\mathbf{H}_{\mathbf{W}X,Y}$ for $\mathbf{W}_{X,Y}$, the adaptation speed $\boldsymbol{\eta}_{\mathbf{W}X,Y}$ of $\mathbf{W}_{X,Y}$ can be made adaptive. Therefore second-order adaptation plays an important role here to control adaptive processes which can be easily modelled as well.

The more specific adaptive network model described in Section 4 will be a refinement of the overall network architecture depicted in Fig 1. Tables 1 and 2 summaries the generic types of states and connections used at and between the three levels within this architecture. Note that the colours used in these tables indicate to which level the states belong, as they correspond to the colours of the planes in the 3D figures such as Fig 1.

At the base level, the learner's (subjective) mental model is defined by connections between base states $\mathrm{BS}_X \rightarrow \mathrm{BS}_Y$; in addition, the connections between observation states $\mathrm{OS}_X \rightarrow \mathrm{OS}_Y$ define the (objective) relations in the real world. Note that, following the quote of Craik [6], p. 51 in Section 2, the causal relations $\mathrm{BS}_X \rightarrow \mathrm{BS}_Y$ defining the mental model, are in a one-to-one correspondence with the causal relations $\mathrm{OS}_X \rightarrow \mathrm{OS}_Y$ between the (observed) world states. Therefore, as can be seen in Figs 2 and 3, within the base plane the subnetwork for the BS-states has a connectivity structure that is isomorphic to the connectivity structure of the subnetwork for the OS-states.

Moreover, the connections from observation state to base state $\mathrm{OS}_Y \rightarrow \mathrm{BS}_Y$ define the mirroring process by which the observations affect the learner's own states.

Table 1. Types of states in the introduced three level network architecture.

BS_Y	Base states for the considered mental model of the learner
OS_Y	The corresponding observation states in the real world
$\mathbf{IS}_{X,Y}$	Representation for the connection weights of the mental model of the instructor
$\mathbf{LW}_{X,Y}$	Representation for the connection weights for the mental model as learnt from observation (using the Hebbian learning principle)
$\mathbf{IW}_{X,Y}$	Representation for the connection weights for the mental model as learnt from instruction (using the instructor)
$\mathbf{RW}_{X,Y}$	Representation for the connection weights for the learner's mental model integrating observational (via $\mathbf{LW}_{X,Y}$) and instructional (via $\mathbf{IW}_{X,Y}$) learning
$\mathbf{CIW}_{X,Y}$	Initiation of instruction: control state for requesting the weight of the connection from X to Y for the mental model from the instructor

https://doi.org/10.1371/journal.pone.0255503.t001

Table 2. Types of connections in the introduced adaptive network architecture.

Intralevel connections		
$\mathrm{BS}_X \rightarrow \mathrm{BS}_Y$	The learner's (subjective) connections between the base states, indicating the current mental model of the learner	
$\mathrm{OS}_X \rightarrow \mathrm{OS}_Y$	The real world's (objective) connections between the observation states, indicating the real-world process	
$\mathrm{OS}_Y \rightarrow \mathrm{BS}_Y$	Mirroring connections defining the mirroring process for the base states. These connections model the effect of observations on the learner.	
$\mathbf{IS}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$	Being informed by the instructor: the communicated instruction concerning the connection from X to Y. These connections $\mathbf{IS}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$ can be controlled by control states $\mathbf{CIW}_{X,Y}$ at the second reification level	
$\mathbf{IW}_{X,Y} \rightarrow \mathbf{RW}_{X,Y}$	Integration of knowledge obtained by instructional learning	
$\mathbf{LW}_{X,Y} \rightarrow \mathbf{RW}_{X,Y}$	Integration of knowledge obtained by observational learning	
Interlevel connections		
$\mathrm{BS}_X \rightarrow \mathbf{LW}_{X,Y}$ $\mathrm{BS}_Y \rightarrow \mathbf{LW}_{X,Y}$	Connections supporting observational learning	Upward from base level to first reification level
$\mathbf{RW}_{X,Y} \rightarrow \mathrm{BS}_Y$	Effectuation of base connection weights in the mental model	Downward from first reification level to base level
$\mathbf{LW}_{X,Y} \rightarrow \mathbf{CIW}_{X,Y}$	Observational learning monitoring connections	Upward from first to second reification level
$\mathbf{CIW}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$	Effectuation of instructional learning control	Downward from second to first reification level

https://doi.org/10.1371/journal.pone.0255503.t002

At the first-order self-model level, the self-model of the mental model from the base level is modeled by states $\mathbf{RW}_{X,Y}$ that explicitly represent the connection weights of the mental model as used in the processing of this mental model at the base level. At first sight, this may seem a double (and therefore redundant) representation of the same mental model, but to handle the process of learning of this mental model well, this explicit 'additional' representation in the form of the mental model's self-model is crucial, because when adaptivity is addressed, the network characteristic (in this case connection weight) is no longer one static parameter value but becomes a variable with values that change over time (as happens for every dynamical system model for which some of its parameters are made adaptive). The way of conceptualization applied here is used more often within neuroscience as a distinction between (1) activation propagation through a network of neurons, (2) plasticity of this network, and (3) metaplasticity as control over this plasticity; e.g., [25, 47–50]. Inspired by this, in the current paper, (1) and (2) of this form of conceptualization are used to model use and adaptation of a mental

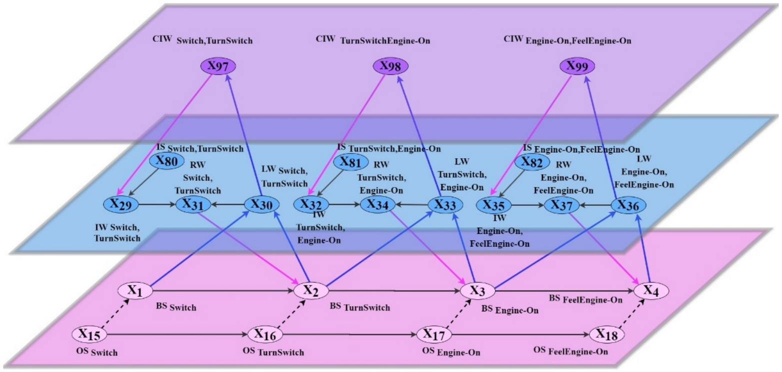

Fig 2. Connectivity for part of the second-order adaptive network model.

https://doi.org/10.1371/journal.pone.0255503.g002

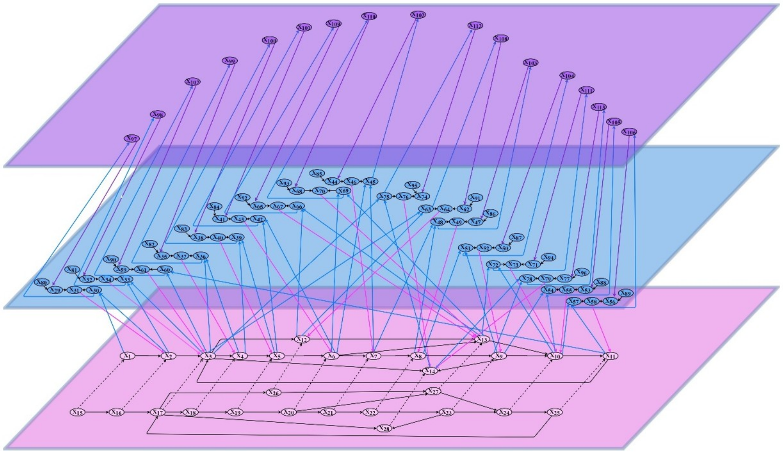

Fig 3. Connectivity for the complete adaptive network model.

https://doi.org/10.1371/journal.pone.0255503.g003

model, respectively, and (3) for the control over this adaptation, as will be explained in more detail below.

In addition to the states $\mathbf{RW}_{X,Y}$, also self-model states $\mathbf{LW}_{X,Y}$ and $\mathbf{IW}_{X,Y}$ are used as part of the mental model's self-model. Here, $\mathbf{LW}_{X,Y}$ represents what has been learnt about the connection from X to Y within the mental model by observational learning and $\mathbf{IW}_{X,Y}$ represents what has been acquired from instructional learning.

