VU Research Portal

Bringing Networks to the Next Level

Treur, Jan

published in
Mental Models and their Dynamics, Adaptation and Control
2022

DOI (link to publisher)
10.1007/978-3-030-85821-6_2

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

document license
Article 25fa Dutch Copyright Act

Link to publication in VU Research Portal

citation for published version (APA)
Treur, J. (2022). Bringing Networks to the Next Level: Self-modeling Networks for Adaptivity and Control of Mental Models. In J. Treur, & L. Van Ments (Eds.), *Mental Models and their Dynamics, Adaptation and Control: A Self-Modeling Network Modeling Approach* (pp. 27-55). (Studies in Systems, Decision and Control; Vol. 394). Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-85821-6_2

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 02. Nov. 2024

Chapter 2
Bringing Networks to the Next Level: Self-modeling Networks for Adaptivity and Control of Mental Models

Jan Treur

Abstract Networks provide an intuitive, declarative way of modeling with a wide scope of applicability. In many cases also adaptivity of a network plays a role, which easily leads to less declarative and transparent forms of modeling by using algorithmic or procedural descriptions for the adaptation processes. This chapter addresses this by exploiting the notion of self-modeling network that has been developed recently. Using that, adaptivity is obtained by adding a self-model to a given base network, with states that represent part of the base network's structure. This adds a next level to the base network, resulting in a two-level network. This construction can easily be iterated to obtain more levels so that multiple orders of adaptation can be covered as well. This brings networks to a next level in more than one way. In particular, a three-level self-modeling network can be used to integrate dynamics, adaptivity and control in one network. It is shown how this can be used to design network models for mental model handling.

Keywords Network-oriented modeling · Self-modeling network · Network reification · Adaptive network model · Controlled adaptation

2.1 Introduction

Recently, within network science new modeling approaches have been developed that can be used to model networks with not only the dynamics of their nodes or states but also the adaptation of their structure and the control of that adaptation. In particular, in this chapter it will be addressed how both within-network dynamics (dynamics of the node states) for networks and adaptivity of the network structure can be addressed using self-modeling networks (Treur 2020a, b).

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

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
J. Treur and L. Van Ments (eds.), *Mental Models and Their Dynamics, Adaptation, and Control*, Studies in Systems, Decision and Control 394,
https://doi.org/10.1007/978-3-030-85821-6_2

Self-modeling networks are networks that include a self-model for part of their own network structure in the form of nodes that represent certain network structure characteristics such as connection weights or excitability thresholds. Any (base) network can be extended by including such a self-model, which can be considered to be at a next level, compared to the base network; this self-modeling step is also called network reification. This construction for networks in particular relates to a long-standing tradition in other areas of AI, namely that of meta-programming and metalevel architectures; e.g., Bowen and Kowalski (1982) Demers and Malenfant (1995), Sterling and Shapiro (1996), Sterling and Beer (1989), Weyhrauch (1980). Having such self-models within a network enables to model adaptation of the network structure by the within-network dynamics of the self-model representing this network structure. As the latter can be specified by declarative means in the form of mathematical relations and functions, also adaptivity of the network structure can be specified in a similar declarative manner. To support the modeler, a dedicated software environment is available that also applies to self-modeling networks; see (Treur 2020b, Chap. 9).

In this chapter, the perspective pointed out above will be illustrated in more detail. First in Sect. 2.2 the network-oriented modeling approach based on self-modeling networks will be briefly introduced. In Sect. 2.3 it is discussed how various adaptation principles from the neuroscientific literature can be modeled. Next, in Sect. 2.4 it will be illustrated by an example of a second-order adaptive network model for emotion regulation dysfunction. In Sect. 2.5 it is shown how a self-modeling network model can be used to obtain a computational network model for mental model handling according to the three-level cognitive architecture described in Van Ments and Treur (2021); see also Van Ments and Treur (2022). Finally, Sect. 2.6 is a discussion.

2.2 Modeling Adaptivity by Self-modeling Networks

In this section, the network-oriented modeling approach by self-modeling networks used is briefly introduced in two steps.

2.2.1 Network-Oriented Modeling

As in this approach *nodes* Y in a network have activation values $Y(t)$ that are dynamic over time t, they serve as state variables and will usually be simply called *states*. For these dynamics, the states are considered to affect each other by the connections within the network. Following Treur (2016, 2020b), a basic *network structure* is characterised by:

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

- **Aggregation characteristics**
  For any node Y, some combination function $\mathbf{c}_Y(..)$ defines aggregation that is applied to the single impacts $\boldsymbol{\omega}_{X_i,Y} X_i(t)$ on Y from its incoming connections from states $X_1, \ldots, X_k$
- **Timing characteristics**
  Each state Y has a *speed factor* $\boldsymbol{\eta}_Y$ defining how fast it changes upon given impact.

Here, the states X_i and Y have activation levels $X_i(t)$ and $Y(t)$ that vary (often within the [0, 1] interval) over time, described by real numbers t. The dynamics of such networks are described by the following difference (or differential) equations that incorporate in a canonical manner the network characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$:

$$Y(t + \Delta t) = Y(t) + \boldsymbol{\eta}_Y\left[\mathbf{c}_\mathbf{Y}(\boldsymbol{\omega}_{X_1,Y} X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y} X_k(t)) - Y(t)\right]\Delta t \qquad (2.1)$$

for any state Y and where $X_1, \ldots, X_k$ are the states from which Y gets its incoming connections. The Eq. (2.1) are useful for simulation purposes and also for analysis of properties of the emerging behaviour of such network models. The overall combination function $\mathbf{c}_Y(..)$ for state Y is taken as the weighted average of some of the available basic combination functions $\mathbf{c}_j(..)$ 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:

$$\mathbf{c}_Y(V_1, \cdots, V_k) = \frac{\boldsymbol{\gamma}_{1,Y}\mathbf{c}_1(V_1, \ldots, V_k) + \ldots + \boldsymbol{\gamma}_{m,Y}\mathbf{c}_m(V_1, \ldots, V_k)}{\boldsymbol{\gamma}_{1,Y} + \ldots + \boldsymbol{\gamma}_{m,Y}} \qquad (2.2)$$

Such Eqs. (2.1), (2.2) are hidden in the dedicated software environment that can be used for simulation and analysis; see Treur (2020b, Chap. 9). This software environment is freely downloadable from URL.

https://www.researchgate.net/project/Network-Oriented-Modeling-Software.

Combination functions are similar to the functions used in a static manner in the deterministic Structural Causal Model perspective described, for example, in Pearl (2000), Wright (1921), Mooi et al. (2013). However, in the Network-Oriented Modelling approach described here they are used in a dynamic manner. For example, Pearl (2000, p. 203), denotes nodes by V_i and combination functions by f_i (although he uses a different term for these functions). In the following quote he points at the issue of underspecification concerning aggregation of multiple connections, as in the often used graph representations the specification of combination functions f_i for nodes V_i, is lacking:

> Every causal model M can be associated with a directed graph (…) This graph merely identifies the endogenous and background variables that have a direct influence on each V_i; it does not specify the functional form of f_i. (Pearl 2000, p. 203)

Therefore, in addition to graph representations for connectivity, at least aggregation in terms of combination functions has to be addressed, as indeed is done for the

Table 2.1 Examples of basic combination functions from the library

Name	Notation	Formula	Parameters
Identity	$\mathbf{id}(V)$	V	–
Scaled sum	$\mathbf{ssum}_{\lambda}(V_1, \ldots, V_k)$	$\frac{V_1+\cdots+V_k}{\lambda}$	Scaling factor λ
Euclidean	$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k)$	$\sqrt[n]{\frac{V_1^n+\cdots+V_k^n}{\lambda}}$	Order n Scaling factor λ
Advanced logistic	$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k)$	$\left[\frac{1}{1+e^{-\sigma(V_1+\cdots+V_k-\tau)}} - \frac{1}{1+e^{\sigma\tau}}\right](1 + e^{-\sigma\tau})$	Steepness $\sigma > 0$ Threshold τ
Stepmod	$\mathbf{stepmod}_{\rho,\delta}(V_1, \ldots, V_k)$	0 if t mod $\rho < \delta$, else 1	Repetition ρ Duration δ

way network models are considered here, in order to avoid this problem of under-specification. That is the reason why aggregation in terms of combination functions is part of the definition of the network structure, in addition to connectivity in terms of connections and their weights and timing in terms of speed factors.

As part of the software environment, a large number > 5 of useful basic combination functions are included in a Combination Function Library, and also a facility to easily indicate any function composition of any available basic combination functions in the library. For a few examples of basic combination functions, see Table 2.1. Here $V_1, \ldots, V_k$ are variables for the single impacts.

The above concepts (the characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$) enable to design network models and their dynamics in a declarative manner, based on mathematically defined functions and relations for them. Note that for each state Y, all characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$ mentioned above affect the activation level of Y, as also can be seen from Eqs. (2.1) and (2.2). Each of these characteristics provide that influence in their own way from a specific role, either for connectivity, for aggregation or for timing. Below, this observation will also turn out useful in the context of self-models to address adaptivity.

2.2.2 Using Self-modeling Networks to Model Adaptive Networks

Realistic network models are usually adaptive: often some of their network characteristics $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$ change over time. For example, for mental networks often the connections are assumed to change by Hebbian learning (Hebb 1949) and for social networks, it is often assumed that connections between persons change, for example through a bonding by homophily principle (McPherson et al. 2001; Pearson et al. 2006; Sharpanskykh and Treur 2014).

Adaptive networks are often modeled in a hybrid manner by considering two different types of separate models that interact with each other: a network model for the base network and its within-network dynamics, and a numerical model for the adaptivity of (some of) the network structure characteristics of the base network. The latter dynamic model is usually specified in a format outside the context of network modeling: in the form of some adaptation-specific procedural or algorithmic programming specification used to run the difference or differential equations underlying the network adaptation process. In contrast, by including *self-models*, a network-oriented conceptualisation similar to what was described above, can also be applied to obtain adaptive networks as well by using a declarative description based on mathematically defined functions and relations for them.

