Modeling Mental Models

Canbaloğlu, Gülay; Treur, Jan

published in
Computational Modeling of Multilevel Organisational Learning and its Control Using Self-Modeling Network Models
2023

DOI (link to publisher)
10.1007/978-3-031-28735-0_4

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

citation for published version (APA)
Canbaloğlu, G., & Treur, J. (2023). Modeling Mental Models: Their Use, Adaptation and Control. In G. Canbaloğlu, J. Treur, & A. Wiewiora (Eds.), *Computational Modeling of Multilevel Organisational Learning and its Control Using Self-Modeling Network Models* (pp. 51-70). (Studies in Systems, Decision and Control; Vol. 468). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-031-28735-0_4

Chapter 4
Modeling Mental Models: Their Use, Adaptation and Control

Gülay Canbaloğlu and Jan Treur

Abstract Using a proper mental model during mental processes is often crucial. Such a mental model has to be learnt and maintained; this involves mental model adaptation. Metacognition is applied to control use and adaptating in a context-sensitive manner. In this chapter, a second-order adaptive network model for handling mental models, covering their use, adaptation and control, is discussed and used to illustrate these processes.

Keywords Adaptive network models · Mental models

4.1 Introduction

Mental processes often use specific mental models, e.g., Gentner and Stevens (1983), Greca and Moreira (2000), Skemp (1971), Seel (2006), Treur and Van Ments (2022). Learning or adaptation of a mental model has a decisive effect on the process of using it. Moreover, metacognitive control determines when and how to focus on learning. Metacognition (Darling-Hammond et al. 2008; Shannon 2008; Mahdavi 2014; Flavell 1979; Koriat 2007; Pintrich 2000) is a form of cognition about cognition. In Koriat (2007) it is described as what people know about their own cognitive processes and how they put that knowledge to use in regulating their cognitive processing and behavior. So, metacognition can be used to control one's own cognitive processes. In the context of mental models such control can apply to the use of a mental model or to its learning or adaptation. A sometimes used closely related term is self-regulation and when the cognitive processes addressed by metacognition concern learning, the term self-regulated learning is used. For example, in Pintrich

G. Canbaloğlu · J. Treur (✉)
Social AI Group, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, Netherlands
e-mail: j.treur@vu.nl

G. Canbaloğlu
e-mail: gulaycnbl@gmail.com

G. Canbaloğlu et al. (eds.), *Computational Modeling of Multilevel Organisational Learning and Its Control Using Self-modeling Network Models*, Studies in Systems, Decision and Control 468, https://doi.org/10.1007/978-3-031-28735-0_4

(2000), self-regulated learning is described as an active, constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognition, motivation, and behavior, guided and constrained by their goals and context.

From a network-oriented modeling perspective, adaptation of mental models can be described by adaptive network models, where some of the network characteristics such as connection weights or excitability thresholds change over time. If, in addition, metacognition is used to regulate or control the change process, this implies that the adaptation of the mental network is itself adaptive as well, which is called second-order adaptation. Thus, a network model for these processes has to address such complex second-order adaptive behaviour; e.g., Bhalwankar and Treur (2021), Van Ments and Treur (2021), Treur and Van Ments (2022). Following this line in the current chapter, using the self-modeling network modeling approach for higher-order adaptive networks from Treur (2018, 2020a, b), a second-order adaptive network model is introduced for metacognitive control over adaptation of a specific mental model.

In this chapter, first in Sect. 4.2 more background knowledge is discussed on metacognition and its role in controlling mental processes and a three-level cognitive architecture for it. In Sect. 4.3 the network-oriented modeling approach used is briefly explained and in Sect. 4.4 it is shown how it can be used to formalize the cognitive architecture as a self-modeling network. Next, in Sect. 4.5 the introduced second-order adaptive network model is described in some detail. In Sect. 4.6, it is shown how this model was used to perform simulations for the illustrative example scenario. Finally, Sect. 4.7 is a discussion and Sect. 4.8 is an Appendix with the full specification of the introduced adaptive network model.

4.2 A Three-Level Cognitive Architecture for Mental Models and Their Use, Adaptation and Control

For the history of the mental model area, often Kenneth Craik is mentioned as a central person. In his book (Craik 1943), he describes a mental model as a *small-scale model* that is carried by an organism within its head as follows:

> 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, utilize 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 1943, p. 61)

This quote mentions both the usage of a mental model by internal mental simulation (‘try out various alternatives’) and the learning of the mental model (‘utilize the knowledge of past events’). Shih and Alessi (1993, p. 157) indicate that mental models are relational structures consisting of states and causal relationships between them:

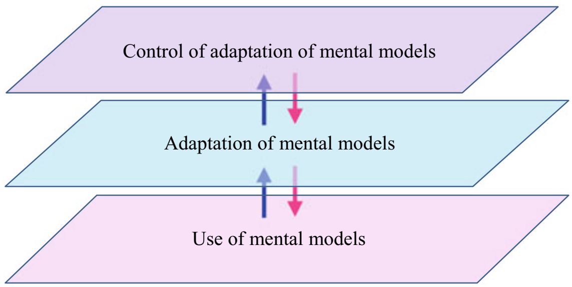

Fig. 4.1 Cognitive architecture for mental model handling with three levels of mental processing for mental models where each next level is monitoring and modulating or controlling the processes at the level below it: (1) the adaptation on the middle level is decisive for the use of a mental model on the base level, and (2) the adaptation itself on the middle level is controlled by the upper level

> By a mental model we mean a person's understanding of the environment. It can represent different states of the problem and the causal relationships among states.

By Van Ments and Treur (2021), an analysis of different types and uses of mental models is provided. This analysis resulted into a three-level cognitive architecture (see Fig. 4.1):

- base level: internal simulation of a mental model
- middle level: the adaptation of the mental model (e.g., formation, learning, revising, and forgetting); this adaptation modulates or controls the internal simulation processes at the base level
- upper level: the (metacognitive) control over the adaptation processes

Learning of mental models can involve observational and instructional learning (Yi and Davis 2003; Van Gog et al. 2009) and combinations thereof, but also learning by mental simulation or by just using a mental model in practice can take place.

Literature on metacognition, sometimes also called self-regulation, can be found, for example in Darling-Hammond et al. (2008), Shannon (2008), Mahdavi (2014), Flavell (1979), Koriat (2007), Pintrich (2000). The focus is here on the role of metacognition applied to learning or adaptation. For example, in Pintrich (2000, pp. 452–453) the following assumptions for self-regulated learning are described by 'Learners can monitor, control, and regulate certain aspects of their own cognition, motivation, and behavior, and some elements of their environment.'. In Koriat (2007, p. 290), metacognition is described by what people know about cognition and their own cognitive processes, and how they use that in regulating their cognitive processes and behavior. It is emphasized that there is a causal relation from monitoring to control (Koriat 2007), p. 315.

So, in both descriptions of Pintrich (2000) and Koriat (2007) on metacognition (as well as in most other literature on metacognition), monitoring and control of the own cognitive processes are central concepts. These processes work through a causal cycle where the own cognitive processes affect the metacognitive monitoring, this monitoring in turn affects the metacognitive control, and this control affects the own cognitive processes. Such a causal cycle is indicated by the upward and downward arrows in Fig. 4.1.

4.3 Higher-Order Adaptive Network Models

In this section, the network-oriented modeling approach used is briefly introduced. Following Treur (2016, 2020b), a temporal-causal network model is characterised by (here X and Y denote nodes of the network, also called states):

- *Connectivity characteristics*
- Connections from a state X to a state Y and their weights $\boldsymbol{\omega}_{X,Y}$
- *Aggregation characteristics*

  For any state Y, some combination function $\mathbf{c}_Y(..)$ defines the aggregation that is applied to the impacts $\boldsymbol{\omega}_{X,Y}X(t)$ on Y from its incoming connections from states X
- *Timing characteristics*

  Each state Y has a speed factor $\boldsymbol{\eta}_Y$ defining how fast it changes for given causal impact.