The intra-level connections $\mathbf{LW}_{X,Y} \rightarrow \mathbf{RW}_{X,Y}$ and $\mathbf{IW}_{X,Y} \rightarrow \mathbf{RW}_{X,Y}$ model the integration within $\mathbf{RW}_{X,Y}$ of what is learnt by observational learning and what is learnt by instructional learning. The connections $\mathbf{IS}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$ model the instruction itself: the communication actions from instructor to learner.

These communication actions from the instructor to the learner depend on control. To this end, in the second-order self-model, states $\mathbf{CIW}_{X,Y}$ are included. Such a state indicates that the learner wants to hear the instructor's knowledge about the connection from X to Y. It is assumed that the instructor will respond accordingly. This happens by giving $\mathbf{CIW}_{X,Y}$ the role of connection weight representation $\mathbf{W}_{\mathbf{IS}X,Y,\mathbf{IW}X,Y}$ for the intended connection $\mathbf{IS}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$ from the instructor to the learner. This works by the processing of the first-order self-model for the connection weight $\boldsymbol{\omega}_{\mathbf{IS}X,Y,\mathbf{IW}X,Y}$ the value of $\mathbf{CIW}_{X,Y}$ is used. Therefore, as long as the value of $\mathbf{CIW}_{X,Y}$ is 0, no communication takes place, while as soon as this value of $\mathbf{CIW}_{X,Y}$ is 1, this communication does take place. This represents the way in which that communication becomes controlled. The effect of activation of $\mathbf{CIW}_{X,Y}$ can be interpreted in the sense that the communication channel from the instructor state $\mathbf{IS}_{X,Y}$ to the learner state $\mathbf{IW}_{X,Y}$ is opened, so that this information is transferred from the instructor state $\mathbf{IS}_{X,Y}$ to the learner state $\mathbf{IW}_{X,Y}$. This opening of the channel $\mathbf{IS}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$ is modeled by the downward connections $\mathbf{CIW}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$, where $\mathbf{CIW}_{X,Y}$ represents the role of connection weight from $\mathbf{IS}_{X,Y}$ to $\mathbf{IW}_{X,Y}$.

The only remaining piece then is to determine when exactly $\mathbf{CIW}_{X,Y}$ should become active. This is done via its incoming observational learning monitoring connection $\mathbf{LW}_{X,Y} \rightarrow \mathbf{CIW}_{X,Y}$ which makes that the control state $\mathbf{CIW}_{X,Y}$ will become active depending on the corresponding $\mathbf{LW}$-state $\mathbf{LW}_{X,Y}$. This models that the part where the learner asks the instructor for verification and confirmation of what was just learnt by observation (and the learner does not ask anything about what not yet has been observed). A more detailed explanation of the network's connectivity for a specific case study can be found in Section 4.

4 Detailed description of the second-order adaptive network model for a case study

In this section, a more detailed description can be found of the designed second-order adaptive network model for a realistic case study, that was created for illustrative purposes. It is described by the following scenario:

> Person A has almost no knowledge about a car's components and their interplay and how to drive a car. This person's mental model of the car and driving it has to be learned during driving lessons. During person A's first driving lesson, instructor B demonstrates how to start a car and get it moving. The observation of B makes that A learns an initial mental model of the car and how it can be operated it (observational learning). During A's further learning, an iterative process of extending and/or modifying the mental model takes place, leading to a more accurate and complete mental model. Besides observational learning, also learning from instruction plays an important role (instructional learning). This instructional learning takes place by incorporating incoming information communicated by B. In this scenario this instructional learning only takes place upon request of the learner (learner-controlled instructional learning), as a form of verification and consolidation after A learnt about it by observational learning.

The network-oriented modeling approach for adaptive networks [22–24] used here has been briefly introduced in Section 3. Some more details will follow here. Recall that for adaptive networks the notion of *self-modeling network* is used. For example, for adaptive connectivity characteristics, states $\mathbf{RW}_{X,Y}$ are added representing adaptive connection weights $\boldsymbol{\omega}_{X,Y}$. They form a *self-model* of the network's own structure in the form of a subnetwork within the network. To graphically distinguish them from states at the level of X and Y, these self-model states are depicted at one level higher (e.g., see the blue planes in Figs 1–3 with representations of weights of adaptive connections from the base planes).

As in this case the learning is controlled, it is adaptive itself, which is depicted by the third level (purple plane) for *second-order adaptation* in Figs 1–3, which include *second-order reification states* $\mathbf{CIW}_{X,Y}$ that represent the weight of the connection $\mathbf{IS}_{X,Y} \rightarrow \mathbf{IW}_{X,Y}$ of the middle level (see Section 3). The structure formed by the lowest two (interacting) levels distinguish the two types of processes (and their interaction): *using* the mental model by changing the **BS**-states represented at the base level (used for *internal simulation* of the mental model) versus *adjusting* the mental model by changing the representations at the self-model level. The different types of states for the detailed model are explained in Tables 4, 5, 6. Fig 2 depicts the connectivity for only a part for a small number of the states for better understanding. Fig 3 shows the connectivity for the complete network model. The second-order self-model level (the purple plane) enables to control the learning process by changing some of the intra-level connections within the first-order self-model (which in turn affects the dynamics of these first-order self-model states), based on the second-level reification **CIW**-states (control states); this is used to model *learner-controlled instruction*, as discussed in Section 3.

The conceptual representation of a network model as mentioned above can easily be transformed in an automated manner into a numerical representation using a dedicated modeling environment; within the software, this results in difference equations ([22, 23], Chapter 9):

$$\begin{aligned} & Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)]\,\Delta t \\ \text{or}\quad & \mathrm{d}Y(t)/\mathrm{d}t = \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)] \qquad (1) \\ \text{where}\quad & \mathbf{aggimpact}_Y(t) = \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) \end{aligned}$$

Here the overall combination function $\mathbf{c}_Y(..)$ for state Y is the weighted average of available basic combination functions $\mathbf{c}_j(..)$ (in the Combination Function Library) by specified weights $\boldsymbol{\gamma}_{j,Y}$ and parameters $\boldsymbol{\pi}_{1,j,Y}$, $\boldsymbol{\pi}_{2,j,Y}$ of $\mathbf{c}_j(..)$ for Y:

$$\mathbf{c}_Y(V_1, \ldots, V_k) = \frac{\gamma_{1,Y}\mathrm{c}_1(V_1,\ \ldots,\ V_k) + \ldots + \gamma_{m,Y}\mathrm{c}_m(V_1,\ \ldots,\ V_k)}{\gamma_{1,Y} + \ldots + \gamma_{m,Y}} \tag{2}$$

In case of self-models, the self-model states define the dynamics of state Y in a canonical manner according to (1), whereby the adaptive characteristics among $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{i,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$ are replaced by the state values of self-model states $\mathbf{W}_{X,Y}$, $\mathbf{C}_{i,Y}$, $\mathbf{P}_{i,j,Y}$, $\mathbf{H}_Y$ at time t, respectively (for more details, see [22, 23]).

In the model presented here, for the states the following combination functions were used, all generating values in [0, 1] (assuming that their arguments are in [0, 1]). The *Euclidean combination function* $\mathbf{eucl}_{\mathbf{n},\boldsymbol{\lambda}}(V_1, \ldots, V_k)$ where $\boldsymbol{n}$ is the order (any positive number), and $\boldsymbol{\lambda}$ the scaling factor is defined by:

$$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \sqrt[n]{\frac{V_1^{\boldsymbol{n}} + \cdots + V_k^{\boldsymbol{n}}}{\boldsymbol{\lambda}}} \tag{3}$$

where $V_1, \ldots, V_k \in [0, 1]$ indicate the impacts $\boldsymbol{\omega}_{Xi,Y}\, X_i\ (t)$ from the states $X_1, \ldots, X_k$ from which Y has an incoming connection. In addition, the *advanced logistic sum combination function* $\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(\ldots)$ is used:

$$\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1, \ldots, V_k) = \left[\frac{1}{1 + \mathrm{e}^{-\boldsymbol{\sigma}}(V_1 + \cdots + V_k - \boldsymbol{\tau})} - \frac{1}{1 + \mathrm{e}^{\boldsymbol{\sigma}\boldsymbol{\tau}})}\right](1 + \mathrm{e}^{-\boldsymbol{\sigma}\tau}) \tag{4}$$

with steepness $\boldsymbol{\sigma}$ and threshold $\boldsymbol{\tau}$ (with similar $V_1, \ldots, V_k$ as above)

Table 3 provides an overview of the base states used to model the mental model (the BS-states) and the base states for the observations (the OS-states). Table 4 summarises the first-order self-model states for the learner's learning (the **RW**-, **LW**- and **IW**-states) and Table 5 addresses the instructor's Information States (the **IS**-states).