The approach using self-models was inspired in a metaphorical sense by the more general idea of self-referencing or 'Mise en abyme', sometimes also called 'the Droste-effect' after the famous Dutch chocolat brand who uses this effect in packaging and advertising of their products since 1904. For some examples, see Fig. 2.1. For more explanation, see for example, https://en.wikipedia.org/wiki/Mise_en_abyme, https://en.wikipedia.org/wiki/Droste_effect. This effect occurs in art when within artwork a small copy of the same artwork is included. This can be applied graphically in paintings or photographs, or in sculptures. Also, it is sometimes used within literature (story-within-the-story), theater (theater-within-the-theater), or movies (movie-within-the-movie).

This idea is applied to model adaptation for a network by adding *self-models* to it as introduced in Treur (2020a, b). This leads to *self-modeling networks*, also called *reified* networks. This works through the addition of new states to the network (called *self-model states*) which represent network characteristics by network states. Then the impacts of these characteristics on a state *Y* as mentioned above can be modelled as impacts from such self-model states. This brings the impacts from these characteristics on a state *Y* in the standard form of a network model where via connections nodes affect other nodes.

Fig. 2.1 Three examples of the Mise en abyme or Droste-effect
http://michel.parpere.pagesperso-orange.fr/pedago/voc/mise%20en%20abyme.htm
https://www.instagram.com/culturfemale/.
https://www.instagram.com/p/CCYmVLMpGPo/

More specifically, adding a self-model for a base network is done in the way that for some of the states Y of the base network and some of the network structure characteristics for connectivity, aggregation and timing (i.e., some from $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$), additional network states $\mathbf{W}_{X,Y}$, $\mathbf{C}_{j,Y}$, $\mathbf{P}_{i,j,Y}$, $\mathbf{H}_Y$ (*self-model states* or *reification states*) are introduced and connected to other states:

(a) **Connectivity self-model**

- Self-model states $\mathbf{W}_{X,Y}$ are added representing connectivity characteristics, in particular connection weights $\boldsymbol{\omega}_{X,Y}$

(b) **Aggregation self-model**

- Self-model states $\mathbf{C}_{j,Y}$ are added representing aggregation characteristics, in particular combination function weights $\boldsymbol{\gamma}_{j,Y}$
- Self-model states $\mathbf{P}_{i,j,Y}$ are added representing aggregation characteristics, in particular combination function parameters $\boldsymbol{\pi}_{i,j,Y}$

(c) **Timing self-model**

- Self-model states $\mathbf{H}_Y$ are added representing timing characteristics, in particular speed factors $\boldsymbol{\eta}_Y$.

Note that the names using the letters **W**, **C**, **P** and **H** can also be chosen in a different manner. For example, for combination function parameter self-model states often names are used that refer to the specific parameter, for example, **T** for excitability threshold parameter $\boldsymbol{\tau}$, and **M** for persistence parameter $\boldsymbol{\mu}$. The step of adding a self-model to a base network is also called *network reification* and the resulting self-modeling network is sometimes called a *reified network*. If such self-model states are dynamic, they describe adaptive network characteristics. In a graphical 3D-format, such self-model states are depicted at a next level (also called *self-model level* or *reification level*), where the original network is at a *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 level) by a self-model state named $\mathbf{W}_{X,Y}$ (e.g., for an objective representation) or $\mathbf{RW}_{X,Y}$ (e.g., for a subjective representation).

Having self-model states to model an adaptation principle in a network-oriented manner is only a first step. To fully model a certain adaptation principle by a self-modeling network, the dynamics of each self-model state itself and its effect on a corresponding target state Y have to be specified in a network-oriented manner by the three general standard types of network characteristics (a) *connectivity*, (b) *aggregation*, and (c) *timing*:

(a) **Connectivity for the self-model states in a self-modeling network**

For the self-model states, their *connectivity* in terms of their incoming and outgoing connections has two different functions:

- **Effectuating its special effect from its specific role**
The *outgoing downward connections* from the self-model states $\mathbf{W}_{X,Y}$, $\mathbf{C}_{j,Y}$, $\mathbf{P}_{i,j,Y}$, $\mathbf{H}_Y$ to state Y represent the specific impact (their special effect from their specific role) each of these self-model states has on Y. These downward impacts are standard per role, and make that the adaptive values $\mathbf{W}_{X,Y}(t)$, $\mathbf{C}_{j,Y}(t)$, $\mathbf{P}_{i,j,Y}(t)$, $\mathbf{H}_Y(t)$ at t are actually used for the adaptive characteristics of the base network in Eqs. (2.1) and (2.2).

- **Indicating the input for the adaptation principle as specified in (b)**
The *incoming upward or leveled connections* to a self-model state are used to specify the *input* needed for the particular adaptation principle that is addressed.

(b) **Aggregation for the self-model states in a self-modeling network**

For the self-model states, their aggregation characteristics have one main aim:

- **Expressing the adaptation principle by a mathematical function**
For the *aggregation* of the incoming impacts for a self-model state, provided as indicated in a), a specific combination function is chosen *to express the adaptation principle* in a declarative mathematical manner.

(c) **Timing for the self-model states in a self-modeling network**

For the self-model states, their timing characteristics have one main aim:

- **Expressing the adaptation speed for the adaptation principle by a number**
Finally, like any other state, self-model states have their own *timing* in terms of speed factors. These speed factors are used as the means to express the adaptation speed.

As a base network extended by including a self-model is also a network model itself, as has been illustrated in Treur (2020b, Chap. 10), this self-modeling construction can easily be applied iteratively to include self-models of multiple self-modeling (or reification) levels. This can provide higher-order adaptive network models, and has turned out quite useful to model, for example, within Cognitive Neuroscience plasticity and metaplasticity (e.g., Abraham and Bear 1996; Garcia 2002; Magerl et al. 2018; Robinson et al. 2016) in a unified form by a second-order adaptive mental network with three levels, one base level and a first- and a second-order self-model level for plasticity and metaplasticity, respectively, as shown in Treur (2020b, Chap. 4).

In the current chapter, the notion of a multi-level self-modeling network model will be illustrated by two examples in Sects. 2.4 and 2.5 from which the latter addresses a higher-order adaptive network model for mental model handling that illustrates how the generic cognitive architecture for mental model handling discussed in Van Ments and Treur (2021, 2022) can be formalized in a computational manner using a self-modeling network.

2.3 Modeling Adaptation Principles

In this section, it will be shown how the modeling approach for self-modeling network models described in Sect. 2.2 can be applied to model adaptation principles as found in empirical sciences. When self-model states are changing over time in a proper manner, this offers a useful method to model any adaptation principle. This does not only apply to first-order adaptive networks, but also to second- or higher-order adaptive networks, for example to model control by using second-order self-models.

2.3.1 First-Order Self-models for First-Order Adaptation Principles

Within Cognitive Neuroscience literature, two types of (first-order) adaptation are often considered, one for connection weights and one for intrinsic neuronal properties; for example, as described in Chandra and Barkai (2018):

> Learning-related cellular changes can be divided into two general groups: modifications that occur at synapses and modifications in the intrinsic properties of the neurons. While it is commonly agreed that changes in strength of connections between neurons in the relevant networks underlie memory storage, ample evidence suggests that modifications in intrinsic neuronal properties may also account for learning related behavioral changes. (Chandra and Barkai 2018, p. 30).

In this chapter for these two types of adaptivity, two examples of first-order adaptation principles are considered: Hebbian Learning for connection weights and Excitability Modulation for the excitability threshold of states.

2.3.1.1 The Hebbian Learning Adaptation Principle

A well-known adaptation principle of the first type (addressing adaptive connectivity) is Hebbian Learning (Hebb 1949), 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.' (Hebb 1949, p. 62) (2.3)

This is sometimes simplified (neglecting the phrase 'as one of the cells firing B') to:

> What fires together, wires together. (Shatz 1992; Keysers and Gazzola 2014) (2.4)

Within a self-modeling network, this can be modeled by using a *connectivity self-model* based on self-model states $\mathbf{W}_{X,Y}$ representing connection weights $\boldsymbol{\omega}_{X,Y}$. These self-model states need incoming and outgoing connections to let them function within the network. To incorporate the 'firing together' part, for the self-model's connectivity, incoming connections from the connected states X and Y to $\mathbf{W}_{X,Y}$ are used; see Fig. 2.2 (upward arrows in blue). These upward connections have weight 1 here. Also a connection from $\mathbf{W}_{X,Y}$ to itself with weight 1 is used to model persistence of the learnt effect; in pictures they are usually left out. In addition, an outgoing connection from $\mathbf{W}_{X,Y}$ to state Y is used to indicate where this self-model state $\mathbf{W}_{X,Y}$ has its effect; see in Fig. 2.2 the (pink) downward arrow. The downward connection indicates that at the base level the value of $\mathbf{W}_{X,Y}$ is actually used for the connection weight of the connection from X to Y.

For the *aggregation characteristics* of the self-model, one of the options for a learning rule is defined by combination function $\mathbf{hebb}_{\mu}(V_1, V_2, W)$ from Table 2.2. This first-order adaptation principle will be illustrated both by the example adaptive network model discussed in Sect. 2.4 and by the example network for mental model handling in Sect. 2.5.