The following difference (or related differential) equations that are used for simulation purposes and also for analysis of temporal-causal networks, incorporate these network characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$ in a standard numerical format:

$$Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]\Delta t \quad (4.1)$$

for any state Y and where X_1 to X_k are the states from which Y gets its incoming connections. Within the software environment described in Treur (2020b, Chap. 9), a large number of around 70 useful basic combination functions are included in a combination function library. The above concepts enable us to design network models and their dynamics in a declarative manner, based on mathematically defined functions and relations.

Realistic network models are usually adaptive: often not only their states but also some of their network characteristics change over time. By using a *self-modeling network* (also called a *reified* network), a similar network-oriented conceptualisation can also be applied to *adaptive* networks to obtain a declarative description using mathematically defined functions and relations for them as well, for more details, see Treur (2018, 2020a, b), see also Chap. 3 of this volume (Canbaloğlu et al. 2023). In brief, this works through the addition of new states to the network (called *self-model states*) which represent (adaptive) network characteristics. In the graphical 3D-format as shown in Sect. 4.4, such additional states are depicted at a next level (called *self-model level* or *reification level*), where the original network is at the *base level*. As an example, the weight $\boldsymbol{\omega}_{X,Y}$ of a connection from state X to state Y can be represented (at a next self-model level) by a self-model state named $\mathbf{W}_{X,Y}$. Similarly, all other network characteristics from $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$ can be made adaptive by including self-model states for them. For example, an adaptive speed factor $\boldsymbol{\eta}_Y$ can be represented by a self-model state named $\mathbf{H}_Y$ and an adaptive excitability threshold parameter $\boldsymbol{\tau}_Y$ can be represented by a self-model state named $\mathbf{T}_Y$. Moreover, a persistence factor $\boldsymbol{\mu}$ of a state Y of used for adaptation can be represented by a self-model state $\mathbf{M}_Y$.

As the outcome of such a process of network reification is also a temporal-causal network model itself, as has been shown in Treur (2020b, Chap. 10), this self-modeling network construction can easily be applied iteratively to obtain multiple orders of self-models at multiple (first-order, second-order, …) self-model levels. In the current chapter, this multi-level self-modeling network perspective will be applied to obtain a second-order adaptive network model addressing metacognitive control of adaptation as needed to learn and use a specific mental model.

4.4 Modeling Adaptation of a Mental Model and Its Metacognitive Control by Self-Modeling Networks

In this section, the adaptive self-modeling network model for mental models is introduced. This adaptive network model follows the cognitive architecture that has processes at three levels (Van Ments and Treur 2021), described in Sect. 4.3.

By using the notion of self-modeling network (or reified network) from Treur (2020a, b), recently this cognitive architecture has been formalized computationally and used in computer simulations for applications of mental models; for an overview of this approach and various applications of it, see Treur and Van Ments (2022). This cognitive architecture can be formalized as a self-modeling network model as shown in Fig. 4.2.

More specifically, the mapping from the three levels of the cognitive architecture to this self-modeling network architecture is as follows (see also Fig. 4.2):

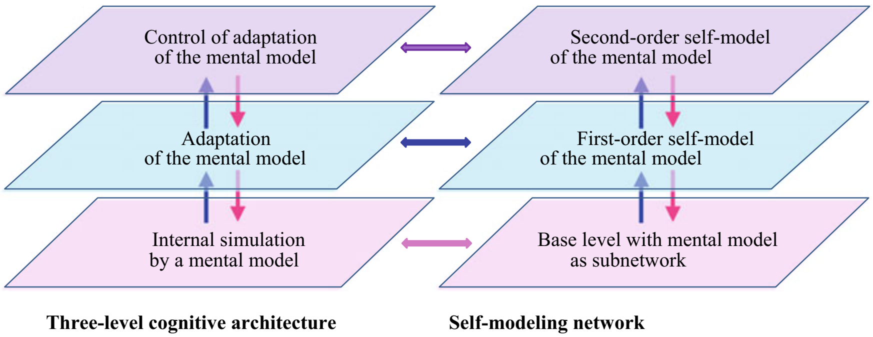

Fig. 4.2 Modeling the three-level cognitive architecture for mental model handling by a self-modeling network

- **Lower level: a mental model as a subnetwork**

  The mental model as a relational structure at the base level within the cognitive architecture is modeled as a (sub)network structure of states (nodes) and connections between them at the base level of the self-modeling network; the dynamics of the states of this subnetwork model internal simulation of the mental model.

- **Middle level: a first-order self-model of the mental model representing adaptation of its network structure**

  The level for adaptation of the mental model within the cognitive architecture is modeled as a first-order self-model of the mental model structure by the base level network; the dynamics of the states of this first-order self-model model adaptation makes changes in the structure of the mental model.

- **Upper level: a second-order self-model of a mental model representing control of adaptation of its network structure**

  The level for control of adaptation of a mental model is modeled as a second-order self-model of the base network for the mental model, which is a self-model for the self-model for adaptation of the mental model; the dynamics of the states of this second-order self-model model control of adaptation by making changes in the structure of the first-order self-model that describes the adaptation of the mental model.

So, mental models and the way they are handled can be considered as being described through multiple representations: they can be viewed from three levels of representation according to the three planes depicted in Fig. 4.2, right hand side. At the lower, base level depicted by the lower (pink) plane, a mental model, which in general essentially is considered a relational structure, is represented by nodes and connections between these nodes. For internal simulation, the nodes have activation levels that vary over time. Based on the relations between the nodes, these activation levels affect each other over time. Next, at the adaptation level depicted in Fig. 4.2 right-hand side by the middle (blue) plane, it is represented how the network connections representing the mental model relations, change over time by some adaptation model. Finally, at the top level depicted by the upper (purple) plane in Fig. 4.2 it is indicated how the adaptation at the middle level is controlled. In this way, to model mental processes in which mental models play a role, within the self-modeling network these mental models do not get a single but a three-fold representation by which the different uses and operations on the mental model are distinguished like they are distinguished by the levels in the cognitive architecture.

4.5 A Second-Order Adaptive Mental Network Model for Metacognitive Control of Adaptation of a Mental Model

The three-level architecture for mental model handling discussed in Sect. 4.4 (see Fig. 4.2) is illustrated in more detail by an example second-order adaptive self-modeling network model. The connectivity of this network model is depicted (according to three different views) in Figs. 4.3, 4.5 and 4.6. The states used are explained in Fig. 4.4. As a case study, in this second-order self-modeling network model, mental models for indirect (big blue ovals in Fig. 4.3) and direct (big purple ovals in Fig. 4.3) communication scenarios are modeled addressing the learning process of people trying to communicate with each other. Direct communication means the direct interaction between two people via asking a question and replying to it. On the other hand, indirect communication requires one or more intermediary people to interact with the target person. The scenario was constructed in the form of a request for an appointment to focus on the understanding of the process. For indirect communication, in the example scenario a mental model is used in which person A asks the personal assistant PA of person B for an appointment. PA relays the request to B and receives the reply of B. Then PA transfers B's reply to A and gets A's notification receipt to let B know about this notification. In contrast, for direct communication, a mental model is used in which A and B interact with each other, therefore, there is no need for a personal assistant. A directly asks B for the appointment and can get the answer immediately.

In the introduced adaptive network model, mental model connections are represented by adaptive first-order self-model states called **W**-states. Control of the adaptation uses second-order self-model states (called $\mathbf{H_W}$-states) for the adaptation speed of the **W**-states. A **W**-state can change when the corresponding $\mathbf{H_W}$-state representing the adaptation speed becomes nonzero; when this $\mathbf{H_W}$-state is 0, no change can take place. The $\mathbf{H_W}$-states can become nonzero in a context-sensitive manner because the context states have connections to them. This is how the context-sensitive metacognitive control over the change of a mental model takes place, thereby following the second-order adaptation principle 'Adaptation accelerates with increasing stimulus exposure' as formulated by Robinson et al. (2016).