The *Hebbian learning combination function* $\mathbf{hebb}_{\boldsymbol{\mu}}(..)$ for learning of the connection from state X to state Y, and used in particular for the **LW**-states is defined by

$$\mathbf{hebb}_{\mu}(V_1, V_2, W) = V_1 V_2(1 - W) + \boldsymbol{\mu} W \tag{5}$$

where $\boldsymbol{\mu}$ is the persistence parameter, V_1 stands for state value $X(t)$, V_2 for $Y(t)$, and W for the learnt connection weight reification state value $\mathbf{LW}_{X,Y}(t)$, which all are in the [0, 1] interval. Hebbian learning is a well-known adaptation principle addressing adaptive connectivity, which can be explained by:

> 'When an axon of cell A is near enough to excite B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.'
>
> [25], p. 62
>
> This is sometimes simplified (neglecting the phrase 'one of the cells firing B') to:
>
> 'What fires together, wires together'
>
> [51, 52]

Table 3. Explanation of the base level states for the mental model within the network model.

Base States		Explanation
X_1	$\text{BS}_{\text{Switch}}$	Learner’s representation state for Switch
X_2	$\text{BS}_{\text{TurnSwitch}}$	Learner’s representation state for TurnSwitch
X_3	$\text{BS}_{\text{Engine-0n}}$	Learner’s representation state for Engine-On
X_4	$\text{BS}_{\text{FeelEngine-On}}$	Learner’s representation state for FeelEngine-On
X_5	$\text{BS}_{\text{PresClutch}}$	Learner’s representation state for PressClutch
X_6	$\text{BS}_{\text{Clutch-On}}$	Learner’s representation state for Clutch-On
X_7	$\text{BS}_{\text{Gearbox-Neutral}}$	Learner’s representation state for Gearbox-Neutral
X_8	$\text{BS}_{\text{PressGear 1}}$	Learner’s representation state for PressGear1
X_9	$\text{BS}_{\text{Gear1-On}}$	Learner’s representation state for Gear1-On
X_{10}	$\text{BS}_{\text{PressAccelerator}}$	Learner’s representation state for PressAccelerator
X_{11}	$\text{BS}_{\text{Accelerator-On}}$	Learner’s representation state for Accelerator-On
X_{12}	$\text{BS}_{\text{RevMeter-On}}$	Learner’s representation state for Rev-Meter-On
X_{13}	$\text{BSe}_{\text{BiteState}}$	Learner’s representation state for BiteState
X_{14}	$\text{BS}_{\text{MovingState}}$	Learner’s representation state for MovingState
X_{15}	$\text{OS}_{\text{Switch}}$	Observation State for Switch
X_{16}	$\text{OS}_{\text{TurnSwitch}}$	Observation State for TurnSwitch
X_{17}	$\text{OS}_{\text{Engine-0n}}$	Observation State for Engine-On
X_{18}	$\text{OS}_{\text{FeelEngine-On}}$	Observation State for FeelEngine-On
X_{19}	$\text{OS}_{\text{PressClutch}}$	Observation State for PressClutch
X_{20}	$\text{OS}_{\text{Clutch-On}}$	Observation State for Clutch-On
X_{21}	$\text{OS}_{\text{Gearbox-Neutral}}$	Observation State for Gearbox-Neutral
X_{22}	$\text{OS}_{\text{PressGear 1}}$	Observation State for PressGear1
X_{23}	OS_{Gear1}	Observation State for Gear1-On
X_{24}	$\text{OS}_{\text{PressAccelerator}}$	Observation State for PressAccelerator
X_{25}	$\text{OS}_{\text{Accelerator-On}}$	Observation State for Accelerator-On
X_{26}	$\text{OS}_{\text{RevMeter-On}}$	Observation State for Rev-Meter-On
X_{27}	$\text{OS}_{\text{BiteState}}$	Observation State for BiteState
X_{28}	$\text{OS}_{\text{MovingState}}$	Observation State for MovingState

https://doi.org/10.1371/journal.pone.0255503.t003

In formula (5), the condition ‘what fires together’ is modeled by the part with the product V_1V_2, as that is high if both states X and Y have a high value and low otherwise. The factor (1-W) provides a kind of normalisation that makes that the value for $W = \mathbf{LW}_{X,Y}(t)$ does not exceed 1. The term $\boldsymbol{\mu}W$ in (5) models persistency, where $\boldsymbol{\mu}$ indicates the fraction of the previously learnt value W that persists per time unit; for example if $\boldsymbol{\mu} = 0.9$, then every time unit 10% is lost (also called extinction).

In Table 6 an overview can be found of the second-order self-model states. They all are **CIW**-states for the control of the instructional learning of certain connection weights. For these **CIW**-states the logistic sum combination function $\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1, \ldots, V_k)$ is used.

5 Simulation results for an example scenario

The second-order adaptive network model was simulated using the dedicated software environment implemented in Matlab as described in [23], Ch 9, to study the learning of a mental model for a car’s functioning and driving it; see Fig 4 and further. For the simulation $\Delta t = 0.5$ was chosen, the total time is 800 (so 1600 simulation steps); the time scale is left abstract here. In the S1 Appendix section the full specification of the network characteristics can be found. The speed factors for the BS-states were set at 0.4, for OS-states at 0.05, for **IW**-states at 0.1,

Table 4. Explanation of the first-order self-model states for the mental model learning in the network model.