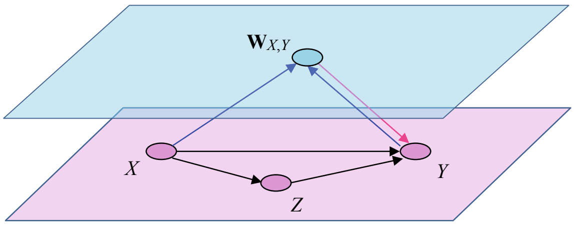

Fig. 2.2 Connectivity characteristics of the self-model for the Hebbian Learning adaptation principle

Table 2.2 Combination functions for self-models modeling first- and second-order adaptation principles used here in the illustrative examples

Name and self-model state	Combination functions	Variables and parameters
Hebbian learning $\mathbf{W}_{X,Y}$	$\mathbf{hebb}_{\mu}(V_1, V_2, W) = V_1 V_2(1 - W) + \mu W$	V_1,V_2 activation levels of the connected states W activation level of self-model state $\mathbf{W}_{X,Y}$ μ persistence factor
Excitability modulation $\mathbf{T}_Y$	$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k)$	$V_1, \ldots, V_k$ single impacts from base states
Exposure accelerates Adaptation $\mathbf{H}_{\mathbf{WT}_Y}$	$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k)$	$V_1, \ldots, V_k$ single impacts from base states and first-order self-model states

2.3.1.2 The Excitability Modulation Adaptation Principle

Although connectivity adaptation is most often addressed in the literature, it more recently has been pointed out that also other characteristics can be made adaptive, such as excitability thresholds. For example, the following quote indicates that synaptic activity induces long-lasting modifications in excitability of neurons:

> Long-lasting modifications in intrinsic excitability are manifested in changes in the neuron's response to a given extrinsic current (generated by synaptic activity or applied via the recording electrode). (Chandra and Barkai 2018, p. 30) (2.5)

For more literature on this form of learning or adaptation (called here the Excitability Modulation adaptation principle), see, for example, Aizenman and Linden (2000), Daoudal and Debanne (2003), Debanne et al. (2019), Lisman et al. (2018), Titley et al. (2017), Zhang and Linden (2003). As here the adaptation depends on activation of a base state Y and the base states (here X, Z) from which it gets its incoming connections, this can be modeled in a self-modeling network in a similar form as above, but this time using a self-model state $\mathbf{T}_Y$; see Fig. 2.3.

In this case, based on literature as referred above it is assumed that exposure enhances excitability, which means that it decreases the excitability threshold. To achieve this, for the self-model state $\mathbf{T}_Y$ a monotonically increasing combination function can be used, while the connection weights from X, Y, Z to $\mathbf{T}_Y$ are negative; examples of monotonically increasing combination functions are the logistic sum functions and the Euclidean function (with odd order $\boldsymbol{n}$) from Table 2.1. In this case, the (pink) downward connection from $\mathbf{T}_Y$ to Y indicates that the value of $\mathbf{T}_Y$ is used for the threshold value of the logistic sum function of base state Y. This first-order adaptation principle will be illustrated by the example network for mental model handling in Sect. 2.5.

2.3.2 *Second-Order Self-models for Second-Order Adaptation Principles*

The two first-order adaptation principles discussed in Sect. 2.3.1 refer to what in neuroscientific literature is called *plasticity*. It was shown how they can be described by a first-order self-model for connectivity or aggregation characteristics of the base

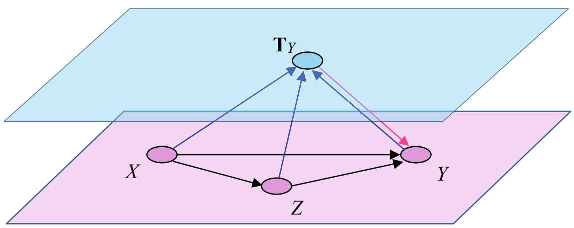

Fig. 2.3 Connectivity characteristics of a self-model for the Excitability Modulation adaptation principle

network, in this case in particular for the connection weights or the excitability thresholds used in aggregation. For an organism, in some circumstances it is better to learn (and change) fast, but in other circumstances, it is better to stay stable and let what has been learnt in the past persist: the Plasticity Versus Stability Conundrum (Sjöström et al. 2008, p. 773). Under which circumstances and to which extent such plasticity actually takes place is controlled by a form of socalled *metaplasticity*; e.g., Abraham and Bear (1996), Garcia (2002), Magerl et al. (2018), Robinson et al. (2016), Sjöström et al. (2008). Such control can address ‘The Plasticity Versus Stability Conundrum’ by only making plasticity happen in circumstances when it is important for the person to change and otherwise stabilise it. In literature as mentioned, various studies show how adaptation (as described, for example, by Hebbian learning), is modulated by accelerating the adaptation process or decelerating or even blocking it. Among the reported factors affecting plasticity in such a way are stimulus exposure, activation, previous experiences, and stress. Here we consider in particular three specific second-order adaptation principles for such control of first-order adaptation: the Adaptation Accelerates with Increasing Exposure, Exposure Modulates Persistence, and Stress Reduces Adaptation adaptation principles.

2.3.2.1 The Adaptation Accelerates with Increasing Exposure Adaptation Principle

For example, in (Robinson et al. 2016) the following compact quote is found summarizing that increasing stimulus exposure makes that the adaptation speed increases:

> Adaptation accelerates with increasing stimulus exposure’ (2.6)
> (Robinson et al. 2016, p. 2).

This indeed describes a form of metaplasticity that controls the speed of adaptation (learning rate). This principle can be modeled by a (dynamic) second-order self-model for timing characteristics (speed factors) of a first-order self-model for the first-order adaptation. Such a second-order is based on self-model states $\mathbf{H}_{\mathbf{W}_{X,Y}}$ or $\mathbf{H}_{\mathbf{T}_Y}$ for adaptive learning speed of any of the two types of (synaptic or intrinsic) learning discussed in Sect. 2.3.1, or $\mathbf{H}_{\mathbf{WT}_Y}$ for both types combined. The principle formulated by (6) indicates that the activation level of these second-order self-model states should depend in a monotonically increasing manner on the activation levels of the base states involved: these base states are Y itself and the states X, Z from which Y gets an incoming connection. This makes that the connectivity of this timing self-model (for both forms of learning) is as shown in Fig. 2.4: the (positive, blue) upward connections from the base states X, Y and Z to the self-model state $\mathbf{H}_{\mathbf{WT}_Y}$ are used to express the part of the principle in (6) referring to ‘stimulus exposure’. For the aggregation, for $\mathbf{H}_{\mathbf{WT}_Y}$, a Euclidean combination function (with odd order $\boldsymbol{n}$) or a logistic sum combination function can be used to get the monotonic effect as needed. The (blue) upward connections from $\mathbf{W}_{X,Y}$ and $\mathbf{T}_Y$ (with negative and positive weight, respectively) to the self-model state $\mathbf{H}_{\mathbf{WT}_{X,Y}}$ indicate a counterbalancing

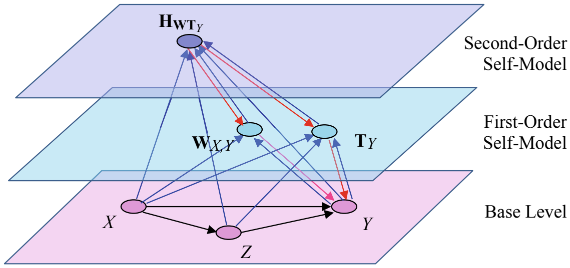

Fig. 2.4 Connectivity of a second-order self-model for the second-order Exposure Accelerates Adaptation adaptation principle for control of first-order self-models for Hebbian Learning and Excitability Modulation

effect that makes that the learning speed is limited depending on a high learnt level as represented by a high value of $\mathbf{W}_{X,Y}$ and a low value of $\mathbf{T}_Y$. The downward (pink) connections from $\mathbf{H}_{\mathbf{WT}_Y}$ to $\mathbf{W}_{X,Y}$ and $\mathbf{T}_Y$ indicate that the value of $\mathbf{H}_{\mathbf{WT}_Y}$ is actually used as speed factor for $\mathbf{W}_{X,Y}$ and $\mathbf{T}_Y$.

This second-order adaptation principle will be illustrated by the example network for mental model handling discussed in Sect. 2.5. As a small preview for this, Fig. 2.4 shows how a specific self-modeling network model for mental model handling can be obtained according to the more general three-level cognitive architecture for handling mental models put forward in Van Ments and Treur (2021) with the following three levels:

- **Base level**
  Applying a mental model: the lower (pink) plane.
- **First-order adaptation level**
  Adapting the mental model: the middle (blue) plane.
- **Second-order adaptation level**
  Controlling the adaptation of the mental model: the upper (purple) plane.

2.3.2.2 The Exposure Modulates Persistence Adaptation Principle

A similar perspective can be applied to obtain a principle for modulation of persistence.

Stimulus exposure modulates persistence of adaptation (2.7)

Depending on further context factors, this can be applied in different ways. Reduced persistence can be used in order to get rid of earlier learnt connections that do not apply anymore. However, enhanced persistence can be used to keep what has been learnt. This also is a form of metaplasticity, which can be described by a second-order adaptive network that is modeled using a dynamic second-order *aggregation self-model*, for persistence characteristics of a first-order self-model for the first-order adaptation, based on self-model states $\mathbf{M}_{\mathbf{W}_{X,Y}}$ for an adaptive persistence factor. This second-order adaptation principle will be illustrated by the example adaptive network model discussed in Sect. 2.4 (Fig. 2.5).

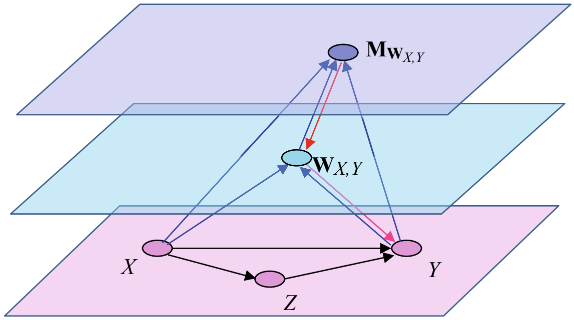

Fig. 2.5 Connectivity of a second-order self-model for the Exposure Modulates Persistence adaptation principle with a first-order self-model for Hebbian learning

2.3.2.3 The Stress Reduces Adaptation Adaptation Principle

In (Garcia 2002) the focus is on the role of stress in reducing or blocking plasticity. Many mental and physical disorders are stress-related, and are hard to overcome due to poor or even blocked plasticity that comes with the stress. Garcia (2002) describes the negative role of stress-related metaplasticity for this, which often becomes a situation that a patient is locked in his or her disorder by that negative pattern. However, he also shows that by some form of therapy this negative cycle may be broken:

> At the cellular level, evidence has emerged indicating neuronal atrophy and cell loss in response to stress and in depression. At the molecular level, it has been suggested that these cellular deficiencies, mostly detected in the hippocampus, result from a decrease in the expression of brain-derived neurotrophic factor (BDNF) associated with elevation of glucocorticoids. (Garcia 2002, p. 629).