The second-order self-model states called $\mathbf{M_W}$-states model the persistence of the mental model connections represented by the related **W**-states. If $\mathbf{M_W}$-states have activation level 1, there is full persistence, if they have level 0 there is no persistence at all. These second-order self-model states (together with the $\mathbf{H_W}$-states) model in a context-sensitive manner the 'Plasticity versus Stability Conundrum' described, for example, in Sjöström et al. (2008). They are used for the Hebbian learning (and forgetting) that takes place during the use of a mental model for internal mental simulation and to change the mental model in focus to another mental model.

Figure 4.3 depicts the view on the base level and first-order self-model level and the interactions between these two levels. From the context state for the indirect interaction case, by upward connections (in blue) the **W**-states for the mental model

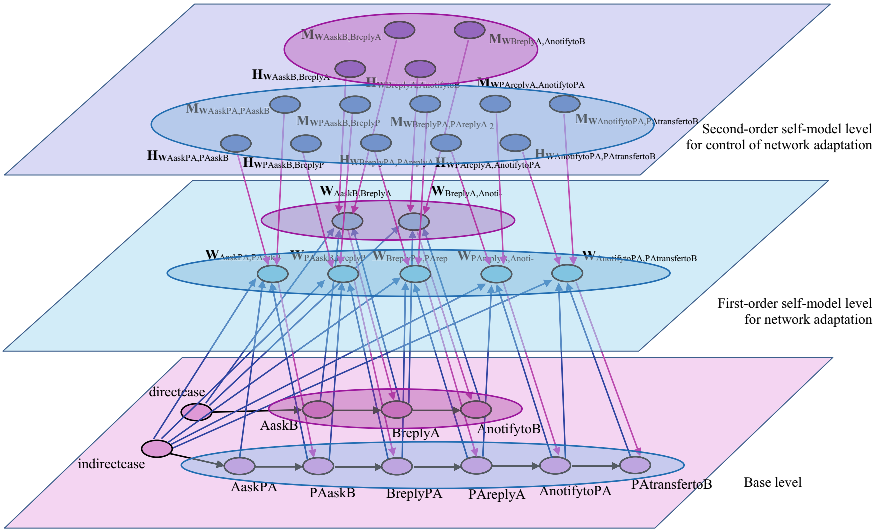

Fig. 4.3 Graphical representation of the connectivity of the second-order adaptive network model for metacognitive control of learning and use of a mental model: view on base level and first-order self-modeling level and their interaction. Here the upward connections to the second-order self-model level have not been depicted for the sake of transparency

for the indirect case are activated (thereby assuming that these **W**-states have a nonzero adaptation speed, see below and Fig. 4.5) and similarly for the direct case. Within the base level, the two context states also have connections to the first state of these mental models, so that these mental models are actually used for internal simulation.

Figure 4.5 shows the view on the interactions of the second-order self-model level with the other two levels for the $\mathbf{H}_\mathbf{W}$-states. It can be seen here that by upward connections (in blue), the adaptation speed as represented by the $\mathbf{H}_\mathbf{W}$-states depends on activation of the related mental model states at the base level and the related connection representation **W**-states at the first-order self-model level. This follows the second-order adaptation principle 'Adaptation accelerates with increasing stimulus exposure' (Robinson et al. 2016) mentioned above and makes that by the downward connections (in pink) from the $\mathbf{H}_\mathbf{W}$-states to the **W**-states, these **W**-states indeed can change. All this depends on the context state that triggers activation of these specific base states and **W**-states for a mental model, which makes it a context-sensitive form of control of adaptation.

Figure 4.6 shows the view on the interactions of the second-order self-model level with the other levels in particular for the $\mathbf{M}_\mathbf{W}$-states. The $\mathbf{M}_\mathbf{W}$-states have values 1 when the related **W**-states needs to be fully persistent and lower values when these **W**-states should not be (fully) persistent. As mentioned above, this models in a

Nr	State	Explanation	Level
X_1	AaskPA	Mental model state for person A asks PA for appointment to personal assistant of person B	Base level
X_2	PAaskB	Mental model state for personal assistant asks B for appointment for person A	
X_3	BreplyPA	Mental model state for person B replies to PA for appointment	
X_4	PAreplyA	Mental model state for personal assistant replies to person A for appointment	
X_5	AnotifytoPA	Mental model state for person A notifies receipt to personal assistant of B	
X_6	PAtransfertoB	Mental model state for personal assistant transfers receipt of A to person B	
X_7	AaskB	Mental model state for person A directly asks person B for appointment	
X_8	BreplyA	Mental model state for person B directly replies to person A for appointment	
X_9	AnotifytoB	Mental model state for person A directly notifies receipt to person B	
X_{10}	indirectcase	Context state for the indirect communication (option 1)	
X_{11}	directcase	Context state for the direct communication (option 2)	
X_{12}	$\mathbf{W}_{\text{AaskPA,PAaskB}}$	Self-model state for speed factor for connection weight between X_1 and X_2	First-order self-model level
X_{13}	$\mathbf{W}_{\text{PAaskB,BreplyPA}}$	Self-model state for speed factor for connection weight between X_2 and X_3	
X_{14}	$\mathbf{W}_{\text{BreplyPA,PAreplyA}}$	Self-model state for speed factor for connection weight between X_3 and X_4	
X_{15}	$\mathbf{W}_{\text{PAreplyA,AnotifytoPA}}$	Self-model state for speed factor for connection weight between X_4 and X_5	
X_{16}	$\mathbf{W}_{\text{AnotifytoPA,PAtransfertoB}}$	Self-model state for speed factor for connection weight between X_5 and X_6	
X_{17}	$\mathbf{W}_{\text{AaskB,BreplyA}}$	Self-model state for speed factor for connection weight between X_7 and X_8	
X_{18}	$\mathbf{W}_{\text{BreplyA,AnotifytoB}}$	Self-model state for speed factor for connection weight between X_8 and X_9	
X_{19}	$\mathbf{H}_{\mathbf{W}\text{AaskPA,PAaskB}}$	Self-model state for speed factor for self-model state X_{12}	Second-order self-model level
X_{20}	$\mathbf{H}_{\mathbf{W}\text{PAaskB,BreplyPA}}$	Self-model state for speed factor for self-model state X_{13}	
X_{21}	$\mathbf{H}_{\mathbf{W}\text{BreplyPA,PAreplyA}}$	Self-model state for speed factor for self-model state X_{14}	
X_{22}	$\mathbf{H}_{\mathbf{W}\text{PAreplyA,AnotifytoPA}}$	Self-model state for speed factor for self-model state X_{15}	
X_{23}	$\mathbf{H}_{\mathbf{W}\text{AnotifytoPA,PAtransfertoB}}$	Self-model state for speed factor for self-model state X_{16}	
X_{24}	$\mathbf{H}_{\mathbf{W}\text{AaskB,BreplyPA}}$	Self-model state for speed factor for self-model state X_{17}	
X_{25}	$\mathbf{H}_{\mathbf{W}\text{BreplyA,AnotifytoB}}$	Self-model state for speed factor for self-model state X_{18}	
X_{26}	$\mathbf{M}_{\mathbf{W}\text{AaskPA,PAaskB}}$	Self-model state for persistence factor for self-model state X_{12}	
X_{27}	$\mathbf{M}_{\mathbf{W}\text{PAaskB,BreplyPA}}$	Self-model state for persistence factor for self-model state X_{13}	
X_{28}	$\mathbf{M}_{\mathbf{W}\text{BreplyPA,PAreplyA}}$	Self-model state for persistence factor for self-model state X_{14}	
X_{29}	$\mathbf{M}_{\mathbf{W}\text{PAreplyA,AnotifytoPA}}$	Self-model state for persistence factor for self-model state X_{15}	
X_{30}	$\mathbf{M}_{\mathbf{W}\text{AnotifytoPA,PAtransfertoB}}$	Self-model state for persistence factor for self-model state X_{16}	
X_{31}	$\mathbf{M}_{\mathbf{W}\text{AaskB,BreplyPA}}$	Self-model state for persistence factor for self-model state X_{17}	
X_{32}	$\mathbf{M}_{\mathbf{W}\text{BreplyA,AnotifytoB}}$	Self-model state for persistence factor for self-model state X_{18}	

Fig. 4.4 The states in the adaptive network model

context-sensitive manner the 'Plasticity versus Stability Conundrum' described, for example, in Sjöström et al. (2008). The context-dependence of this form of control is modeled specifically by the upward connections (in blue) from the two context states at the base level to their related $\mathbf{M}_{\mathbf{W}}$-states.