First-order self-model states for the learning		Explanation
X_{29}	$\mathbf{IW}_{\text{Switch,TurnSwitch}}$	Representation state for Informed Connection Weight Switch→TurnSwitch
X_{30}	$\mathbf{LW}_{\text{Switch,TurnSwitch}}$	Representation state for Learnt Connection Weight Switch→TurnSwitch
X_{31}	$\mathbf{RW}_{\text{Switch,TurnSwitch}}$	Representation state for overall Connection Weight Switch→TurnSwitch
X_{32}	$\mathbf{IW}_{\text{TurnSwitch,Engine-On}}$	Representation state for Informed Connection Weight TurnSwitch→Engine-On
X_{33}	$\mathbf{LW}_{\text{TurnSwitch,Engine-On}}$	Representation state for Learnt Connection Weight TurnSwitch→Engine-On
X_{34}	$\mathbf{RW}_{\text{TurnSwitch,Engine-On}}$	Representation state for overall Connection Weight TurnSwitch→Engine-On
X_{35}	$\mathbf{IW}_{\text{Engine-On,FeelEngine-On}}$	Representation state for Informed Connection Weight Engine-On→FeelEngine-On
X_{36}	$\mathbf{LW}_{\text{Engine-On,FeelEngine-On}}$	Representation state for Learnt Connection Weight Engine-On→FeelEngine-On
X_{37}	$\mathbf{RW}_{\text{Engine-On,FeelEngine-On}}$	Representation state for overall Connection Weight Engine-On→FeelEngine-On
X_{38}	$\mathbf{IW}_{\text{FeelEngine-On,PressClutch}}$	Representation state for Informed Connection Weight FeelEngine-On→PressClutch
X_{39}	$\mathbf{LW}_{\text{eelEngine-On,PressClutch}}$	Representation state for Learnt Connection Weight FeelEngine-On→PressClutch
X_{40}	$\mathbf{RW}_{\text{FeelEngine-On,PressClutch}}$	Representation state for overall Connection Weight FeelEngine-On→PressClutch
X_{41}	$\mathbf{IW}_{\text{PressClutch,Clutch-On}}$	Representation state for Informed Connection Weight PressClutch→Clutch-On
X_{42}	$\mathbf{LW}_{\text{PressClutch,Clutch-On}}$	Representation state for Learnt Connection Weight PressClutch→Clutch-On
X_{43}	$\mathbf{RW}_{\text{PressClutch,Clutch-On}}$	Representation state for overall Connection Weight PressClutch→Clutch-On
X_{44}	$\mathbf{IW}_{\text{Clutch-On,Gearbox-Neutral}}$	Representation state for Informed Connection Weight Clutch-On→Gearbox-Neutral
X_{45}	$\mathbf{LW}_{\text{Clutch-On,Gearbox-Neutral}}$	Representation state for Learnt Connection Weight Clutch-On→Gearbox-Neutral
X_{46}	$\mathbf{RW}_{\text{Clutch-On,Gearbox-Neutral}}$	Representation state for overall Connection Weight Clutch-On→Gearbox-Neutral
X_{47}	$\mathbf{IW}_{\text{Gearbox-Neutral,PressGear1}}$	Representation state for Informed Connection Weight Gearbox-Neutral→PressGear1
X_{48}	$\mathbf{LW}_{\text{Gearbox-Neutral,PressGear1}}$	Representation state for Learnt Connection Weight Gearbox-Neutral→PressGear1
X_{49}	$\mathbf{RW}_{\text{Gearbox-Neutral,PressGear1}}$	Representation state for overall Connection Weight Gearbox-Neutral→PressGear1
X_{50}	$\mathbf{IW}_{\text{PressGear1,Gear1-On}}$	Representation state for Informed Connection Weight PressGear1→Gear1-On
X_{51}	$\mathbf{LW}_{\text{PressGear1,Gear1-On}}$	Representation state for Learnt Connection Weight PressGear1→Gear1-On
X_{52}	$\mathbf{RW}_{\text{PressGear1,Gear1-On}}$	Representation state for overall Connection Weight PressGear1→Gear1-On
X_{53}	$\mathbf{IW}_{\text{Gear1-On,PressAccelerator}}$	Representation state for Informed Connection Weight Gear1-On→PressAccelerator
X_{54}	$\mathbf{LW}_{\text{Gear1-On,PressAccelerator}}$	Representation state for Learnt Connection Weight Gear1-On→PressAccelerator
X_{55}	$\mathbf{RW}_{\text{Gear1-On,PressAccelerator}}$	Representation state for overall Connection Weight Gear1→On,PressAccelerator
X_{56}	$\mathbf{IW}_{\text{PressAccelerator,Accelerator-On}}$	Representation state for Informed Connection Weight PressAccelerator→Accelerator-On
X_{57}	$\mathbf{LW}_{\text{PressAccelerator,Accelerator-On}}$	Representation state for Learnt Connection Weight PressAccelerator→Accelerator-On
X_{58}	$\mathbf{RW}_{\text{PressAccelerator,Accelerator-On}}$	Representation state for Connection Weight PressAccelerator→Accelerator-On
X_{59}	$\mathbf{IW}_{\text{Accelerator-On,Engine-On}}$	Representation state for Informed Connection Weight Accelerator-On→Engine-On
X_{60}	$\mathbf{LW}_{\text{Accelerator-On,Engine-On}}$	Representation state for Learnt Connection Weight Accelerator-On→Engine-On
X_{61}	$\mathbf{RW}_{\text{Accelerator-On,Engine-On}}$	Representation state for overall Connection Weight Accelerator-On→Engine-On
X_{62}	$\mathbf{IW}_{\text{Engine-On,Rev-Meter-On}}$	Representation state for Informed Connection Weight Engine-On→Rev-Meter-On
X_{63}	$\mathbf{LW}_{\text{Engine-On,Rev-Meter-On}}$	Representation state for Learnt Connection Weight Engine-On→Rev-Meter-On
X_{64}	$\mathbf{RW}_{\text{Engine-On,Rev-Meter-On}}$	Representation state for overall Connection Weight Engine-On→Rev-Meter-On
X_{65}	$\mathbf{IW}_{\text{RevMeter-On,BiteState}}$	Representation state for Informed Connection Weight RevMeter-On→BiteState
X_{66}	$\mathbf{LW}_{\text{RevMeter-On,BiteState}}$	Representation state for Learnt Connection Weight RevMeter-On→BiteState
X_{67}	$\mathbf{RW}_{\text{RevMeter-On,BiteState}}$	Representation state for overall Connection Weight RevMeter-On→BiteState
X_{68}	$\mathbf{IW}_{\text{Clutch-On,BiteState}}$	Representation state for Informed Connection Weight Clutch-On→BiteState
X_{69}	$\mathbf{LW}_{\text{Clutch-On,BiteState}}$	Representation state for Learnt Connection Weight Clutch-On→BiteState
X_{70}	$\mathbf{RW}_{\text{Clutch-On,BiteState}}$	Representation state for overall Connection Weight Clutch-On→BiteState
X_{71}	$\mathbf{IW}_{\text{BiteState,PressAccelerator}}$	Representation state for Informed Connection Weight BiteState→PressAccelerator
X_{72}	$\mathbf{LW}_{\text{BiteState,PressAccelerator}}$	Representation state for Learnt Connection Weight BiteState→PressAccelerator
X_{73}	$\mathbf{RW}_{\text{BiteState,PressAccelerator}}$	Representation state for overall Connection Weight BiteState→PressAccelerator
X_{74}	$\mathbf{IW}_{\text{Engine-On,MovingState}}$	Representation state for Informed Connection Weight Engine-On→MovingState
X_{75}	$\mathbf{LW}_{\text{Engine-On,MovingState}}$	Representation state for Learnt Connection Weight Engine-On→MovingState

(*Continued*)

Table 4. (Continued)

First-order self-model states for the learning		Explanation
X_{76}	$\mathbf{RW}_{\text{Engine-On,MovingState}}$	Representation state for overall Connection Weight Engine-On→MovingState
X_{77}	$\mathbf{IW}_{\text{Gear1-On,MovingState}}$	Representation state for Informed Connection Weight Gear1-On→MovingState
X_{78}	$\mathbf{LW}_{\text{Gear1-On,MovingState}}$	Representation state for Learnt Connection Weight Gear1-On→MovingState
X_{79}	$\mathbf{RW}_{\text{Gear1-On,MovingState}}$	Representation state for overall Connection Weight Gear1-On→MovingState

https://doi.org/10.1371/journal.pone.0255503.t004

and for **LW**-states and **RW**-states at 0.4. For the second-order **CIW**-states, their speed factors were set at 0.4. All BS- and OS-states use either the Euclidean function (3) or the logistic sum function (4), all **LW**-states the Hebbian learning function (5), the **IW**-states the logistic sum function (4), the **RW**-states the first-order Euclidean function (3), and the **CIW**-states the logistic sum function (4). All BS-states have initial value 0. All OS-States have initial value 0, except the first OS-State X_{15} which has an initial value of 1. For all the **IW**-, **LW**-, and **RW**-states, the initial value was set at 0.1.

The **IS**-states have constant value 1, as they refer to the knowledge of the instructor (see also Fig 5). Note that it has been specified in such a way that only one of the **IW**-state or **LW**-state is not enough to get a related **RW**-state with a high value close to 1. A typical pattern is that first, based on a learnt **LW**-state only, the **RW**-state gets a value somewhere in the middle of the 0–1 interval, and only after instructional learning making the **IW**-state high, the **RW**-state value increases to a high value close to 1. This makes that first based on the value of **LW**-state (i.e., by observational learning), the second-order **CIW**-state is activated (Fig 6) which in turn makes the **IW**-state getting a value close to 1 (Fig 7). Only after the learner seeks instructional information making the **IW**-state high, the **RW**-state value increases to 1. This shows that the learner actively engages seeking more information to confirm the accuracy of what he/she has learnt by observation.

Table 5. Explanation of the first-order self-model states for the instructor's Information States in the network model.