> …modifications in the threshold for synaptic plasticity that enhances cognitive function is referred here to as 'positive' metaplasticity. In contrast, changes in the threshold for synaptic plasticity that yield impairment of cognitive functions, for example (..) in response to stress (..), is referred to as 'negative' metaplasticity. (Garcia 2002, pp. 630–631).

> In summary, depressive-like behavior in animals and human depression are associated with high plasma levels of glucocorticoids that produce 'negative' metaplasticity in limbic structures (…). This stress-related metaplasticity impairs performance on certain hippocampal-dependent tasks. Antidepressant treatments act by increasing expression of BDNF in the hippocampus. This antidepressant effect can trigger, in turn, the suppression of stress-related metaplasticity in hippocampal-hypothalamic pathways thus restoring physiological levels of glucocorticoids.' (Garcia 2002, p. 634).

For this second-order adaptation principle, a picture similar to what is shown in Fig. 2.4 can be drawn, but then for the case that one of the base states represents the stress level and the upward connection of that base state to the **H**-state at the second-order self-model level has a negative weight. This second-order adaptation principle will be illustrated in more detail by the example adaptive network model discussed in Sect. 2.4.

2.4 A Second-Order Adaptive Mental Self-modeling Network Model for Emotion Regulation Dysfunction

In this section, an example of a second-order adaptive network model is described for a mental health context. In such a second-order adaptive network, the base network has its own internal dynamics, but it also uses first-order adaptation principles. Moreover, these first-order adaptation principles themselves change based on second-order adaptation principles.

2.4.1 Design of the Adaptive Network Model for Emotion Regulation Dysfunction

Based on the literature discussed in Sect. 2.3, a second-order adaptive self-modeling network model for plasticity and metaplasticity has been designed with connectivity as shown in Fig. 2.6. Table 2.3 displays the explanations of the states. Here, blocked plasticity for emotion regulation due to stressful feelings is modeled, which leads to dysfunctioning emotion regulation (Garcia 2002).

In the base network, *s* is a stressful stimulus leading to stressful emotional response ps_b and stress feeling fs_b. State cs_b performs stress regulation by its negative outgoing connection, as soon as it is activated. However, to get cs_b activated, the connections

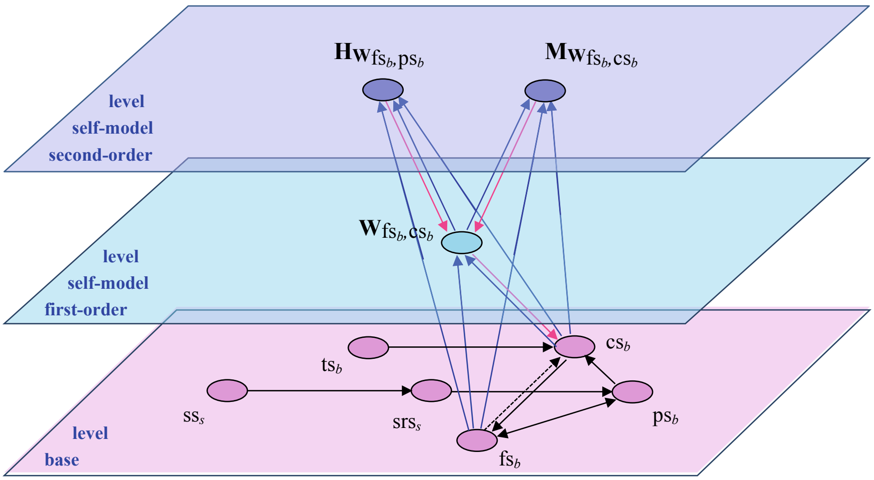

Fig. 2.6 Connectivity of the second-order adaptive network model for plasticity and metaplasticity of emotion regulation with base level (lower plane, pink), first-order self-model level (middle plane, blue) and second-order self-model level (upper plane, purple), and upward connections (blue) and downward connections (red) defining interlevel relations. The dashed arrow indicates the base level connection that is adaptive

Table 2.3 States of the second-order adaptive network model

State nr	State name	Explanation	Level
X_1	ss_s	Sensor state for stimulus s	Base level
X_2	srs_s	Sensory representation state for stimulus s	
X_3	ps_b	Preparation state for emotional response b	
X_4	fs_b	Feeling state for b	
X_5	cs_b	Control state for b	
X_6	ts_b	Therapy state for b	
X_7	$\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}$	Representation state for connection weight $\boldsymbol{\omega}_{\mathrm{srs}_s,\mathrm{ps}_a}$	First-order self-model level
X_8	$\mathbf{H}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$	Representation state for speed factor $\boldsymbol{\eta}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$ for representation state $\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}$	Second-order self-model level
X_9	$\mathbf{M}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$	Representation state for persistence factor parameter $\boldsymbol{\mu}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$ for representation state $\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}$	

from fs_b and ps_b to cs_b play an important role, and such connections have to be learnt and maintained, which may not happen if that plasticity is out of order. Many disorders, both physical and mental, originate from such poor functioning stress regulation. In the example, for the sake of simplicity, we focus on the connection from fs_b to cs_b. The learning process for this connection is modeled by the Hebbian learning principle (see Sect. 2.3.1.1 and Fig. 2.2) represented by first-order self-model state $\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}$. This state models a Hebbian learning adaptation principle defined as a combination function by the function $\mathbf{hebb}_{\boldsymbol{\mu}}(V_1, V_2, W)$ as indicated in Table 2.2.

However, as discussed in Sect. 2.3.2, whether or not learning takes place depends on metaplasticity, modelled here by second-order self-model states $\mathbf{H}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$ and $\mathbf{M}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$ for learning speed and persistence, respectively (see also Sects. 2.3.2.1 and 2.3.2.2 and Figs. 2.4 and 2.5). Note that the **M**-state is an aggregation self-model state for a combination function parameter: the persistence parameter $\boldsymbol{\mu}$ for the combination function $\mathbf{hebb}_{\boldsymbol{\mu}}(..)$ for first-order adaptation state $\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}$ for Hebbian learning of the connection from fs_b to cs_b. In principle, the learning will start to work or accelerate when the external stimulus s is sensed through sensor state ss_s. As discussed in Sect. 2.3.2.3, stress creates a negative metaplasticity effect, which corresponds to low values for these second-order self-model states. For example, a value close to 0 for learning speed representation $\mathbf{H}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$ practically blocks the learning, and a value around 0.5 for persistence representation $\mathbf{M}_{\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}}$ makes that every time unit, around 50% of the learnt effect is lost, which is a dramatic effect if no additional learning takes place.

For most of the states in the designed network model, for the aggregation characteristics $\mathbf{alogistic}_{\sigma,\tau}(..)$ is used as combination function; see Table 2.1. The only exceptions are $\mathbf{W}_{\mathrm{fs}_b,\mathrm{cs}_b}$ which uses $\mathbf{hebb}_{\boldsymbol{\mu}}(..)$ defined by (4) and Table 2.2, and therapy state ts_b which is an external input modeled by $\mathbf{stepmod}_{\rho,\delta}$**(..)** which defines an (independent) activation after *duration* $\boldsymbol{\delta}$ and *repetition* of the cycle after a time period $\boldsymbol{\rho}$.

2.4.2 Specification of the Adaptive Network Model for Emotion Regulation Disfunction

To get a well-defined standard format to specify a design of a self-modeling network, the socalled role matrix format has been introduced in Treur (2020b). In the first place, a specification in that format can be used as a compact but detailed, neat and standardised form of documentation for human use, for example as a basis for communication among designers. It does not only cover all information on connectivity as represented in graphical format in pictures such as shown in Fig. 2.6, but also on all other network characteristics defining a network model, such as $\boldsymbol{\omega}_{X,Y}$, $\boldsymbol{\gamma}_{j,Y}$, $\boldsymbol{\pi}_{i,j,Y}$, $\boldsymbol{\eta}_Y$ as discussed in Sect. 2.2. Therefore, a self-modeling network model is fully defined by this specification in role matrix format. For an overview of the five role matrices in relation to the network characteristics, see Table 2.4. In the second place, the standardized format of the role matrices makes it relatively easy to implement a software environment that can use them as input and based on that runs simulations. So, role matrices do not only form a good basis for documentation and communication among humans, they are also a good basis for communication with computers. Such a software environment and how to use it is described in Treur (2020b, Chap. 9); see also Treur (2022a).

In Figs. 2.7 and 2.8 all network characteristics for the designed adaptive network model for emotion regulation dysfunction are specified in the form of role matrices. Role matrix **mb** in Fig. 2.7 specifies the *base connectivity* characteristics. On each row, for the given state (in the left column) it indicates from which states at the same or a lower level it gets an incoming connection (the black and blue arrows in Fig. 2.6). Note that for some of the states a connection from the state itself occurs. The latter applies to all (first- and second-order) self-model states, as can be seen in **mb**. Such

Table 2.4 Overview of the role matrices for the different types of network characteristics