The combination functions from the library used in the introduced network model are defined as follows:

- The *advanced logistic sum combination function* $\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k)$ is defined by:

$$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = \left[\frac{1}{1 + \mathrm{e}^{-\sigma(V_1 + \cdots + V_k - \tau)}} - \frac{1}{1 + \mathrm{e}^{\sigma\tau}}\right]\left(1 + \mathrm{e}^{-\sigma\tau}\right) \quad (4.2)$$

where σ is a steepness parameter and τ a threshold parameter and $V_1, \ldots, V_k$ are the impacts from the states from which the considered state Y gets incoming connections

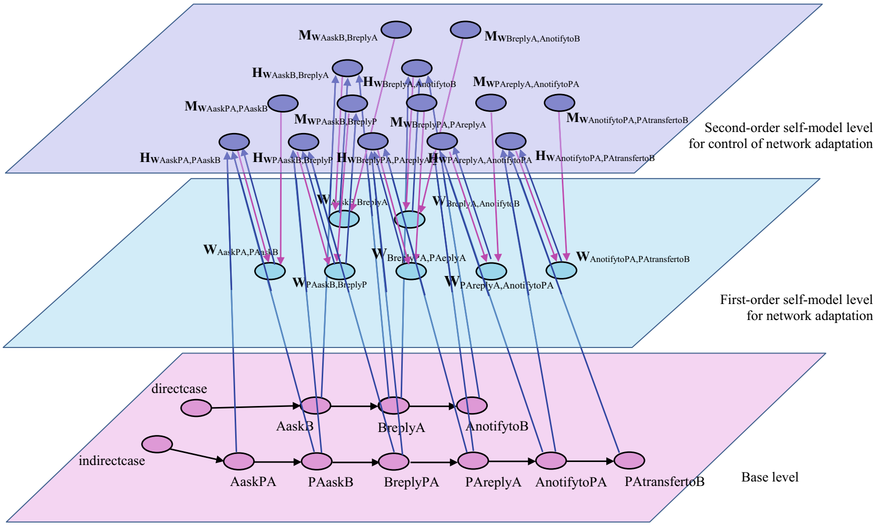

Fig. 4.5 Graphical representation of the connectivity of the second-order adaptive mental network model for metacognitive control of focusing on a mental model. View for the interactions of the second-order self-model level $\mathbf{H_W}$-states

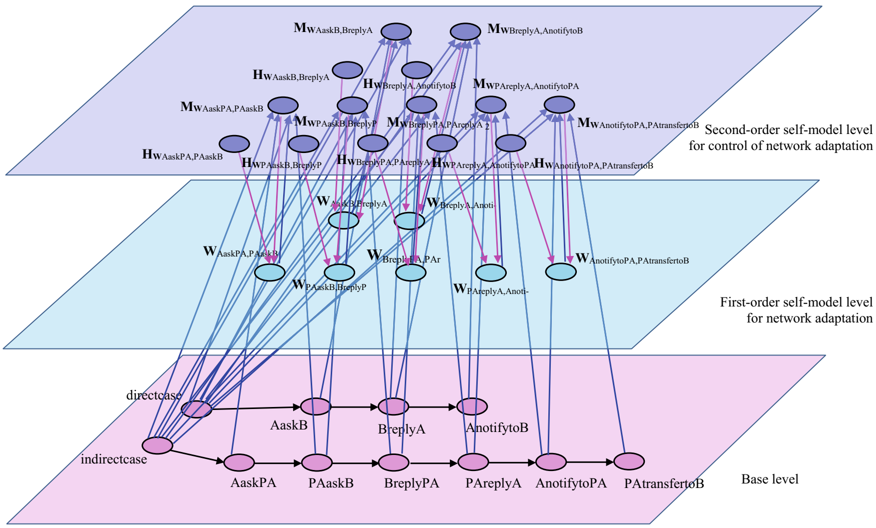

Fig. 4.6 Graphical representation of the connectivity of the second-order adaptive mental network model for metacognitive control of focusing on a mental model. View on the interactions of the second-order self-model level $\mathbf{M_W}$-states

- The *composed max-hebbian learning combination function* is defined by

$$\mathbf{maxhebb}_{\boldsymbol{\mu}}(V_1, V_2, V_3, V_4, V_5) = \max(V_1 * V_2(1 - V_3) + \boldsymbol{\mu} V_3, \max(V_4, V_5)) \tag{4.3}$$

where $\boldsymbol{\mu}$ is a persistence parameter, V_3 represents the weight of the connection, and V_1, V_2 are the activation levels of the connected states.

- The $\mathbf{steponce}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$**(..)** function is 1 between time points $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and 0 else.

The full specification can be found in the Appendix Sect. 4.7; see also the Linked Data at URL https://www.researchgate.net/publication/353667091.

4.6 Example Simulation Scenario

The introduced second-order self-modeling network model is illustrated for an example case concerning indirect and direct communication scenarios, as described in Sect. 4.5. In this section, a simulation example of the adaptive mental network model is discussed displaying the learning process of people trying to communicate with each other. Not so surprisingly, direct communication is much faster than indirect communication, so when that is an option, it is not efficient to apply a mental model for indirection communication.

The designed network model can be characterized in terms of connectivity, aggregation and timing. Two matrices for connectivity, **mb** for base connectivity and **mcw** for connection weights characterize the connections of the network model. Two matrices for aggregation, **mcfw** for weights of combination functions and **mcfp** for parameters of the combination functions specify aggregation. And finally, one matrix **ms** for speed factors characterizes the timing. The full specification of these matrices can be found in Appendix Sect. 4.8.

As briefly indicated in Sect. 4.3 and 4.4, the self-modeling feature of the modeling method adds higher-order levels above the base level. In our model, the base level includes the mental models for interaction states between people and in addition two context states for switching between the mental models for indirect and direct cases, beginning from X_1 to X_{11} inclusive.

In the base level, for the indirect case, there is a mental model of 6 states that can be seen as the steps of the communication (big blue oval in the base level in Fig. 4.3), and for the mental model for the direct case there are 3 (big purple oval in the base level in Fig. 4.3). It indeed shows that indirect communication is a longer process.

At the first-order self-model level, there are 7 **W**-states (in the big blue and purple ovals in the middle level plane in Fig. 4.3) for all connections of the two mental models shown at the base level both for the direct and indirect cases. This level makes the model adaptive in terms of the connections between base level states. For these **W**-states Hebbian learning is used to provide learning (and forgetting), and the **W**-states are changed to obtain focusing on them.

At the second-order self-model level, $\mathbf{H_W}$-states and $\mathbf{M_W}$-states are used to control the mentioned forms of adaptation itself in the first-order self-model level. In our model, there are 7 $\mathbf{H_W}$-states and 7 $\mathbf{M_W}$-states (in the big blue and purple ovals in the upper-level plane in Fig. 4.3) for all of the 7 **W**-states from the middle level plane. The simulation outcomes depicted in the overall graphs in Fig. 4.7 and the partial graphs in Figs. 4.8, 4.9 and 4.10 show how the process runs. In each of the periods from 25 to 125 and from 200 to 300, a different mental model comes in focus. In both cases the mental model is not perfect as its connection weights are only 0.8. However, by using the mental model for internal mental simulation, Hebbian learning (Hebb 1949) takes place by which the weights increase to 1 or close to it. The first context state finishes at time 125 and the next, other context state starts at time 200. In the meantime, the weights of the mental model that was first in focus drop to 0 (so, it is not used anymore) and instead, after time 200 the other mental model relating to the new context gets high weights and is used then.