First-order self-model states for the instructor's information		Explanation
X_{80}	$\text{IS}_{\text{Switch,TurnSwitch}}$	Representation state for Information State for Switch→TurnSwitch
X_{81}	$\text{IS}_{\text{TurnSwitch,Engine-On}}$	Representation state for instructor's Information State for TurnSwitch→Engine-On
X_{82}	$\text{IS}_{\text{Engine-On,FeelEngine-On}}$	Representation state for instructor's Information State for Engine-On→FeelEngine-On
X_{83}	$\text{IS}_{\text{FeelEngineOn,PressClutch}}$	Representation state for instructor's Information State for FeelEngineOn→PressClutch
X_{84}	$\text{IS}_{\text{PressClutch,Clutch-On}}$	Representation state for instructor's Information State for PressClutch→Clutch-On
X_{85}	$\text{IS}_{\text{Clutch-On,Gear-BoxNeutral}}$	Representation state for instructor's Information State for Clutch-On→Gear-BoxNeutral
X_{86}	$\text{IS}_{\text{GearBox-Neutral,PressGear1}}$	Representation state for instructor's Information State for GearBox-Neutral,PressGear1
X_{87}	$\text{IS}_{\text{PressGear1,Gear1-On}}$	Representation state for instructor's Information State for PressGear1→Gear1-On
X_{88}	$\text{IS}_{\text{Gear1-On, PressAccelerator}}$	Representation state for instructor's Information State for Gear1-On→PressAccelerator
X_{89}	$\text{IS}_{\text{PressAccelerator, Accelerator-On}}$	Representation state for instructor's Information State for PressAccelerator→Accelerator-On
X_{90}	$\text{IS}_{\text{Accelerator-On, Engine-On}}$	Representation state for instructor's Information State for Accelerator-On→Engine-On
X_{91}	$\text{IS}_{\text{Engine-On,Rev-Meter-On}}$	Representation state for instructor's Information State for Engine-On→Rev-Meter-On
X_{92}	$\text{IS}_{\text{RevMeter-On,BiteState}}$	Representation state for instructor's Information State for RevMeter-On→BiteState
X_{93}	$\text{IS}_{\text{Clutch-On,BiteState}}$	Representation state for instructor's Information State for Clutch-On→BiteState
X_{94}	$\text{IS}_{\text{BiteState,PressAccelerator}}$	Representation state for instructor's Information State for BiteState→PressAccelerator
X_{95}	$\text{IS}_{\text{Engine-On,MovingState}}$	Representation state for instructor's Information State for Engine-On→MovingState
X_{96}	$\text{IS}_{\text{Gear-On1,MovingState}}$	Representation state for instructor's Information State for Gear-On1→MovingState

https://doi.org/10.1371/journal.pone.0255503.t005

Table 6. Explanation of the second-order self-model states for control of the mental model learning in the network model.

Second-order self-model states for the control of the learning		Explanation
X_{97}	$\mathbf{CIW}_{\text{Switch,TurnSwitch}}$	Representation state for Control of Information Weight for Switch→TurnSwitch
X_{98}	$\mathbf{CIW}_{\text{TurnSwitch,Engine-On}}$	Representation state for Control of Information Weight for TurnSwitch→EngineOn
X_{99}	$\mathbf{CIW}_{\text{Engine-On,FeelEngine-On}}$	Representation state for Control of Information Weight for Engine-On→FeelEngineOn
X_{100}	$\mathbf{CIW}_{\text{FeelEngineOn,PressClutch}}$	Representation state for Control of Information Weight for FeelEngineOn→PressClutch
X_{101}	$\mathbf{CIW}_{\text{PressClutch,Clutch-On}}$	Representation state for Control of Information Weight for PressClutch→Clutch-On
X_{102}	$\mathbf{CIW}_{\text{Clutch-On,Gear-BoxNeutral}}$	Representation state for Control of Information Weight for Clutch-On→Gear-BoxNeutral
X_{103}	$\mathbf{CIW}_{\text{GearBox-Neutral,PressGear1}}$	Representation state for Control of Information Weight for GearBox-Neutral,PressGear1
X_{104}	$\mathbf{CIW}_{\text{PressGear1,Gear1-On}}$	Representation state for Control of Information Weight for PressGear1→Gear1-On
X_{105}	$\mathbf{CIW}_{\text{Gear1-On, PressAccelerator}}$	Representation state for Control of Information Weight for Gear1-On→PressAccelerator
X_{106}	$\mathbf{CIW}_{\text{PressAccelerator, Accelerator-On}}$	Representation state for Control of Information Weight for PressAccelerator→Accelerator-On
X_{107}	$\mathbf{CIW}_{\text{Accelerator-On, Engine-On}}$	Representation state for Control of Information Weight for Accelerator-On→Engine-On
X_{108}	$\mathbf{CIW}_{\text{Engine-On,Rev-Meter-On}}$	Representation state for Control of Information Weight for Engine-On→Rev-Meter-On
X_{109}	$\mathbf{CIW}_{\text{RevMeter-On,BiteState}}$	Representation state for Control of Information Weight for RevMeter-On→BiteState
X_{110}	$\mathbf{CIW}_{\text{Clutch-On,BiteState}}$	Representation state for Control of Information Weight for Clutch-On→BiteState
X_{111}	$\mathbf{CIW}_{\text{BiteState,PressAccelerator}}$	Representation state for Control of Information Weight for BiteState→PressAccelerator
X_{112}	$\mathbf{CIW}_{\text{Engine-On,MovingState}}$	Representation state for Control of Information Weight for Engine-On→MovingState
X_{113}	$\mathbf{CIW}_{\text{Gear-On1,MovingState}}$	Representation state for Control of Information Weight for Gear-On1→MovingState

https://doi.org/10.1371/journal.pone.0255503.t006

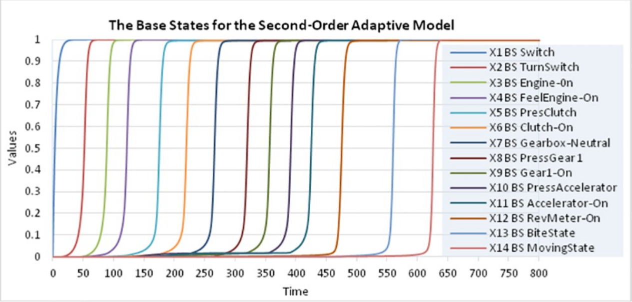

Fig 4. Dynamics of the base states X_1-X_{14} showing internal simulation of the mental model.

https://doi.org/10.1371/journal.pone.0255503.g004

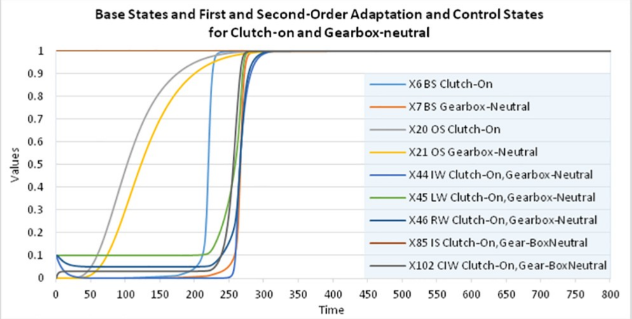

Fig 5. Base states X_6 (Clutch-on) and X_7 (Gearbox-neutral) with impact from OS-states X_{20}, X_{21} and LW-state X_{45}, RW-state X_{46} and learner IW-state X_{44} and instructor IS-state X_{85} with control by CIW-state X_{102}.

https://doi.org/10.1371/journal.pone.0255503.g005

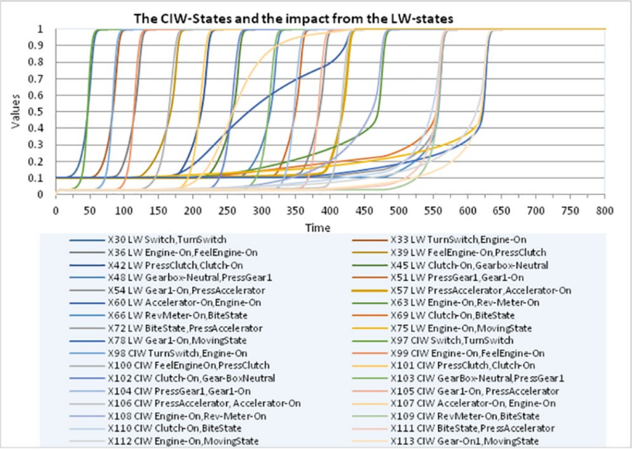

Fig 6. All control CIW-states showing impact from the corresponding observational learning LW-states.

https://doi.org/10.1371/journal.pone.0255503.g006

The learner hence controls the amount of information (s)he needs in addition to complete her/his learning based on her/his current level of understanding by own observation (see also Section 3). The results indicated in Fig 5 display the connection between two **BS**-states X_6 and X_7. Here it can be seen that, together with the value of X_6 becoming 1 at time 210, the **OS**-state X_{21} affects the value of X_7 together with **RW**-state $X_{46,}$ which combines the weights of the related **LW**-and **IW**-states. The **CIW**-state controls the weight of **IW**-state according to the **LW**-state's weight. State X_7 reaches value 1 at time 250 by an S-curve. The **IS**-state representing knowledge of the instructor remains at 1 all the time (a knowledgeable instructor).