	Network characteristics	Role matrix	Notation
Connectivity characteristics	Base connectivity	**mb**	Picture (upward and horizontal arrows)
	Connection weights	**mcw**	$\boldsymbol{\omega}_{X,Y}$
Aggregation characteristics	Combination function weights	**mcfw**	$\boldsymbol{\gamma}_{j,Y}$
	Combination function parameters	**mcfp**	$\boldsymbol{\pi}_{i,j,Y}$
Timing characteristics	Speed factors	**ms**	$\boldsymbol{\eta}_Y$

mb base connectivity		1	2	3	4
X_1	ss_s	X_1			
X_2	srs_s	X_1			
X_3	ps_b	X_2	X_4		
X_4	fs_b	X_3	X_5		
X_5	cs_b	X_3	X_4	X_6	
X_6	ts_b	X_6			
X_7	$\mathbf{W}_{fs_b,cs_b}$	X_4	X_5	X_7	
X_8	$\mathbf{H}\mathbf{w}_{fs_b,cs_b}$	X_4	X_5	X_7	X_8
X_9	$\mathbf{M}\mathbf{w}_{fs_b,cs_b}$	X_4	X_5	X_7	X_9

mcw connection weights		1	2	3	4
X_1	ss_s	1			
X_2	srs_s	1			
X_3	ps_b	1			
X_4	fs_b	1	-1		
X_5	cs_b	0.5	X_7	1	
X_6	ts_b	1			
X_7	$\mathbf{W}_{fs_b,cs_b}$	1	1	1	
X_8	$\mathbf{H}\mathbf{w}_{fs_b,cs_b}$	-0.5	1	0.1	1
X_9	$\mathbf{M}\mathbf{w}_{fs_b,cs_b}$	-0.15	0.5	0.05	1

ms speed factors		1
X_1	ss_s	0.5
X_2	srs_s	0.5
X_3	ps_b	0.2
X_4	fs_b	0.02
X_5	cs_b	0.2
X_6	ts_b	4
X_7	$\mathbf{W}_{fs_b,cs_b}$	X_8
X_8	$\mathbf{H}\mathbf{w}_{fs_b,cs_b}$	0.1
X_9	$\mathbf{M}\mathbf{w}_{fs_b,cs_b}$	0.02

Fig. 2.7 Specification of the *connectivity characteristics* and *timing characteristics* of the second-order adaptive network model for emotion regulation dysfunction by role matrices **mb**, **mcw** and **ms**

mcfw combination function weights		1 alogistic	2 hebb	3 stepmod
X_1	ss_s	1		
X_2	srs_s	1		
X_3	ps_b	1		
X_4	fs_b	1		
X_5	cs_b	1		
X_6	ts_b			1
X_7	$\mathbf{W}_{fs_b,cs_b}$		1	
X_8	$\mathbf{H}\mathbf{w}_{fs_b,cs_b}$	1		
X_9	$\mathbf{M}\mathbf{w}_{fs_b,cs_b}$	1		

mcfp	function	1 alogistic		2 hebb		3 stepmod	
	parameter	1 σ	2 τ	1 μ	2	1 ρ	2 δ
X_1	ss_s	5	0.2				
X_2	srs_s	5	0.2				
X_3	ps_b	5	0.2				
X_4	fs_b	5	0.4				
X_5	cs_b	5	0.7				
X_6	ts_b					200	100
X_7	$\mathbf{W}_{fs_b,cs_b}$			X_9			
X_8	$\mathbf{H}\mathbf{w}_{fs_b,cs_b}$	5	0.9				
X_9	$\mathbf{M}\mathbf{w}_{fs_b,cs_b}$	5	0.6				

Fig. 2.8 Specification of the *aggregation characteristics* of the second-order adaptive network model for emotion regulation disfunction by role matrices **mcfw** and **mcwfp**

connections are usually not depicted in graphical representations such as the one in Fig. 2.6.

As an example, in the second row, it is indicated that state X_2 (= srs_s) only has one incoming base connection, from state X_1 (= ss_s). As another example, the seventh row indicates that state X_7 (= $\mathbf{W}_{fs_b,cs_b}$) has incoming base connections from X_4 (= fs_b), X_5 (= cs_b), X_7 (= $\mathbf{W}_{fs_b,cs_b}$) itself, and in that order. This order is important as the Hebbian combination function $\mathbf{hebb}_\mu$(..) used is not symmetric in its arguments. Note that the more informative state names such as ss_s, and so on, in each of the role matrices depicted in Figs. 2.7 and 2.8 are actually not part of the specification as used in the computer, but are only for human understanding. In a similar way the other types of role matrices are defined; see Figs. 2.7 and 2.8: role matrices **mcw** for connection weights, **mcfw** for combination function weights, **mcfp** for combination function parameters, and **ms** for speed factor roles. Here the combination functions selected from the library are specified by **mcf** = [….], for the current example it is **mcf** = [2 3 35]; here the numbers 2, 3, 35 refer to the numbers in the combination function library, where **alogistic**(…) has number 2 and **hebb**(…) number 3; number 35 is the **stepmod** function used to create independent events. By specifying **mcf** = [2 3 35] for this specific network model they become combination function numbers 1 to 3 as also shown in role matrices **mcfw** and **mcfp**.

Within each role matrix, for an adaptive network characteristic, entries in red cells indicate a reference to the name of another state that as self-model state represents that characteristic, while entries in green cells indicate fixed values for nonadaptive characteristics. In this way the red cells represent the pink downward connections from the self-model states in pictures as shown in Fig. 2.6, with their specific roles **W**, **H**, **C**, **P** indicated by the type of role matrix: the type of role matrix in which they are represented, defines the roles of the self-model states. For example, in **mcw** the X_7 in the peach-red cell in the row for X_5 defines that the connection weight for the connection to X_5 (= cs_b) from X_4 (= fs_b) is adaptive with value represented by X_7 (= $\mathbf{W}_{fs_b,cs_b}$). So, by specifying X_7 in role matrix **mcw** in that cell, X_7 gets the role of connection weight self-model state for the connection from fs_b to cs_b. Similarly, role matrix **ms** indicates (in peach-red) that X_8 plays the role of the (adaptive) speed factor of X_7, and (in green) that the speed factors of all other states have fixed values.

In Fig. 2.8 the role matrices **mcfw** and **mcfp** are shown for aggregation characteristics in terms of combination function weights and parameters, respectively. Matrix **mcfp** is a 3D matrix with first dimension for the states, second dimension for the (two) combination function parameters and third dimension for the combination functions. For example, in Fig. 2.8 the name X_9 in the red cell in role matrix **mcfp** indicates that the value of the persistence parameter $\boldsymbol{\mu}$ for X_7 (= $\mathbf{W}_{fs_b,cs_b}$) is adaptive and is represented by the value of state X_9 (= $\mathbf{M}_{\mathbf{W}_{fs_b,cs_b}}$). In contrast, the 5 in the first green cell of **mcfp** for X_5 indicates the static value of the steepness of the logistic function for X_5 (= cs_b).

For this example network model, the selection of combination functions from the library for the network is specified by **mcf** = [2 3 35], being $\mathbf{alogistic}_{\sigma,\tau}$(..), $\mathbf{hebb}_{\mu}$(..), $\mathbf{stepmod}_{\rho,\delta}$**(..)**, respectively. So, in terms of (2) for this network it holds c_1(..) = $\mathbf{alogistic}_{\sigma,\tau}$(..), c_2(..) = $\mathbf{hebb}_{\mu}$(..), c_3(..) = $\mathbf{stepmod}_{\rho,\delta}$**(..)**.

2.4.3 Simulations for the Adaptive Network for Emotion Regulation Dysfunction

A number of simulation experiments have been performed using the dedicated software environment for self-modeling network models described in Treur (2020), Chap. 9); see also Treur (2022a). In particular, a scenario is shown here in which the focus was on the effect of the stress level of fs_b on plasticity, following Garcia (2002). In Fig. 2.9 simulation results are shown for the characteristics in Fig. 2.1 and 2.2. Here a person is considered who for some time (before time 0) has had a

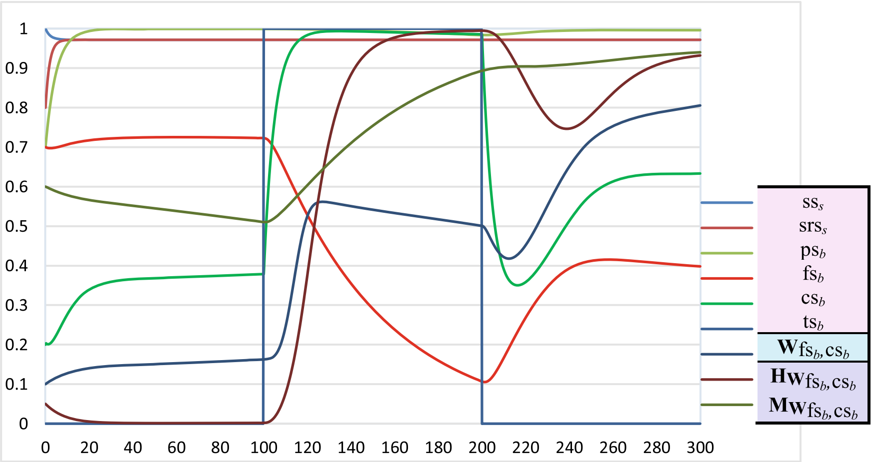

Fig. 2.9 Simulations for the second-order adaptive example network model. First phase (time 0 to 100): a high stress level while metaplasticity blocks plasticity, due to which emotion regulation does not function properly. Second phase (time 100–200): a therapy artifically boosts emotion regulation, which makes metaplasticity unblock the plasticity which in turn makes emotion regulation stronger. Third phase (time 200–300): due to the unblocked plasticity, the emotion regulation is now able to learn further and keep the stress level low without therapy

Table 2.5 Initial values for the simulation

ss_s	srs_s	ps_b	fs_b	cs_b	ts_b	$\mathbf{W}_{fs_b,cs_b}$	$\mathbf{HW}_{fs_b,cs_b}$	$\mathbf{MW}_{fs_b,cs_b}$
1	0.8	0.7	0.7	0.2	0.1	0.1	0.05	0.6

stressful life, so that the initial values already reflect a high stress level. Initial values were as shown in Table 2.5.

The graph in Fig. 2.9 shows the activation levels of all states, including how the weight of the connection from fs_b to cs_b represented by $\mathbf{W}_{fs_b,cs_b}$ is learnt, and what speed and persistence values are applied for that, represented by $\mathbf{HW}_{fs_b,cs_b}$ and $\mathbf{MW}_{fs_b,cs_b}$. As can be seen, in the first phase until time 100 a high stress level represented by fs_b (the red line) leads to maintaining low values of $\mathbf{HW}_{fs_b,cs_b}$ (the brown line starting at 0.05) and $\mathbf{MW}_{fs_b,cs_b}$ (the grey line starting at 0.6); therefore learning is practically blocked in this phase (the blue line for $\mathbf{W}_{fs_b,cs_b}$ starting at 0.1 stays low). In the next phase, from time 100 to time 200 a therapy is applied, represented by ts_b, that gives a boost to the activation level of cs_b, which in turn reduces the stress level represented by fs_b. This also increases the speed and persistence of the Hebbian learning (second-order states $\mathbf{HW}_{fs_b,cs_b}$ and $\mathbf{MW}_{fs_b,cs_b}$ increase to 1 and 0.9, respectively) and because of this, now indeed learning takes place. In the last phase from time 200 to time 300 the therapy has finished, but as due to the therapy the stress regulation mechanism has been unlocked, now it is able to keep the stress levels low without external help.