The **steponce** function was used in our model to do the indirect-direct context case shift. In Figs. 4.7, 4.8, 4.9 and 4.10 the time interval 25–125 demonstrates the first case (indirect communication) and the time interval 200–300 demonstrates the second one (direct communication). As it can be seen, base states become higher in their level after the increase of the **W**-states because of the learning that takes place. Each different **W**-state affects its corresponding base states, thus, also in harmony with real life, within a mental model the represented communication steps do not trace the same route, they follow each other with a small time difference. Learning is a process; it does not happen all of a sudden.

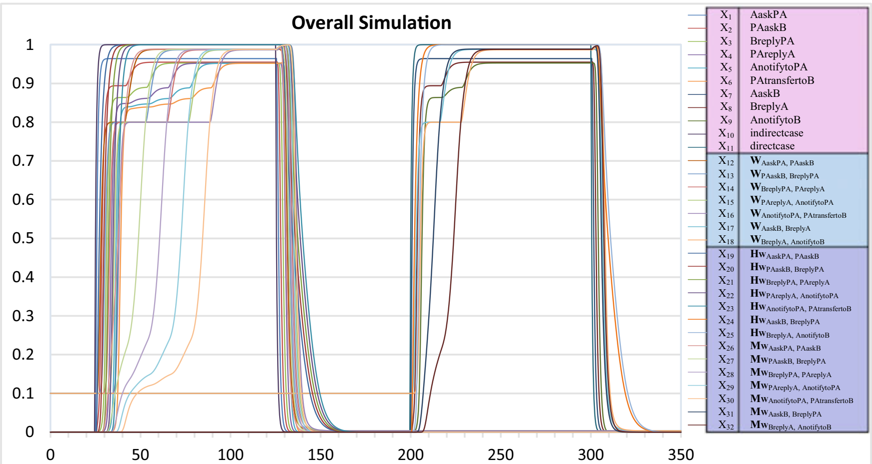

Fig. 4.7 Overall picture of all states in the simulation

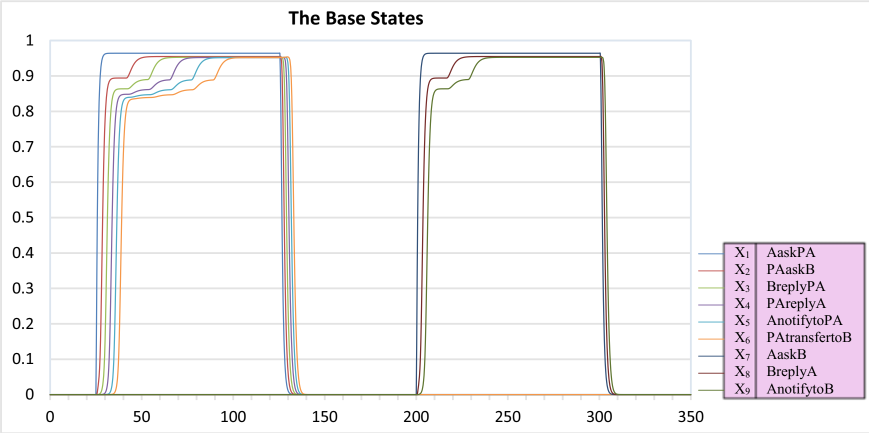

Fig. 4.8 Picture for the base states in the simulation

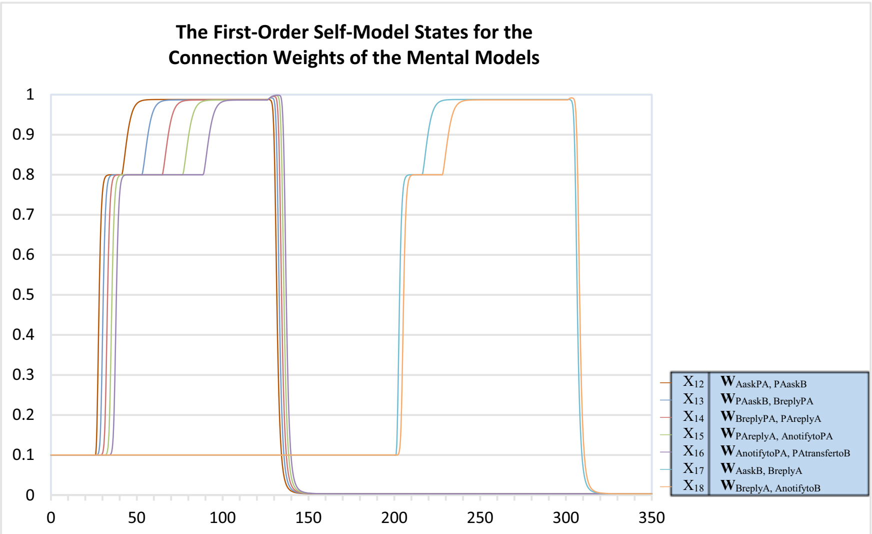

Fig. 4.9 Picture for the first-order self-model states representing the connection weights of the mental models in the simulation

The learning process of indirect communication only takes place in the first time interval (25–125) with the help of the context states based on the **steponce** function. All the base and adaptation states of the indirect case get increased, but between time 125 and time 150, all become 0 because the context state of this case becomes 0 after time 125. During the gap between the cases, until the second case context state becomes nonzero, everything is zero except the **W**-states of the second case. A minimal nonzero weight is used for the initialization of the learning of direct

Fig. 4.10 Picture for the second-order self-model states in the simulation exerting context-sensitive control both over the focusing on a mental model and the learning of it

communication. Then the second case starts and the learning process of this direct communication only happens in the second time interval from time 200 to about 300. Similar to the first case, adaptation states such as **W**-, $\mathbf{H_W}$- and $\mathbf{M_W}$-states increase hand in hand with the base states and that corresponds to the learning process.

4.7 Discussion

Part of the content of this chapter is based on Canbaloğlu and Treur (2021). Within mental processes, often mental models are used for internal simulation to obtain predictions. These mental models are also often adapted. Adaptation processes can be described by adaptive network models. If metacognition is used to regulate adaptation, for example during learning (Pintrich 2000), the adaptation becomes itself adaptive as well, so then it involves second-order adaptation.

In this chapter, a second-order adaptive network model was introduced for metacognitive control of adaptation processes needed for learning of a specific mental model and using this mental model. It was shown how a second-order self-modeling network model provides adequate means to model the different aspects that make the addressed topic complex: the network has a self-model about its own structure, it models mental models and adaptation of them, and it models context-sensitive metacognitive control of this adaptation. The model was applied to simulate an illustrative example scenario that explains what the model does. In further work other scenarios have been addressed as well, in particular for the use of mental models in teams in hospitals, building further, for example, on Van Ments et al. (2021) and Treur and Van Ments (2022).

4.8 Appendix: Full Specification by Role Matrices

In Figs. 4.11, 4.12, 4.13, 4.14 and 4.15, the different role matrices are shown that provide a full specification of the network characteristics defining the network model in a standardised table format. Here in each role matrix, each state has its row where it is listed which are the impacts on it from that role.