As a form of evaluation, in Figs 6 and 7 it is displayed how the activation of each **CIW**-state indeed follows the activation of the corresponding **LW**-state, and how in turn the activation of the **CIW**-state indeed is followed by the corresponding **IW**-state. This confirms that the model displays the intended behavior that first observational learning takes place, after which there is a learner initiative to request corresponding instructional information, and appropriate instructional learning indeed takes place after that.

The simulation results presented by these figures are in accordance with and illustrate the educational science literature such as [28, 44] (as discussed in Section 2) on the use of learner-control of the timing of instruction.

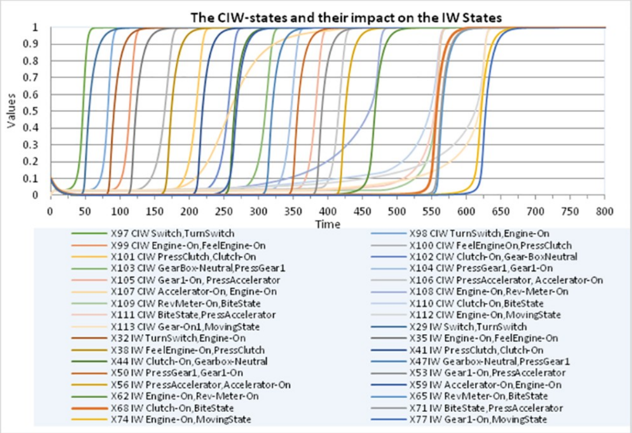

Fig 7. All control CIW-states showing their impact on the corresponding instructional learning IW-states.

https://doi.org/10.1371/journal.pone.0255503.g007

6 Verification of the network model by equilibrium analysis

To verify whether the introduced and implemented self-modeling network model behaves as expected from its design specification, a number of network states equilibrium values were analyzed for the example simulation.

6.1 Criterion for equilibria of self-modeling network models

A *stationary point* for a state Y occurs at time t if d$Y(t)$/dt = 0. An *equilibrium* occurs when all states have a stationary point simultaneously. From Eq (1) in Section 4, for any state Y for being stationary the following general *criterion* in terms of the network characteristics can be derived:

$$\boldsymbol{\eta}_Y = 0 \text{ or } \mathbf{aggimpact}_Y(t) = Y(t) \tag{6}$$

where

$$\mathbf{aggimpact}_Y(t) = \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t))$$

with X_1 to X_k the states from which Y gets its incoming connections.

The equation in (6) is also called an *equilibrium equation*. As a test, using the example simulation presented above for the apparent equilibrium at time 800, for 68 of the 113 network states it has been verified (independent of the implemented model) that the aggregated impact $\mathbf{aggimpact}_{Xi}$ (t) matches the state value for equilibrium values observed in the example simulation. In particular, this has been done for all 4x17 = 68 **LW**-, **IW**-, **RW**-, and **CIW**-states. The results are discussed in Sections 6.2 to 6.4.

6.2 Equilibrium analysis of the LW-states and the CIW-states

The **LW**-states use the Hebbian learning function $\mathbf{hebb}_{\mu}$ (V_1, V_2, W) as combination function. Using this function, by (5) for any **LW**-state Y holds

$$\mathbf{aggimpact}_Y(t) = \mathbf{hebb}_{\boldsymbol{\mu}}(V_{X_1}, V_{X_2}, V_{\mathbf{LW}}) = V_{X_1}V_{X_2}(1 - V_{\mathbf{LW}}) + \boldsymbol{\mu}V_{\mathbf{LW}} \tag{7}$$

where V_{X1}, V_{X2} are the state values of the connected base states X_1 and X_2 and $V_{\mathbf{LW}}$ the state value of **LW**-state Y. So, for this case the equilibrium equation in (6) becomes

$$V_{X_1}V_{X_2}(1 - V_{\mathbf{LW}}) + \boldsymbol{\mu}\, V_{\mathbf{LW}} = V_{\mathbf{LW}} \tag{8}$$

Assuming the denominator nonzero, this can also be rewritten into (also see [40], Section 3.6.1):

$$V_{\mathbf{LW}} = \frac{V_{X_1}V_{X_2}}{1 - \boldsymbol{\mu} + V_{X_1}V_{X_2}} \tag{9}$$

For the example simulation, it was set $\boldsymbol{\mu}$ = 1; therefore (8) is equivalent to

$$V_{X_1} = 0 \text{ or } V_{X_2} = 0 \text{ or } V_{\mathbf{LW}} = 1 \tag{10}$$

In the simulation, at t = 800 all **LW**-states have value 1 in a precision of 15 digits (and the values V_{X1}, and V_{X2} are always nonzero). Therefore, for all **LW**-states criterion (6) is fulfilled with deviations $< 10^{-15}$. This provides one piece of evidence that the implemented network model is correct with respect to its design specification.

The 17 **CIW**-states use the combination function **alogistic$_{10,0.4}$(..)** described by (4) and the weight of the connection from the related **LW**-state to the **CIW**-state is 1 so

$$\mathbf{aggimpact}_{\mathbf{CIW}} = \mathbf{alogistic}_{10,0.4}(V_{\mathbf{LW}}) \tag{11}$$

where $V_{\mathbf{LW}}$ is the value of the **LW**

Therefore, for this case the equilibrium equation from criterion (6) is

$$\mathbf{alogistic}_{10,0.4}(V_{\mathbf{LW}}) = V_{\mathbf{CIW}} \tag{12}$$

where $V_{\mathbf{CIW}}$ is the value of the **CIW**-state.

Now, as already found above at $t = 800$, for all **LW**-states $V_{\mathbf{LW}} = 1$ in a precision of 15 digits, and **alogistic$_{10,0.4}$**(1) = 0.997482089170521. Moreover, at $t = 800$ it is found that $V_{\mathbf{CIW}}$ = 0.997482089170520 for all **CIW**-states. This makes a deviation of 0.997482089170520–0.997482089170521 = -10^{-15}. This very small deviation provides a second piece of evidence that the implemented network model is correct with respect to its design specification.

6.3 Equilibrium analysis of the IW-states

The **IW**-states use the combination function **alogistic$_{10,0.7}$(..)** described by (4) and has incoming connections from themselves (with weight 1), and from the related **IS**-state. Moreover, the connection from this **IS**-state to the **IW**-state has weight represented by the related **CIW**-state, whereas the state values of the **IS**-states are constant 1. Therefore, it holds

$$\mathbf{aggimpact}_{\mathbf{IW}} = \mathbf{alogistic}_{10,0.7}(V_{\mathbf{IW}}, V_{\mathbf{CIW}}) \tag{13}$$

where $V_{\mathbf{IW}}$ is the value of the **IW**-state itself and $V_{\mathbf{CIW}}$ is the value of the **CIW**-state.

So, for this case the equilibrium equation from criterion (6) is

$$\mathbf{alogistic}_{10,0.7}(V_{\mathbf{IW}}, V_{\mathbf{CIW}}) = V_{\mathbf{IW}} \tag{14}$$

These values have been computed (independent of the implemented model), as shown in Table 7. Here the second and fifth column display the values for the **CIW**- and **IW**-state from the simulation at $t = 800$, and the values in the third and fourth columns were calculated based on that. The fourth column indicates the left hand side of the above Eq (14), the fifth column the right hand side and the sixth column the difference between the two. It turns out that all deviations are $< 10^{-7}$, which is a third piece of evidence that the implemented network model is correct with respect to its design specification.

6.4 Equilibrium analysis of the RW-states

The **RW**-states use the combination function **eucl$_{1,2}$(.,.)** described by (3), which makes the average of its two arguments. They have incoming connections with weight 1 from the related **LW**-state and **IW**-state. Therefore it holds

$$\mathbf{aggimpact}_{\mathbf{RW}} = \frac{V_{\mathbf{LW}} + V_{\mathbf{IW}}}{\mathbf{2}} \tag{15}$$

where $V_{\mathbf{LW}}$ is the value of the **LW**-state and $V_{\mathbf{IW}}$ is the value of the **IW**-state.