2.5 An Example Network Model for Mental Model Handling

In this section it is shown how an adaptive network model to handle a mental model can be designed. This is done based on the three-level cognitive architecture described in Van Ments and Treur (2021, 2022). For this, first an example scenario is described in Sect. 2.5.1. After that in Sects. 2.5.2 and 2.5.3 the network design and its specifation are presented.

2.5.1 An Example Scenario for Mental Model Handling

The scenario used for this section concerns learning of a mental model by a new person in a company who has to learn to recognize a colleague. It goes as follows.

Example Scenario

A new person in a company has to learn to recognize a colleague from only seeing his face; this face is stimulus s. Two colleagues a_1 and a_2 are assumed that are options to choose from. Picking one of them is indicated by activation of sensory representation state srs_{a_i}. A belief bs_1 suggests that it is colleague a_1, and a belief bs_2 that it is colleague a_2. These beliefs are only meant indicative (e.g., based on the location at which the person is encountered), but not sufficient to decide for one of them. As the beliefs and s are triggered by independent circumstantial factors, for the network model they just happen. Two types of network characteristics are addressed as adaptive: the weights of the connections from sensory representation srs_s to srs_{a_1} and to srs_{a_2}, and the excitability thresholds for sensory representation states srs_{a_1} and srs_{a_2}. The small network consisting of these three base level states together with the first-order self-model states representing the characteristics for connection weights and excitability thresholds form a mental model of our subject for the colleague considered here; the connection defines how strongly (in the mental model in the mind of our subject) the face relates to the name of the person. During the scenario these characteristics are learnt so that over time a better mental model and decision result. Then in future situations an encounter with s (also at unexpected locations, such as in a shop or in another town) leads to correct recognition.

2.5.2 Connectivity and Aggregation for the Adaptive Network Model

In this section a second-order adaptive network model is introduced that addresses the above example scenario, according to the three-level cognitive architecture pointed out in Van Ments and Treur (2021, 2022):

- a base level for the base mental model
- a first-order self-model level for adaptation of the mental model
- a second-order self-model for control of the adaptation.

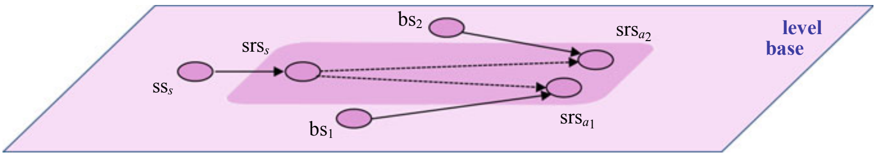

Fig. 2.10 The base level for the example mental model for recognition; the darker shaded part is the mental model, the dashed arrows indicate the connections that are adaptive

As discussed in Van Ments and Treur (2021, 2022), the general idea is that a mental model is some structure defined by relations. Using the network-oriented modeling approach adopted here, they can be described by a base network where the relations are modeled as connections. As these relations can change, for example by learning, a (first-order) self-model of the base model is used with self-model states that represent the relations of the base level. At the self-model level, these self-model states are also connected by some type of relations that define when and how adaptation can take place. These relations are represented by second-order self-model states in a second-order self-model. These second-order self-model states determine control of the adaptation at the first self-model level. To determine that, they have their own relations for the second-order self-model level. Based on the network-oriented modeling approach used here, all such relations at (and between) the different levels are modeled as connections.

For the example described in Sect. 2.5.1, first in Fig. 2.10 the connectivity of the base level is depicted. The base mental model consists of three states srs_s, srs_{a_1} and srs_{a_2} (in the darker sheded area) and relations between them. Here especially the relations from srs_s to srs_{a_1} and from srs_s to srs_{a_2} are considered, as depicted by dashed arrows in Fig. 2.10.

After observing the colleague (via ss_s), sensory representation state srs_s gets activated, which, depending on the weights of the connections from srs_s to srs_{a_1} and srs_{a_2} and the excitability thresholds of srs_{a_1} and srs_{a_2} ideally activates one of srs_{a_1} and srs_{a_2} indicating the correct colleague, without any of the beliefs bs_1 and bs_2 being activated. However, in the beginning it is not that ideal: in the first phase, the activation of the correct belief is needed to be able to make a choice between srs_{a_1} and srs_{a_1}. Due to learning, later on this dependence on the beliefs is not needed anymore as the connection from srs_s to the relevant option is strengthened by this learning and the excitability threshold gets lower for that option.

The following two types of self-model states are used to define the adaptation by learning in the considered mental model:

- **Connectivity self-model states for connection weights**
  The states $\mathbf{W}_{X,Y}$ play the role of connection weight for the adaptive connection from X to Y. They model the Hebbian Learning adaptation principle (Hebb 1949); see also (4) in Sect. 2.3.1.1.

- **Aggregation self-model states for excitability thresholds**
  The states $\mathbf{T}_Y$ play the combination function parameter role for state Y's adaptive excitability threshold $\boldsymbol{\tau}$. They model the Excitability Modulation adaptation principle (Chandra and Barkai 2018); see also (5) in Sect. 2.3.1.2.

Moreover, to control the adaptation, second-order timing self-model states are used:

- **Timing self-model states for connection weight self-model states**
  The second-order self-model states $\mathbf{H}_{\mathbf{WT}_{X,Y}}$ play the role of speed factor for the first-order connection weight self-model states $\mathbf{W}_{X,Y}$ for the adaptive connections from X to Y and for the first-order excitability threshold self-model states $\mathbf{T}_Y$ for the adaptive excitability thresholds of Y. These second-order self-model states model the second-order adaptation principle called Adaptation Accelerates with Increased Exposure (Robinson et al. 2016); see also (6) in Sect. 2.3.2.1.

All in all, this creates a (sub)network for the core mental model and including its adaptation and control based on the following states:

- base state $\text{srs}_s$for the image of the face
- the two base states srs_{a_1} and srs_{a_2} for the options of colleagues and their excitability thresholds
- the two connections (dashed arrows) from srs_s to srs_{a_1} and srs_{a_2} for the options of colleagues with their weights
- the two first-order connectivity self-model states $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$ and $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_2}}$ for the weights of these two base connections
- the two first-order aggregation self-model states $\mathbf{T}_{\text{srs}_{a_1}}$ and $\mathbf{T}_{\text{srs}_{a_2}}$ for the excitability thresholds of states srs_{a_1} and srs_{a_2}
- the two second-order timing self-model states $\mathbf{H}_{\mathbf{WT}_{X,Y}}$ for the connectivity self-model states $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$ and $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_2}}$ for the weights of these two base connections and for the first-order aggregation self-model states $\mathbf{T}_{\text{srs}_{a_1}}$ and $\mathbf{T}_{\text{srs}_{a_2}}$ for the excitability thresholds of states srs_{a_1} and srs_{a_1}

This core mental model can also be extended by adding the base level belief states to it as well. In a graphical representation of the mental model's *connectivity* in a 3D format, the first-order self-model states are placed in a second (blue) plane, above the (pink) plane for the base mental model, and the second-order self-model states in a third (purple) plane above the second (blue) plane. See Fig. 2.11 and see Table 2.2 for explanations of all states. The following types of connection are used: upward and downward connections, and horizontal leveled connections. Downward connections have a particular effect, as they are effectuating one of the types of adaptive characteristics indicated by their role **W**, **C**, **P** or **H**; see also Sect. 2.2 (Fig. 2.11).

For *aggregation*, in the example mental model, for the three base level states the logistic function $\textbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(\ldots)$ is used, and also for the aggregation (excitability threshold) self-model states $\mathbf{T}_{\text{srs}_{a_1}}$ and $\mathbf{T}_{\text{srs}_{a_2}}$. To model Hebbian learning, combination function $\textbf{hebb}_{\boldsymbol{\mu}}(V_1, V_2, W)$ is applied for the two connectivity (connection weight) self-model states $\mathbf{W}_{X,Y}$ of the mental model; see Table 2.6.

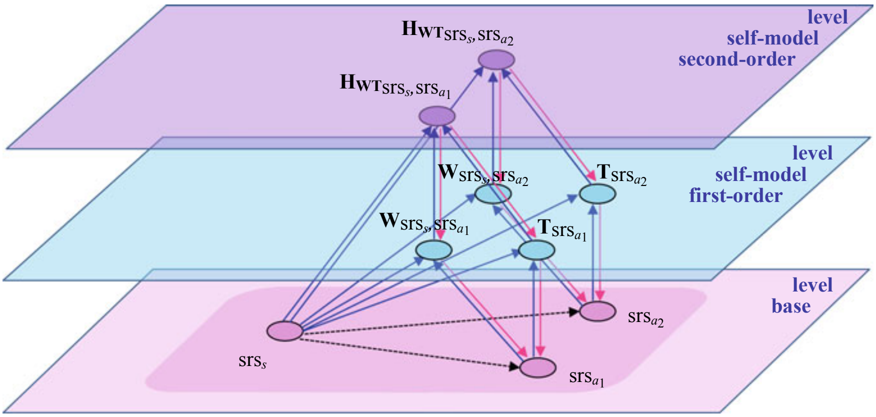

Fig. 2.11 3D representation of the connectivity of the core mental model for recognition, including: (1) *base level* for the face recognition (depicted by the lower, pink plane), (2) *first-order self-model level* (depicted by the middle, blue plane) for the two **W**-states for the weights of the base connections from srs_s to srs_{a_1} and srs_{a_2} and for the two **T**-states for the excitability thresholds for the two base states srs_{a_1} and srs_{a_2}, (3) *second-order self-model level* (depicted by the upper, purple plane) for the two $\mathbf{H}_{\mathbf{WT}}$-states for the speed factors for the **W**-states for the weights of the base connections from srs_s to srs_{a_1} and srs_{a_2} and for the two **T**-states for the excitability thresholds for the two base states srs_{a_1} and srs_{a_2}