Fig. 4.11 Role matrices for the connectivity: **mb** for base connectivity

	mb **base connectivity**	1	2	3	4	5	6
X_1	AaskPA	X_{10}					
X_2	PAaskB	X_1					
X_3	BreplyPA	X_2					
X_4	PAreplyA	X_3					
X_5	AnotifytoPA	X_4					
X_6	PAtransfertoB	X_5					
X_7	AaskB	X_{11}					
X_8	BreplyA	X_7					
X_9	AnotifytoB	X_8					
X_{10}	indirectcase	X_{10}					
X_{11}	directcase	X_{11}					
X_{12}	$\mathbf{W}_{AaskPA,PAaskB}$	X_1	X_2	X_{12}	X_{10}	X_{11}	
X_{13}	$\mathbf{W}_{PAaskB,BreplyPA}$	X_2	X_3	X_{13}	X_{10}	X_{11}	
X_{14}	$\mathbf{W}_{BreplyPA,PAreplyA}$	X_3	X_4	X_{14}	X_{10}	X_{11}	
X_{15}	$\mathbf{W}_{PAreplyA,AnotifytoPA}$	X_4	X_5	X_{15}	X_{10}	X_{11}	
X_{16}	$\mathbf{W}_{AnotifytoPA,PAtransfertoB}$	X_5	X_6	X_{16}	X_{10}	X_{11}	
X_{17}	$\mathbf{W}_{AaskB,BreplyA}$	X_7	X_8	X_{17}	X_{10}	X_{11}	
X_{18}	$\mathbf{W}_{BreplyA,AnotifytoB}$	X_8	X_9	X_{18}	X_{10}	X_{11}	
X_{19}	$\mathbf{H}_{\mathbf{W}AaskPA,PAaskB}$	X_1	X_2	X_{12}	X_{19}		
X_{20}	$\mathbf{H}_{\mathbf{W}PAaskB,BreplyPA}$	X_2	X_3	X_{13}	X_{20}		
X_{21}	$\mathbf{H}_{\mathbf{W}BreplyPA,PAreplyA}$	X_3	X_4	X_{14}	X_{21}		
X_{22}	$\mathbf{H}_{\mathbf{W}PAreplyA,AnotifytoPA}$	X_4	X_5	X_{15}	X_{22}		
X_{23}	$\mathbf{H}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	X_5	X_6	X_{16}	X_{23}		
X_{24}	$\mathbf{H}_{\mathbf{W}AaskB,BreplyPA}$	X_7	X_8	X_{17}	X_{24}		
X_{25}	$\mathbf{H}_{\mathbf{W}BreplyA,AnotifytoB}$	X_8	X_9	X_{18}	X_{25}		
X_{26}	$\mathbf{M}_{\mathbf{W}AaskPA,PAaskB}$	X_1	X_2	X_{12}	X_{26}	X_{10}	X_{11}
X_{27}	$\mathbf{M}_{\mathbf{W}PAaskB,BreplyPA}$	X_2	X_3	X_{13}	X_{27}	X_{10}	X_{11}
X_{28}	$\mathbf{M}_{\mathbf{W}BreplyPA,PAreplyA}$	X_3	X_4	X_{14}	X_{28}	X_{10}	X_{11}
X_{29}	$\mathbf{M}_{\mathbf{W}PAreplyA,AnotifytoPA}$	X_4	X_5	X_{15}	X_{29}	X_{10}	X_{11}
X_{30}	$\mathbf{M}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	X_5	X_6	X_{16}	X_{30}	X_{10}	X_{11}
X_{31}	$\mathbf{M}_{\mathbf{W}AaskB,BreplyPA}$	X_7	X_8	X_{17}	X_{31}	X_{10}	X_{11}
X_{32}	$\mathbf{M}_{\mathbf{W}BreplyA,AnotifytoB}$	X_8	X_9	X_{18}	X_{32}	X_{10}	X_{11}

4.8.1 Role Matrices for Connectivity Characteristics

The connectivity characteristics are specified by role matrices **mb** and **mcw** shown in Figs. 4.11 and 4.12. Role matrix **mb** lists the other states (at the same or lower level) from which the state gets its incoming connections, whereas in role matrix **mcw** the connection weights are listed for these connections.

Nonadaptive connection weights are indicated in **mcw** by a number (in a green shaded cell), but adaptive connection weights are indicated by a reference to the (self-model) state representing the adaptive value (in a peach-red shaded cell). This can be seen for states X_2 to X_6 (with self-model states X_{12} to X_{16}) and states X_8 and X_9 (with self-model states X_{17} and X_{18}).

Fig. 4.12 Role matrices for the connectivity: **mcw** for connection weights

mcw	**connection weights**	1	2	3	4	5	6
X_1	AaskPA	1					
X_2	PAaskB	X_{12}					
X_3	BreplyPA	X_{13}					
X_4	PAreplyA	X_{14}					
X_5	AnotifytoPA	X_{15}					
X_6	PAtransfertoB	X_{16}					
X_7	AaskB	1					
X_8	BreplyA	X_{17}					
X_9	AnotifytoB	X_{18}					
X_{10}	indirectcase	1					
X_{11}	directcase	1					
X_{12}	$\mathbf{W}_{AaskPA,PAaskB}$	1	1	1	0.8	0	
X_{13}	$\mathbf{W}_{PAaskB,BreplyPA}$	1	1	1	0.8	0	
X_{14}	$\mathbf{W}_{BreplyPA,PAreplyA}$	1	1	1	0.8	0	
X_{15}	$\mathbf{W}_{PAreplyA,AnotifytoPA}$	1	1	1	0.8	0	
X_{16}	$\mathbf{W}_{AnotifytoPA,PAtransfertoB}$	1	1	1	0.8	0	
X_{17}	$\mathbf{W}_{AaskB,BreplyA}$	1	1	1	0	0.8	
X_{18}	$\mathbf{W}_{BreplyA,AnotifytoB}$	1	1	1	0	0.8	
X_{19}	$\mathbf{H}_{\mathbf{W}AaskPA,PAaskB}$	1	1	-0.1	0.6		
X_{20}	$\mathbf{H}_{\mathbf{W}PAaskB,BreplyPA}$	1	1	-0.1	0.6		
X_{21}	$\mathbf{H}_{\mathbf{W}BreplyPA,PAreplyA}$	1	1	-0.1	0.6		
X_{22}	$\mathbf{H}_{\mathbf{W}PAreplyA,AnotifytoPA}$	1	1	-0.1	0.6		
X_{23}	$\mathbf{H}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	1	1	-0.1	0.6		
X_{24}	$\mathbf{H}_{\mathbf{W}AaskB,BreplyPA}$	1	1	-0.1	0.6		
X_{25}	$\mathbf{H}_{\mathbf{W}BreplyA,AnotifytoB}$	1	1	-0.1	0.6		
X_{26}	$\mathbf{M}_{\mathbf{W}AaskPA,PAaskB}$	1	1	1	1	-1	-1
X_{27}	$\mathbf{M}_{\mathbf{W}PAaskB,BreplyPA}$	1	1	1	1	-1	-1
X_{28}	$\mathbf{M}_{\mathbf{W}BreplyPA,PAreplyA}$	1	1	1	1	-1	-1
X_{29}	$\mathbf{M}_{\mathbf{W}PAreplyA,AnotifytoPA}$	1	1	1	1	-1	-1
X_{30}	$\mathbf{M}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	1	1	1	1	-1	-1
X_{31}	$\mathbf{M}_{\mathbf{W}AaskB,BreplyPA}$	1	1	1	1	-1	-1
X_{32}	$\mathbf{M}_{\mathbf{W}BreplyA,AnotifytoB}$	1	1	1	1	-1	-1

4.8.2 *Role Matrices for Aggregation Characteristics*

The network characteristics for aggregation are defined by the selection of combination functions from the library and values for their parameters. In role matrix **mcfw** it is specified by weights which state uses which combination function; see Fig. 4.13.

In role matrix **mcfp** (see Fig. 4.14) it is indicated what the parameter values are for the chosen combination functions. Some of them are adaptive, as can be seen in the rows from X_{12} to X_{18} (see the indications for the persistence factors **μ** represented by the self-model states X_{26} to X_{32}).