Then for this case the equilibrium equation from criterion (6) is

$$\frac{V_{\mathbf{LW}} + V_{\mathbf{IW}}}{\mathbf{2}} = V_{\mathbf{RW}} \tag{16}$$

where $V_{\mathbf{RW}}$ is the value of the **RW**-state. Like above, these values have been computed

Table 7. Equilibrium analysis results for the IW-states.

nr	V_{CIW}	$V_{CIW} + V_{IW}$	**aggimpact$_{IW}$**	V_{IW}	**deviation**
X_{29}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{32}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{35}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{38}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{41}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{44}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{47}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{50}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{53}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{56}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{59}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{62}	0.997482089170520	1.997479769043310	0.999997679872790	0.999997679872789	$-1.11022\ 10^{-15}$
X_{65}	0.997482089170520	1.997479769026860	0.999997679872790	0.999997679856344	$-1.64454\ 10^{-11}$
X_{68}	0.997482089170520	1.997479769036000	0.999997679872790	0.999997679865477	$-7.31293\ 10^{-12}$
X_{71}	0.997482089170520	1.997479769029510	0.999997679872790	0.999997679858985	$-1.38045\ 10^{-11}$
X_{74}	0.997482089170520	1.997479763727770	0.999997679872667	0.999997674557245	$-5.31542\ 10^{-9}$
X_{77}	0.997482089170520	1.997479758743110	0.999997679872551	0.999997669572591	$-1.03\ 10^{-8}$

https://doi.org/10.1371/journal.pone.0255503.t007

(independent of the implemented model) from the simulation values at $t = 800$ for the **IW** and **LW**-states, and compared to the simulation values of the **RW**-states as shown in Table 8. It turns out that all deviations are $< 10^{-8}$, which is a fourth piece of evidence that the implemented network model is correct with respect to its design specification.

7 Discussion

In this paper, a computational network model was presented for controlled learning of a mental model. Learning of a mental model often involves observational learning and instructional

Table 8. Equilibrium analysis results for the RW-states.

nr	V_{IW}	V_{LW}	**aggimpact$_{RW}$**	V_{RW}	**deviation**
X_{31}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{34}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{37}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{40}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{43}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{46}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{49}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{52}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{55}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{58}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{61}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{64}	0.999997679872789	1.000000000000000	0.999998839936394	0.999998839936394	$<10^{-15}$
X_{67}	0.999997679856344	1.000000000000000	0.999998839928172	0.999998839925431	$-2.74081\ 10^{-12}$
X_{70}	0.999997679865477	1.000000000000000	0.999998839932738	0.999998839931520	$-1.21869\ 10^{-12}$
X_{73}	0.999997679858985	1.000000000000000	0.999998839929493	0.999998839927192	$-2.30072\ 10^{-12}$
X_{76}	0.999997674557245	1.000000000000000	0.999998837278622	0.999998836392726	$-8.85897\ 10^{-10}$
X_{79}	0.999997669572591	1.000000000000000	0.999998834786295	0.999998833069649	$-1.71665\ 10^{-9}$

https://doi.org/10.1371/journal.pone.0255503.t008

learning. To obtain an effective learning process, appropriate timing of these types of learning is needed, which requires some form of control. For such control, a mental model adaptation process itself has to be made adaptive as well, which is a form of second-order adaptation for this mental model. So, all in all, a mental model can be used in three different manners: (1) it is executed to draw conclusions from it, (2) it is adapted to learn and improve it, and (3) these adaptation processes are controlled. These three properties, and their interplay require three different types of modeling that interact with each other. In this paper, the network-oriented modeling approach for self-modeling adaptive networks described in [23] was applied to address these processes for mental models. A generic three-level self-modeling network architecture introduced in [26] was applied to support this. Based on this general architecture, a second-order adaptive mental network model was presented, in which the base level includes a mental model as it can be used, the second level models a first-order adaptation process for the learning process of this mental model and the third level models a second-order adaptation process that controls the focus and timing of the types of learning.

It has turned out that the network self-modeling mechanism (also called network reification) fits very well to what is needed for (1), (2) and (3) for mental models. The idea of self-modeling networks was originally (in [22, 23]) mainly inspired by the extensive neuroscience literature on plasticity versus metaplasticity in the brain; e.g., [47–50]. Paper [26] was the first paper that demonstrated the usefulness of the same conceptualisation for the social domain of teaching and learning. As far as the authors know, there is no other computational model covering (1), (2) and (3).

The introduced network model was illustrated for a case study of learner-controlled mental model learning for how a car works, and driving it. Here the learner is in control of the use and timing of observational learning and instructional learning. Using the dedicated software environment described in [40], Ch. 9 the network model was implemented and simulated. By this it was shown to work as expected from the literature. Moreover, by verification of the implemented model based on equilibrium analysis (for a representative test set of 68 of the 113 network states), it was found that all deviations are $<10^{-7}$ (see Section 6). This provides strong evidence that the implemented model is correct with respect to its design specification. Further validation by comparison to empirical data would be interesting for future research; currently, such data are not available to the authors.

Much literature exists which describes the learning of mental models and was discussed in the paper. However, computational models addressing it are very rare; a few exceptions are [13, 19, 20, 53]. For example [20], addresses simulation of students' construction of energy models in physics in a production rule modeling format and in [13] the PDP modeling format was applied to model mental models. In [53] a mental God model was addressed, and in [19] the focus is on model-based learning to drive a car. In all four cases [13, 19, 20, 53] no control of the learning processes is modeled, which is a main difference with the current paper, where the focus is on the control and this is addressed by designing a second-order adaptive mental network model.

In the meantime, the self-modeling network modeling perspective [22, 23] and the general three-level second-order adaptive network architecture introduced in [26] and also described in more detail in the current paper to model dynamics, adaptation of control for mental models, has been found to be very general and applicable for handling mental models in many other application cases where mental models are used. For example, in [54] this architecture and also the learning mechanisms contributed by [26] have been applied successfully to model how shared mental models are used in hospital teamwork. A more detailed overview of this general approach to mental models originating to a large extent in [26] and many of its applications will be presented in the forthcoming book [55].

However, note that the literature on mental models is very diverse. Therefore, although having turned out applicable in many cases, it cannot be claimed that the way in which mental models are addressed from a network-oriented perspective here, would be applicable for all forms of mental models addressed in the literature.

Supporting information

S1 Appendix. Full specification of the second-order adaptive network model.
(DOCX)

Author Contributions

Conceptualization: Rajesh Bhalwankar, Jan Treur.

Formal analysis: Jan Treur.

Investigation: Rajesh Bhalwankar, Jan Treur.

Methodology: Rajesh Bhalwankar, Jan Treur.

Software: Rajesh Bhalwankar, Jan Treur.

Supervision: Jan Treur.

Writing – original draft: Rajesh Bhalwankar, Jan Treur.

References

1. Buckley BC. Interactive multimedia and model-based learning in biology. International Journal of Science Education 2000; 22(9): 895–935.
2. Craik KJW. The Nature of Explanation. Cambridge University Press; 1943.
3. De Kleer J, Brown J. Assumptions and ambiguities in mechanistic mental models. In: Gentner D, Stevens A, editors. Mental models. Hillsdale NJ: Lawrence Erlbaum; 1983. pp. 155–190.
4. Ellison CG, Bradshaw M, Kuyel N, Marcum JP. Attachment to God, stressful life events, and changes in psychological distress. Rev Relig Res. 2012; 53(4): 493–511.
5. Gentner D, Stevens AL. Mental models. Hillsdale NJ: Erlbaum; 1983.
6. Johnson-Laird PN. Mental models: Towards a cognitive science of language, inference, and consciousness. Harvard University Press; 1983.
7. Johnson-Laird PN. Mental models. In: Posner MI, editor. Foundations of cognitive science. MIT Press; 1989. pp. 469–499.
8. Kim D. The link between individual and organizational learning. In: Starkey K,Tempest S, Mckinlay A, editors. How Organizations Learn, 2nd ed. London: Thomson Learning; 2004. pp. 29–50.
9. Mayer RE. Models for understanding. Review of educational research 1989; 59(1): 43–64.
10. Olson JR. The what and why of mental models in human computer interaction. In: Booth PA, editor. Mental Models in Everyday Activities. Robinson College; 1992. pp. 132–146
11. Piaget J. Origins of intelligence in the child. London: Routledge & Kegan Paul; 1936.
12. Rouse WB, Morris NM: On looking into the black box: Prospects and limits in the search for mental models. Psychological Bulletin 1986; 100(3): 349–363.
13. Rumelhart DE, Smolensky P, McClelland JL, Hinton GE: Schemata and sequential thought processes in PDP models. In: McClelland JL, Rumelhart DE. The PDP research group, editors. Parallel distributed processing. Explorations in the microstructure of cognition. Psychological and biological models, vol 2. Cambridge MA: MIT Press; 1986. pp. 7–57.
14. Seel NM. World knowledge and mental models (in German). Göttingen: Hogrefe; 1991.
15. Seel NM. Mental models in learning situations. In: Advances in Psychology, vol. 138. Amsterdam: North-Holland; 2006. pp. 85–107.
16. Skemp RR. The Psychology of Learning Mathematics, Penguin Books, Harmondsworth; 1971

17. Fein RM, Olson GM, Olson JS. A mental model can help with learning to operate a complex device. In: The Interact'93 and Chi'93 conferences companion on Human factors in computing systems. New York: ACM Press; 1993. pp. 157–158.