Table 2.6 The states used in the example network model for mental model handling

State nr	name	Explanation
X_1	ss_s	Sensor state for stimulus *s* (seeing a face)
X_2	srs_s	Sensory representation state for stimulus *s*
X_3	bs_1	Belief state 1 (belief that it is Person 1)
X_4	bs_2	Belief state 2 (belief that it is Person 2)
X_5	srs_{a1}	Sensory representation state for recognition as Person 1
X_6	srs_{a2}	Sensory representation state for recognition as Person 2
X_7	$\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$	First-order self-model state for the weight of the connection from srs_s to srs_{a_1}
X_8	$\mathbf{W}_{\text{srs}_s,\text{srs}_{a_2}}$	First-order self-model state for the weight of the connection from srs_s to srs_{a_2}
X_9	$\mathbf{T}_{\text{srs}_{a_1}}$	First-order self-model state for the excitability threshold of srs_{a_1}
X_{10}	$\mathbf{T}_{\text{srs}_{a_2}}$	First-order self-model state for the excitability threshold of srs_{a_2}
X_{11}	$\mathbf{H}_{\mathbf{WT}_{\text{srs}_s,\text{srs}_{a_1}}}$	Second-order self-model for the speed factor of the first-order self-model state $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$ for the weight of the connection from srs_s to srs_{a_1} and of the first-order self-model state $\mathbf{T}_{\text{srs}_{a_1}}$ for the excitability threshold of srs_{a_1}
X_{12}	$\mathbf{H}_{\mathbf{WT}_{\text{srs}_s,\text{srs}_{a_2}}}$	Second-order self-model for the speed factor of the first-order self-model state $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_2}}$ for the weight of the connection from srs_s to srs_{a_1} and of the first-order self-model state $\mathbf{T}_{\text{srs}_{a_2}}$ for the excitability threshold of srs_{a_2}

2.5.3 Specification of the Example Network Model for Mental Model Handling

The network model for mental model handling was specified by role matrices as shown in Fig. 2.12. As discussed in Sect. 2.4.2, they are **mb** (for the base connection role), **mcw** (for the connection weight role), **ms** (for the speed factor role), **mcfw** (for the combination function weight role), and **mcfp** (for the combination function parameter role).

For a designed model a list of combination functions used is specified by **mcf** = [….], for the current example it is **mcf** = [2 3 35]; here the numbers 2, 3, 35 refer to the numbers in the combination function library, where $\mathbf{alogistic}_{\sigma,\tau}(\ldots)$ has number 2 and $\mathbf{hebb}_{\mu}(\ldots)$ number 3; number 35 is the stepmod function used to create independent events. Figure 2.12 shows all role matrices for the adaptive network model addressing mental model handling.

Role matrix **mb** specifying *base connectivity* indicates at each row for the indicated state X_j from which states it gets incoming connections from the same or a lower level. For example, the 5[th] row indicates for state X_5 (= srs_{a_1}) two incoming base connections, one from state X_2 (= srs_s), and one from state X_3 (= bs_1). For another example, row 7 indicates that state X_7 (= $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$) has incoming base connections from X_2 (= srs_s), X_5 (= srs_{a_1}) and from X_7 itself in that order; this ordering is crucial since the Hebbian combination function **hebb**(…) used for this state X_7 (= $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$) is not symmetric in its three arguments, as can be seen in Table 2.2.

The other four role matrices are as follows: role matrices **mcw** for the connection weight role and **ms** for the speed factor role, and role matrices **mcfw** for the combination function weight role and **mcfp** for the combination function parameter role (see Fig. 2.12). Within each of these non-base role matrices cell entries in peach-red cells show the name of a state (at a higher level) that as self-model state represents an adaptive characteristic; in contrast, entries in green cells indicate static values for nonadaptive characteristics. Therefore, as seen in Fig. 2.12 the peach-red cells in **mcw** and **mcfp** refer to the (self-model) states X_7 to X_{10}. For example, in role matrix **mcw** the indication X_7 in the peach-red cell at row 5 and column 1 specifies that the value of state X_7 represents the weight of the connection from srs_s to srs_{a_1} (as indicated in **mb**). Unlike this, the 1 in green cell at row 7, column 1 of **mcw** shows the nonadaptive value of weight of the connection from X_2 (= srs_s) to X_7 (= $\mathbf{W}_{\text{srs}_s,\text{srs}_{a_1}}$). In role matrix **mcfp** specifying the combination function parameter role, in the peach-red cell at row 5 and column 2 it is specified that the actual value for the excitability threshold of srs_{a_1} is represented by the value of self-model state X_9 (= $\mathbf{T}_{\text{srs}_{a_1}}$). More explanation of this specification format and how it is used to automatically generate simulations can be found in Treur (2020a, b, 2022a).

A simulation scenario and a more extensive analysis from an informational viewpoint of this example network for mental model handling can be found in Sects. 16.6 and 16.7 (in Chap. 16 of this volume) of Treur (2022e).

mb	base connectivity	1	2	3
X_1	ss_s	X_1		
X_2	srs_s	X_1		
X_3	bs_1	X_3		
X_4	bs_2	X_4		
X_5	srs_{a1}	X_2	X_3	
X_6	srs_{a2}	X_2	X_4	
X_7	$\mathbf{W}srs_s,srs_{a1}$	X_2	X_5	X_7
X_8	$\mathbf{W}srs_s,srs_{a2}$	X_2	X_6	X_8
X_9	$\mathbf{T}srs_{a1}$	X_2	X_5	X_9
X_{10}	$\mathbf{T}srs_{a2}$	X_2	X_6	X_{10}
X_{11}	$\mathbf{HWT}srs_s,srs_{a1}$	X_1	X_7	X_9
X_{12}	$\mathbf{HWT}srs_s,srs_{a2}$	X_1	X_8	X_{10}

mcw	connection weights	1	2	3
X_1	ss_s	1		
X_2	srs_s	1		
X_3	bs_1	1		
X_4	bs_2	1		
X_5	srs_{a1}	X_7	0.5	
X_6	srs_{a2}	X_8	0.5	
X_7	$\mathbf{W}srs_s,srs_{a1}$	1	1	1
X_8	$\mathbf{W}srs_s,srs_{a2}$	1	1	1
X_9	$\mathbf{T}srs_{a1}$	-0.2	-0.2	1
X_{10}	$\mathbf{T}srs_{a2}$	-0.2	-0.2	1
X_{11}	$\mathbf{HWT}srs_s,srs_{a1}$	1	-1	1
X_{12}	$\mathbf{HWT}srs_s,srs_{a2}$	1	-1	1

mcfw	combination function weights	1 alo-gistic	2 hebb	3 step-mod
X_1	ss_s			1
X_2	srs_s	1		
X_3	bs_1			1
X_4	bs_2			1
X_5	srs_{a1}	1		
X_6	srs_{a2}	1		
X_7	$\mathbf{W}srs_s,srs_{a1}$		1	
X_8	$\mathbf{W}srs_s,srs_{a2}$		1	
X_9	$\mathbf{T}srs_{a1}$	1		
X_{10}	$\mathbf{T}srs_{a2}$	1		
X_{11}	$\mathbf{HWT}srs_s,srs_{a1}$	1		
X_{12}	$\mathbf{HWT}srs_s,srs_{a2}$	1		

mcfp	function	1 alogistic		2 hebb		3 stepmod	
	parameter	1 σ	2 τ	1 μ	2	1 ρ	2 δ
X_1	ss_s					50	25
X_2	srs_s	5	0.8				
X_3	bs_1					70	60
X_4	bs_2					50	25
X_5	srs_{a1}	5	X_9				
X_6	srs_{a2}	5	X_{10}				
X_7	$\mathbf{W}srs_s,ps_{a1}$			0.95			
X_8	$\mathbf{W}srs_s,ps_{a2}$			0.95			
X_9	$\mathbf{T}srs_{a1}$	5	0.4				
X_{10}	$\mathbf{T}srs_{a2}$	5	0.4				
X_{11}	$\mathbf{HWT}srs_s,srs_{a1}$	5	0.8				
X_{12}	$\mathbf{HWT}srs_s,srs_{a2}$	5	0.8				

ms	speed	1
X_1	ss_s	2
X_2	srs_s	0.5
X_3	bs_1	2
X_4	bs_2	2
X_5	srs_{a1}	0.2
X_6	srs_{a2}	0.5
X_7	$\mathbf{W}srs_s,srs_{a1}$	0.3
X_8	$\mathbf{W}srs_s,srs_{a2}$	0.3
X_9	$\mathbf{T}srs_{a1}$	0.07
X_{10}	$\mathbf{T}srs_{a2}$	0.07
X_{11}	$\mathbf{HWT}srs_s,srs_{a1}$	0.2
X_{12}	$\mathbf{HWT}srs_s,srs_{a2}$	0.2

initial values		
X_1	ss_s	0
X_2	srs_s	0
X_3	bs_1	0
X_4	bs_2	0
X_5	srs_{a1}	0
X_6	srs_{a2}	0
X_7	$\mathbf{W}srs_s,ps_{a1}$	0.3
X_8	$\mathbf{W}srs_s,ps_{a2}$	0.3
X_9	$\mathbf{T}srs_{a1}$	0.8
X_{10}	$\mathbf{T}srs_{a2}$	0.8
X_{11}	$\mathbf{HWT}srs_s,srs_{a1}$	0
X_{12}	$\mathbf{HWT}srs_s,srs_{a2}$	0

Fig. 2.12 Role matrices specification for the example network for mental model handling

2.6 Discussion

For many domains network models provide an intuitive, declarative way of modeling supported by graphical representations. Connections between nodes in a network can be used as a format to model different types of relations occurring in real-world situations. Once network models are represented within a computer, these relations can be used for some types of computational processes to generate *within-network dynamics*, such as simulation processes or reasoning processes or some (other) types of analyses. However, as relations in real-world domains often change over time themselves too, network models for realistic situations also need facilities to change their structure. This is called dynamics *of* networks, or *network adaptation*. It has turned out that socalled self-modeling networks enable to model such changes in network structure relatively easily. These are networks that include nodes that represent specific network characteristics of the network itself and in this way form a self-model of part of the network's own structure. By using a self-model, the adaptation of the network structure can be modeled as within-network dynamics of this self-model.