Fig. 4.13 Role matrices for the aggregation characteristics: combination function weights

mcfw	**combination function weights**	1 **alo-gistic**	2 **step-once**	3 **max-hebb**
X_1	AaskPA	1		
X_2	PAaskB	1		
X_3	BreplyPA	1		
X_4	PAreplyA	1		
X_5	AnotifytoPA	1		
X_6	PAtransfertoB	1		
X_7	AaskB	1		
X_8	BreplyA	1		
X_9	AnotifytoB	1		
X_{10}	indirectcase		1	
X_{11}	directcase		1	
X_{12}	$\mathbf{W}_{AaskPA,PAaskB}$			1
X_{13}	$\mathbf{W}_{PAaskB,BreplyPA}$			1
X_{14}	$\mathbf{W}_{BreplyPA,PAreplyA}$			1
X_{15}	$\mathbf{W}_{PAreplyA,AnotifytoPA}$			1
X_{16}	$\mathbf{W}_{AnotifytoPA,PAtransfertoB}$			1
X_{17}	$\mathbf{W}_{AaskB,BreplyA}$			1
X_{18}	$\mathbf{W}_{BreplyA,AnotifytoB}$			1
X_{19}	$\mathbf{H}_{\mathbf{W}AaskPA,PAaskB}$	1		
X_{20}	$\mathbf{H}_{\mathbf{W}PAaskB,BreplyPA}$	1		
X_{21}	$\mathbf{H}_{\mathbf{W}BreplyPA,PAreplyA}$	1		
X_{22}	$\mathbf{H}_{\mathbf{W}PAreplyA,AnotifytoPA}$	1		
X_{23}	$\mathbf{H}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	1		
X_{24}	$\mathbf{H}_{\mathbf{W}AaskB,BreplyPA}$	1		
X_{25}	$\mathbf{H}_{\mathbf{W}BreplyA,AnotifytoB}$	1		
X_{26}	$\mathbf{M}_{\mathbf{W}AaskPA,PAaskB}$	1		
X_{27}	$\mathbf{M}_{\mathbf{W}PAaskB,BreplyPA}$	1		
X_{28}	$\mathbf{M}_{\mathbf{W}BreplyPA,PAreplyA}$	1		
X_{29}	$\mathbf{M}_{\mathbf{W}PAreplyA,AnotifytoPA}$	1		
X_{30}	$\mathbf{M}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	1		
X_{31}	$\mathbf{M}_{\mathbf{W}AaskB,BreplyPA}$	1		
X_{32}	$\mathbf{M}_{\mathbf{W}BreplyA,AnotifytoB}$	1		

Fig. 4.14 Role matrices for the aggregation characteristics: combination function parameters

mcfp	combination function parameters	1 alogistic		2 step-once		3 max-hebb	
		1	2	1	2	1	2
		σ	τ	α	β	μ	
X_1	AaskPA	5	0.3				
X_2	PAaskB	5	0.3				
X_3	BreplyPA	5	0.3				
X_4	PAreplyA	5	0.3				
X_5	AnotifytoPA	5	0.3				
X_6	PAtransfertoB	5	0.3				
X_7	AaskB	5	0.3				
X_8	BreplyA	5	0.3				
X_9	AnotifytoB	5	0.3				
X_{10}	indirectcase			25	125		
X_{11}	directcase			200	300		
X_{12}	$\mathbf{W}_{AaskPA,PAaskB}$					X_{26}	
X_{13}	$\mathbf{W}_{PAaskB,BreplyPA}$					X_{27}	
X_{14}	$\mathbf{W}_{BreplyPA,PAreplyA}$					X_{28}	
X_{15}	$\mathbf{W}_{PAreplyA,AnotifytoPA}$					X_{29}	
X_{16}	$\mathbf{W}_{AnotifytoPA,PAtransfertoB}$					X_{30}	
X_{17}	$\mathbf{W}_{AaskB,BreplyA}$					X_{31}	
X_{18}	$\mathbf{W}_{BreplyA,AnotifytoB}$					X_{32}	
X_{19}	$\mathbf{H}_{\mathbf{W}AaskPA,PAaskB}$	5	0.3				
X_{20}	$\mathbf{H}_{\mathbf{W}PAaskB,BreplyPA}$	5	0.3				
X_{21}	$\mathbf{H}_{\mathbf{W}BreplyPA,PAreplyA}$	5	0.3				
X_{22}	$\mathbf{H}_{\mathbf{W}PAreplyA,AnotifytoPA}$	5	0.3				
X_{23}	$\mathbf{H}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	5	0.3				
X_{24}	$\mathbf{H}_{\mathbf{W}AaskB,BreplyPA}$	5	0.3				
X_{25}	$\mathbf{H}_{\mathbf{W}BreplyA,AnotifytoB}$	5	0.3				
X_{26}	$\mathbf{M}_{\mathbf{W}AaskPA,PAaskB}$	5	2				
X_{27}	$\mathbf{M}_{\mathbf{W}PAaskB,BreplyPA}$	5	2				
X_{28}	$\mathbf{M}_{\mathbf{W}BreplyPA,PAreplyA}$	5	2				
X_{29}	$\mathbf{M}_{\mathbf{W}PAreplyA,AnotifytoPA}$	5	2				
X_{30}	$\mathbf{M}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	5	2				
X_{31}	$\mathbf{M}_{\mathbf{W}AaskB,BreplyPA}$	5	2				
X_{32}	$\mathbf{M}_{\mathbf{W}BreplyA,AnotifytoB}$	5	2				

4.8.3 Role Matrices for Timing Characteristics

In Fig. 4.15, the role matrix **ms** for speed factors is shown, which lists all speed factors. Next to it the list of initial values can be found. Also here, a few them are adaptive: the speed factors of X_{12} to X_{18} are represented by self-model states X_{19} to X_{25}. In Fig. 4.15, also the initial values of all states used in the example simulation are shown.

Fig. 4.15 Role matrices for the timing characteristics and initial values: **ms** for speed factors

ms	speed factors	1
X_1	AaskPA	1
X_2	PAaskB	1
X_3	BreplyPA	1
X_4	PAreplyA	1
X_5	AnotifytoPA	1
X_6	PAtransfertoB	1
X_7	AaskB	1
X_8	BreplyA	1
X_9	AnotifytoB	1
X_{10}	indirectcase	1
X_{11}	directcase	1
X_{12}	$\mathbf{W}_{AaskPA,PAaskB}$	X_{19}
X_{13}	$\mathbf{W}_{PAaskB,BreplyPA}$	X_{20}
X_{14}	$\mathbf{W}_{BreplyPA,PAreplyA}$	X_{21}
X_{15}	$\mathbf{W}_{PAreplyA,AnotifytoPA}$	X_{22}
X_{16}	$\mathbf{W}_{AnotifytoPA,PAtransfertoB}$	X_{23}
X_{17}	$\mathbf{W}_{AaskB,BreplyA}$	X_{24}
X_{18}	$\mathbf{W}_{BreplyA,AnotifytoB}$	X_{25}
X_{19}	$\mathbf{H}_{\mathbf{W}AaskPA,PAaskB}$	0.5
X_{20}	$\mathbf{H}_{\mathbf{W}PAaskB,BreplyPA}$	0.5
X_{21}	$\mathbf{H}_{\mathbf{W}BreplyPA,PAreplyA}$	0.5
X_{22}	$\mathbf{H}_{\mathbf{W}PAreplyA,AnotifytoPA}$	0.5
X_{23}	$\mathbf{H}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	0.5
X_{24}	$\mathbf{H}_{\mathbf{W}AaskB,BreplyPA}$	0.5
X_{25}	$\mathbf{H}_{\mathbf{W}BreplyA,AnotifytoB}$	0.5
X_{26}	$\mathbf{M}_{\mathbf{W}AaskPA,PAaskB}$	0.5
X_{27}	$\mathbf{M}_{\mathbf{W}PAaskB,BreplyPA}$	0.5
X_{28}	$\mathbf{M}_{\mathbf{W}BreplyPA,PAreplyA}$	0.5
X_{29}	$\mathbf{M}_{\mathbf{W}PAreplyA,AnotifytoPA}$	0.5
X_{30}	$\mathbf{M}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	0.5
X_{31}	$\mathbf{M}_{\mathbf{W}AaskB,BreplyPA}$	0.5
X_{32}	$\mathbf{M}_{\mathbf{W}BreplyA,AnotifytoB}$	0.5