18. Kieras DE, Bovair S. The role of a mental model in learning to operate a device. Cognitive science 1984; 8(3): 255–273.

19. Bhalwankar R, Treur J. Modeling the Development of Internal Mental Models by an Adaptive Network Model. In: Proc. of the 11th Annual International Conference on Brain-Inspired Cognitive Architectures for AI, BICA*AI'20. Procedia Computer Science Elsevier vol. 190; 2021. pp. 90–101.

20. Devi R, Tiberghien A, Baker MJ, Brna P. Modelling Students' construction of energy models in physics. Instructional Science 1996; 24: 259–293.

21. Greca IM, Moreira MA. Mental models, conceptual models, and modelling. International journal of science education 2000; 22(1): 1–11.

22. Treur J. Modeling Higher-Order Adaptivity of a Network by Multilevel Network Reification. Network Science 2020; 8: S110–S144.

23. Treur J. Network-Oriented Modeling for Adaptive Networks: Designing Higher-order Adaptive Biological, Mental and Social Network Models. Cham, Switzerland: Springer Nature; 2020.

24. Treur J. Multilevel Network Reification: Representing Higher-order Adaptivity in a Network. In: Proc. of the 7th International Conference on Complex Networks and their Applications, ComplexNetworks'18, vol. 1. Studies in Computational Intelligence, vol. 812. Cham, Switzerland: Springer Nature; 2018. pp. 635–651.

25. Hebb DO. The organization of behavior: A neuropsychological theory. Wiley; 1949.

26. Bhalwankar R, Treur J. A Second-Order Adaptive Network Model for Learner-Controlled Mental Model Learning Processes. In: Benito RM, Cherifi C, Cherifi H, Moro E, Rocha LM, Sales-Pardo M, editors. Complex Networks & Their Applications IX. Proceedings COMPLEX NETWORKS 2020. Studies in Computational Intelligence, vol. 944, Springer Nature Switzerland AG; 2021. pp. 245–259.

27. Bandura A, Walters RH. Social learning theory, vol. 1. Englewood Cliffs, NJ: Prentice-hall; 1977.

28. Kozma RB. Learning with media. Review of educational research 1991; 61(2): 179–211.

29. Meela P, Yuenyong C. The study of grade 7 mental model about properties of gas in science learning through model based inquiry (MBI). In: Proc. of the International Conference for Science Educators and Teachers. AIP Conference Proceedings, vol. 2081(030028). AIP Publishing LLC; 2019. pp. 1–6.

30. Moray N. The psychodynamics of human-machine interaction. In: Harris D, editor, Engineering Psychology and Cognitive Ergonomics, vol. 4. Ashgate; 1999. pp. 225–235.

31. Neilson D, Campbell T, Allred B. Model-based inquiry: A buoyant force module for high school physics classes. The Science Teacher 2010; 77(8): 38–43.

32. Nersessian N. Should physicists preach what they practice? Science and Education 1995; 4: 203–226.

33. Norman DA. Some observations on mental models. In: Gentner D, Stevens AL, editors, Mental models, pp. 7–14. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers Inc.; 1983.

34. Potvin P, et al. Models of conceptual change in science learning: establishing an exhaustive inventory based on support given by articles published in major journals, Studies in Science Education 2020; 56 (2): 157–211, https://doi.org/10.1080/03057267.2020.1744796

35. Piaget J. The construction of reality in the child. New York: Basic Books Inc; 1954.

36. Benbassat J. Role modeling in medical education: the importance of a reflective imitation. Academic Medicine 1954; 89(4): 550–554.

37. Doll BB, Simon DA, Daw ND. The ubiquity of model-based reinforcement learning. Current Opinion in Neurobiology 2012; 22: 1075–1081. https://doi.org/10.1016/j.conb.2012.08.003 PMID: 22959354

38. Halloun I. Schematic modelling for meaningful learning of physics. Journal of Research in Science Teaching 1996; 33: 1019–1041.

39. Yi MY, Davis FD: Developing and validating an observational learning model of computer software training and skill acquisition. Information Systems Research 2003; 14(2): 146–169.

40. Hurley S. The shared circuits model (SCM): How control, mirroring, and simulation can enable imitation, deliberation, and mindreading. Behavioral and brain sciences 2008; 31(1): 1–22.

41. Rizzolatti G, Craighero L. The mirror-neuron system. Annu. Rev. Neurosci. 2004; 27: 169–192. https://doi.org/10.1146/annurev.neuro.27.070203.144230 PMID: 15217330

42. Van Gog T, Paas F, Marcus N, Ayres P, Sweller J: The mirror neuron system and observational learning: Implications for the effectiveness of dynamic visualizations. Educational Psychology Review 2009; 21(1): 21–30.

43. Hogan KE, Pressley ME. Scaffolding student learning: Instructional approaches and issues. Brookline Books; 1997.

44. Gibbons J, Gray M. An integrated and experience-based approach to social work education: The Newcastle model. Social Work Education 2002; 21(5): 529–549.

45. Annett J, Kay H. Knowledge of results and skilled performance. Occupational Psychology 1957; 31: 69–79.

46. Welford AT. Fundamentals of skill. London: Methuen; 1968.

47. Abraham WC, Bear MF. Metaplasticity: the plasticity of synaptic plasticity. Trends in Neuroscience 1996; 19(4): 126–130. https://doi.org/10.1016/s0166-2236(96)80018-x PMID: 8658594

48. Garcia R. Stress, metaplasticity, and antidepressants. Current Molecular Medicine 2002; 2: 629–638 https://doi.org/10.2174/1566524023362023 PMID: 12420802

49. Magerl W, Hansen N, Treede RD, Klein T. The human pain system exhibits higher-order plasticity (metaplasticity). Neurobiology of Learning and Memory 2018; 154: 112–120 https://doi.org/10.1016/j.nlm.2018.04.003 PMID: 29631001

50. Robinson BL, Harper NS, McAlpine D. Meta-adaptation in the auditory midbrain under cortical influence. Nat. Commun. 2016; 7;13442. https://doi.org/10.1038/ncomms13442 PMID: 27883088

51. Shatz CJ. The developing brain. Sci. Am. 1992; 267: 60–67. https://doi.org/10.1038/scientificamerican0992-60 PMID: 1502524

52. Keysers C, Gazzola V. Hebbian learning and predictive mirror neurons for actions, sensations and emotions. Philos Trans R Soc Lond B Biol Sci 2014; 369: 20130175. https://doi.org/10.1098/rstb.2013.0175 PMID: 24778372

53. Van Ments L, Roelofsma P, Treur J. Modelling the effect of religion on human empathy based on an adaptive temporal–causal network model. Comput. Soc. Netw. 2018; 5(1): e1.

54. Treur J, Van Ments L, editors. Mental Models and their Dynamics, Adaptation, and Control: a Self-Modeling Network Modeling Approach. Springer Nature Switzerland; 2022. forthcoming.

55. Van Ments L, Treur J, Klein J, Roelofsma PHMP. A Second-Order Adaptive Network Model for Shared Mental Models in Hospital Teamwork. In: Nguyen NT, et al., editors. Proc. of the 13th International Conference on Computational Collective Intelligence, ICCCI'21. Lecture Notes in AI, Springer Nature; 2021. forthcoming.