The above applies in particular to mental models as they also are usually described by relations and these relations can change, for example, due to learning. In this chapter it has been illustrated how self-modeling networks can be applied as useful means to design computational models for mental model handling. This does not only concern the use of a mental model and the adaptation of it as indicated above, but also control over the adaptation. In this way a three-level self-modeling network architecture is obtained that fits very well to the global cognitive architecture found in Van Ments and Treur (2021); see also Van Ments and Treur (2022). As for self-modeling networks a dedicated software environment is available to easily simulate them, this can be used to relate this cognitive architecture to a computational network model by which simulation experiments can be performed.

To analyse the scope of applicability of this network-oriented modeling approach based on self-modeling networks, following Ashby (1960) in Treur (2017), Sect. 2.3.1 it has been shown that any (state-determined) dynamical system as defined in Ashby (1960) and also used in Port and van Gelder (1995) can be described by a set of first-order differential equations, and conversely; see also Treur (2021a, d). Moreover, in Treur (2017), Sect. 2.3.2 it has also been shown how any set of first-order differential equations can be (re)modeled by a network model. These methods can also be applied to adaptive processes: any description of an adaptation process by a dynamical system or by first-order differential equations can be rewritten as a self-model in a self-modeling network. This has been used in Treur 2021a) to show that any adaptive dynamical system can be modeled as a self-modeling network; see also Treur (2022d).

Finally, analysis of stationary points and equilibria for self-modeling network models has been addressed in Treur (2016, Chap 12, 2020b, Chap. 11–14, 2021a, 2022b). Validation and parameter tuning has been addressed (Treur 2016, Chap. 14, 2022c).

References

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

Aizenman, C.D., Linden, D.J.: Rapid, synaptically driven increases in the intrinsic excitability of cerebellar deep nuclear neurons. Nat. Neurosci. **3**, 109–111 (2000)

Anten, J., Earle, J., Treur, J.: An Adaptive computational network model for strange loops in political evolution in society. In; Proceedings of the 20th International Conference on Computational Science, ICCS'20, vol. 2, pp. 604–617. Lecture Notes in Computer Science, vol. 12138. Springer (2020)

Ashby, W.R.: Design for a Brain, 2nd extended edn. Chapman and Hall, London. First edition, 1952 (1960)

Bowen, K.A., Kowalski, R.: Amalgamating language and meta-language in logic programming. In: Clark, K., Tarnlund, E. (eds.) Logic Programming, pp. 153–172. Academic Press, New York (1982)

Carley, K.M.: Inhibiting adaptation. In Proceedings of the 2002 Command and Control Research and Technology Symposium, pp. 1–10. Naval Postgraduate School, Monterey, CA

Carley, K.M.: Destabilization of covert networks. Comput. Math. Organiz. Theor. **12**, 51–66 (2006)

Chandra, N., Barkai, E.: A non-synaptic mechanism of complex learning: modulation of intrinsic neuronal excitability. Neurobiol. Learn. Memory **154**, 30–36 (2018)

Daoudal, G., Debanne, D.: Long-term plasticity of intrinsic excitability: learning rules and mechanisms. Learn. Memory **10**, 456–465 (2003)

Debanne, D., Inglebert, Y., Russier, M.: Plasticity of intrinsic neuronal excitability. Curr. Opin. Neurobiol. **54**, 73–82 (2019)

Demers, F.N., Malenfant, J.: Reflection in logic, functional and objectoriented programming: a Short Comparative Study. In: IJCAI'95 Workshop on Reflection and Meta-Level Architecture and their Application in AI, pp. 29–38 (1995)

Fessler, D.M.T., Clark, J.A., Clint, E.K.: Evolutionary psychology and evolutionary anthropology. In: The Handbook of Evolutionary Psychology, D.M. Buss edn., pp. 1029–1046. Wiley (2015)

Fessler, D.M.T., Eng, S.J., Navarrete, C.D.: Elevated disgust sensitivity in the first trimester of pregnancy: evidence supporting the compensatory prophylaxis hypothesis. Evol. Hum. Behav. **26**(4), 344–351 (2005)

Garcia, R.: Stress, metaplasticity, and antidepressants. Curr. Mol. Med. **2**, 629–638 (2002)

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

Hofstadter, D.R.: Gödel, Escher, Bach. Basic Books, New York (1979)

Keysers, C., Gazzola, V.: Hebbian learning and predictive mirror neurons for actions, sensations and emotions. Philos. Trans. r. Soc. Lond. B Biol. Sci. **369**, 20130175 (2014)

Levy, D.A., Nail, P.R.: Contagion: a theoretical and empirical review and reconceptualization. Genet. Soc. Gen. Psychol. Monogr. **119**(2), 233–284 (1993)

Lisman, J., Cooper, K., Sehgal, M., Silva, A.J.: Memory formation depends on both synapse-specific modifications of synaptic strength and cell-specific increases in excitability. Nat. Neurosci. **21**, 309–314 (2018)

Magerl, W., Hansen, N., Treede, R.D., Klein, T.: The human pain system exhibits higher-order plasticity (metaplasticity). Neurobiol. Learn. Memory **154**, 112–120 (2018)

McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. Annu. Rev. Sociol. **27**, 415–444 (2001)

Mooij, J.M., Janzing, D., Schölkopf, B.: From differential equations to structural causal models: the deterministic case. In: Nicholson, A., Smyth, P. (eds.) Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI-13), pp. 440--448. AUAI Press (2013)

Pearl, J.: Causality. Cambridge University Press (2000)

Pearson, M., Steglich, C., Snijders, T.: Homophily and assimilation among sport-active adolescent substance users. Connections **27**(1), 47–63 (2006)

Port, R.F., Van Gelder, T.: Mind as Motion: Explorations in the Dynamics of Cognition. MIT Press, Cambridge, MA (1995)
Robinson, B.L., Harper, N.S., McAlpine, D.: Meta-adaptation in the auditory midbrain under cortical influence. Nat. Commun. **7**, 13442 (2016)
Rojas, R.: Neural Networks. Springer, Berlin (1996)
Sharpanskykh, A., Treur, J.: Modelling and analysis of social contagion in dynamic networks. Neurocomputing **146**, 140–150 (2014)
Shatz, C.J.: The developing brain. Sci. Am. **267**, 60–67 (1992). https://doi.org/10.1038/scientificamerican0992-60
Sjöström, P.J., Rancz, E.A., Roth, A., Hausser, M.: Dendritic excitability and synaptic Plasticity. Physiol Rev **88**, 769–840 (2008)
Sterling, L., Shapiro, E.: The Art of Prolog, Chap. 17, pp. 319–356. MIT Press (1996)
Sterling, L., Beer, R.: Metainterpreters for expert system construction. J. Log. Program. **6**, 163–178 (1989)
Titley, H.K., Brunel, N., Hansel, C.: Toward a neurocentric view of learning. Neuron **95**, 19–32 (2017)
Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer (2016)
Treur, J.: On the applicability of network-oriented modeling based on temporal-causal networks: why network models do not just model networks. J Inf Telecommun **1**(1), 23–40 (2017)
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, ComplexNetworks'18, vol. 1. Studies in Computational Intelligence, vol. 812, pp. 635–651. Springer (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, Cham, Switzerland (2020b)
Treur, J.: On the Dynamics and Adaptivity of Mental Processes: Relating Adaptive Dynamical Systems and Self-Modeling Network Models by Mathematical Analysis. Cognitive Systems Research, vol. 70, pp. 93–100 (2021a)
Treur, J.: Equilibrium Analysis of within-network dynamics: from linear to nonlinear aggregation. 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. 94–110. Springer (2021b)
Treur, J.: With a Little help: a modeling environment for self-modeling network models. In: Treur, J., van Ments, L. (eds.) Mental Models and their Dynamics, Adaptation and Control: a Self-Modeling Network Modeling Approach, Chap. 17. Springer, Switzerland (this volume) (2022a)
Treur, J.: Where is this leading me: stationary point and equilibrium analysis of self-modeling network models. In: Treur, J., van Ments, L. (eds.) Mental Models and their Dynamics, Adaptation and Control: a Self-Modeling Network Modeling Approach, Chap. 18. Springer, Switzerland (this volume) (2022b)
Treur J.: Does this suit me: validation of self-modeling network models by parameter tuning. In: Treur, J., van Ments, L. (eds.) Mental Models and their Dynamics, Adaptation and Control: A Self-Modeling Network Modeling Approach, Chap. 19. Springer, Switzerland (this volume) (2022c)
Treur, J.: How far do self-modeling network models reach: relating them to adaptive dynamical systems. In: Treur, J., van Ments, L. (eds.) Mental Models and their Dynamics, Adaptation and Control: A Self-Modeling Network Modeling Approach, Chap 20. Springer, Switzerland (this volume) (2022d)
Treur, J.: How the brain creates emergent information by the development of mental models: an analysis from the perspective of temporal factorisation and criterial causation. In: Treur, J., van Ments, L. (eds.) Mental Models and their Dynamics, Adaptation and Control: a Self-Modeling Network Modeling Approach, Chap 16. Springer, Switzerland (this volume) (2022e)

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.: Dynamics, adaptation and control for mental models: a cognitive architecture. In: Treur, J., van Ments, L. (eds.) Mental Models and their Dynamics, Adaptation and Control: a Self-Modeling Network Modeling Approach, Chap. 1. Springer, Switzerland (this volume) (2022)

Weyhrauch, R.W.: Prolegomena to a theory of mechanized formal reasoning. Artif. Intell. **13**, 133–170 (1980)

Wright, S.: Correlation and causation. J. Agric. Res. **20**, 557–585 (1921)

Zhang, W., Linden, D.J.: The other side of the engram: experience-driven changes in neuronal intrinsic excitability. Nat. Rev. Neurosci. **4**, 885–900 (2003)