iv	initial values	1
X_1	AaskPA	0
X_2	PAaskB	0
X_3	BreplyPA	0
X_4	PAreplyA	0
X_5	AnotifytoPA	0
X_6	PAtransfertoB	0
X_7	AaskB	0
X_8	BreplyA	0
X_9	AnotifytoB	0
X_{10}	indirectcase	0
X_{11}	directcase	0
X_{12}	$\mathbf{W}_{AaskPA,PAaskB}$	0.1
X_{13}	$\mathbf{W}_{PAaskB,BreplyPA}$	0.1
X_{14}	$\mathbf{W}_{BreplyPA,PAreplyA}$	0.1
X_{15}	$\mathbf{W}_{PAreplyA,AnotifytoPA}$	0.1
X_{16}	$\mathbf{W}_{AnotifytoPA,PAtransfertoB}$	0.1
X_{17}	$\mathbf{W}_{AaskB,BreplyA}$	0.1
X_{18}	$\mathbf{W}_{BreplyA,AnotifytoB}$	0.1
X_{19}	$\mathbf{H}_{\mathbf{W}AaskPA,PAaskB}$	0
X_{20}	$\mathbf{H}_{\mathbf{W}PAaskB,BreplyPA}$	0
X_{21}	$\mathbf{H}_{\mathbf{W}BreplyPA,PAreplyA}$	0
X_{22}	$\mathbf{H}_{\mathbf{W}PAreplyA,AnotifytoPA}$	0
X_{23}	$\mathbf{H}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	0
X_{24}	$\mathbf{H}_{\mathbf{W}AaskB,BreplyPA}$	0
X_{25}	$\mathbf{H}_{\mathbf{W}BreplyA,AnotifytoB}$	0
X_{26}	$\mathbf{M}_{\mathbf{W}AaskPA,PAaskB}$	0
X_{27}	$\mathbf{M}_{\mathbf{W}PAaskB,BreplyPA}$	0
X_{28}	$\mathbf{M}_{\mathbf{W}BreplyPA,PAreplyA}$	0
X_{29}	$\mathbf{M}_{\mathbf{W}PAreplyA,AnotifytoPA}$	0
X_{30}	$\mathbf{M}_{\mathbf{W}AnotifytoPA,PAtransfertoB}$	0
X_{31}	$\mathbf{M}_{\mathbf{W}AaskB,BreplyPA}$	0
X_{32}	$\mathbf{M}_{\mathbf{W}BreplyA,AnotifytoB}$	0

References

Abraham, W.C., Bear, M.F.: Metaplasticity: the plasticity of synaptic plasticity. Trends Neurosci. **19**(4), 126–130 (1996)

Bhalwankar, R., Treur, J.: Modeling learner-controlled mental model learning processes by a second-order adaptive network model. PLoS ONE 16(8): e0255503 (2021)

Canbaloğlu, G., Treur, J.: Modeling context-sensitive metacognitive control of focusing on a mental model during a mental process. In: Silhavy, R., Silhavy, P., Prokopova, Z. (eds.) Data Science and Intelligent Systems. Proceedings of CoMeSySo 2021. Lecture Notes in Networks and Systems, vol. 231, pp. 992–1009. Springer Nature (2021). https://www.researchgate.net/publication/353667091

Canbaloğlu, G., Treur, J., Wiewiora, A. (eds.).: Computational Modeling of Multilevel Organizational Learning and its Control Using Self-Modeling Network Models. Springer Nature (2023) (this volume)

Craik, K.J.W.: The nature of explanation. Cambridge, MA: University Press. (1943).

Darling-Hammond, L., Austin, K., Cheung, M., Martin, D.: Thinking about Thinking: Metacognition (2008)

Flavell, J.H.: Metacognition and cognitive monitoring: a new area of cognitive–developmental inquiry. Am. Psychol. **34**(10), 906–911 (1979)

Gentner, D., Stevens, A.L.: Mental Models. Erlbaum, Hillsdale NJ (1983)

Greca, I.M., Moreira, M.A.: Mental models, conceptual models, and modelling. Int. J. Sci. Educ. **22**(1), 1–11 (2000)

Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley (1949)

Koriat, A.: Metacognition and consciousness. In: Zelavo, P.D., Moscovitch, M., Thompson, E. (eds.) Cambridge Handbook of Consciousness. Cambridge University Press, New York (2007)
Magerl, W., Hansen, N., Treede, R.D., Klein, T.: The human pain system exhibits higher-order plasticity (metaplasticity). Neurobiol. Learn. Mem. **154**, 112–120 (2018)
Mahdavi, M.: An overview: metacognition in education. Int. J. Multidisc. Curr. Res. **2**, 529–535 (2014)
Pintrich, P.R.: The role of goal orientation in self-regulated learning. In: Boekaerts, M., Pintrich, P., Zeidner, M. (eds.) Handbook of Self-Regulation Research and Applications, pp. 451–502. Academic Press, Orlando, FL (2000)
Robinson, B.L., Harper, N.S., McAlpine, D.: Meta-adaptation in the auditory midbrain under cortical influence. Nat. Commun. **7**, 13442 (2016)
Seel, N.M.: Mental models in learning situations. In: Advances in Psychology, vol. 138, pp. 85–107. North-Holland, Amsterdam (2006)
Shannon, S.V.: Using metacognitive strategies and learning styles to create self-directed learners. Inst. Learn. Styles J. **1**, 14–28 (2008)
Shih, Y.F. Alessi, S.M.: Mental models and transfer of learning in computer programming. Journal of Research in Computing Education, **26**(2), 154–175 (1993)
Sjöström, P.J., Rancz, E.A., Roth, A., Hausser, M.: Dendritic excitability and synaptic plasticity. Physiol. Rev. **88**(769–840), 2008 (2008)
Skemp, R.R.: The Psychology of Learning Mathematics. Penguin Books, Harmondsworth (1971)
Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive. Springer Publishers, Affective and Social Interactions (2016)
Treur, J.: Multilevel network reification: representing higher order adaptivity in a network. In: Aiello, L., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L. (eds.) Proceedings of the 7th International Conference on Complex Networks and their Applications, Complex Networks'18, vol. 1. Studies in Computational Intelligence, vol. 812, pp. 635–651, Springer Nature (2018)
Treur, J.: Modeling higher-order adaptivity of a network by multilevel network reification. Netw. Sci. **8**, S110–S144 (2020a)
Treur, J.: Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models. Springer Nature Publishing, Cham, Switzerland (2020b)
Treur, J.: An adaptive network model covering metacognition to control adaptation for multiple mental models. Cogn. Syst. Res. **67**, 18–27 (2021)
Treur, J., Van Ments, L. (eds.): Mental Models and their Dynamics, Adaptation and Control: A Self-Modeling Network Modeling Approach. Springer Nature, Cham, Switzerland (2022)
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 **21**(1), 21-30 (2009)
Van Ments, L., Treur, J.: Reflections on dynamics, adaptation and control: a cognitive architecture for mental models. Cogn. Syst. Res. **70**, 1–9 (2021)
Van Ments, L., Treur, J., Klein, J., Roelofsma, P.H.M.P.: A second-order adaptive network model for shared mental models in hospital teamwork. In: Nguyen, N.T., et al. (eds.) Proceedings of the 13th International Conference on Computational Collective Intelligence, ICCCI'21. Lecture Notes in AI, vol 12876, pp 126–140. Springer Nature (2021)
Yi, M.Y., Davis, F.D.: Developing and validating an observational learning model of computer software training and skill acquisition. Information Systems Research **14**(2), 146–169 (2003)