Modeling higher-order network adaptation by multilevel network reification

Treur, Jan

published in
Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models

2020

DOI (link to publisher)
10.1007/978-3-030-31445-3_4

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. (2020). Modeling higher-order network adaptation by multilevel network reification. In J. Treur (Ed.), *Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models* (pp. 99-119). (Studies in Systems, Decision and Control; Vol. 251). Springer International Publishing AG. https://doi.org/10.1007/978-3-030-31445-3_4

Chapter 4
Modeling Higher-Order Network Adaptation by Multilevel Network Reification

Abstract In network models for real-world domains, often some form of network adaptation has to be incorporated, based on certain network adaptation principles. In some cases, also higher-order adaptation occurs: the adaptation principles themselves also change over time. To model such multilevel adaptation processes, it is useful to have some generic architecture. Such an architecture should describe and distinguish the dynamics within the network (base level), but also the dynamics of the network itself by certain adaptation principles (first-order adaptation), and also the adaptation of these adaptation principles (second-order adaptation), and maybe still more levels of higher-order adaptation. This chapter introduces a multilevel network architecture for this, based on the notion of network reification. Reification of a network occurs when a base network is extended by adding explicit reification states representing the characteristics of the structure of the base network (Connectivity, Aggregation, and Timing). In Chap. 3, it was shown how this construction can be used to explicitly represent network adaptation principles within a network. In the current chapter, it is discussed how, when the reified network is itself also reified, also second-order adaptation principles can be explicitly represented. For the multilevel network reification construction introduced here, it is shown how it can be used to model plasticity and metaplasticity as known from Cognitive Neuroscience. Here, plasticity describes how connections between neurons change over time, for example, based on a first-order adaptation principle for Hebbian learning, and metaplasticity describes second-order adaptation principles determining how the extent of plasticity is affected by certain circumstances; for example, under which circumstances plasticity will be accelerated or decelerated.

4.1 Introduction

Within the complex dynamical systems area, adaptive behaviour is an interesting and quite relevant challenge, addressed in various ways; see, for example, Helbing et al. (2015), Perc and Szolnoki (2010). In particular for network-oriented dynamic modeling approaches, network models for real-world domains often show some

J. Treur, *Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models*, Studies in Systems, Decision and Control 251, https://doi.org/10.1007/978-3-030-31445-3_4

form of network adaptation based on certain network adaptation principles. Such principles describe how certain characteristics of the network structure change over time, for example, the connection weights in Mental Networks with Hebbian learning (Hebb 1949) or in Social Networks with bonding based on homophily; e.g., Byrne (1986), McPherson et al. (2001), Pearson et al. (2006), Sharpanskykh and Treur 2014). Sometimes also higher-order adaptation occurs in the sense that the adaptation principles for a network themselves also change over time. For example, *plasticity* in Mental Networks as described, for example, by Hebbian learning is not a constant feature, but usually varies over time, according to what in Cognitive Neuroscience has been called *metaplasticity*; e.g., Abraham and Bear (1996), Magerl et al. (2018), Parsons (2018), Schmidt et al. (2013), Sehgal et al. (2013), Zelcer et al. (2006). For more examples of processes which are adaptive of different orders, see Chap. 1, Sects. 1.2 and 1.3.

To model such multilevel network adaptation processes in a principled manner it is useful to have some generic architecture. Such architecture should be able to distinguish and describe:

(1) the dynamics within the base network
(2) the dynamics of the base network structure by network adaptation principles (first-order adaptation)
(3) the adaptation of these adaptation principles (second-order adaptation)
(4) interactions between the levels
(5) and maybe still more levels of higher-order adaptation.

In the current chapter, it is shown how such distinctions indeed can be made within a Network-Oriented Modeling framework using the notion of *reified network architecture*.

As also described in Chap. 3, reification is known from different scientific areas. According to the Merriam-Webster and Oxford dictionaries, it literally means representing something abstract as a material or concrete thing, or making something abstract more concrete or real. Reification offers substantial advantages in modeling and programming languages, as shown for other areas of AI and Computer Science; e.g., Bowen and Kowalski (1982), Demers and Malenfant (1995), Galton (2006), Hofstadter (1979), Sterling and Shapiro (1996), Sterling and Beer (1989), Weyhrauch (1980). Modeling adaptivity and enhanced expressive power are some of these advantages. Network reification has similar advantages. In Chap. 3, it has been shown how network reification can be used to explicitly represent adaptation principles for networks in a transparent and unified manner. Examples of such adaptation principles are, among others, principles for Hebbian learning (to model plasticity in the brain) and for bonding based on homophily (to model adaptive social networks). Using network reification, adaptive Mental Networks and adaptive Social Networks can be addressed well, as shown by many examples in Chap. 3.

Including reification states for the characteristics of the base network structure (connection weights, speed factors, and combination functions and their parameters) in the extended network is one step. A next step is defining proper temporal-causal relations for them and relating them to the other states. Then a reified network is obtained that explicitly represents the characteristics of the base network, and, moreover, how this base network evolves over time, based on adaptation principles that change the causal network relations. In Chap. 3, it was shown how this can be used for a variety of adaptation principles known from Cognitive Neuroscience and Social Science.

Such reified adaptive networks form again a basic network structure defined by certain characteristics, such as learning rate or adaptation speed of connections. Adaptation principles may be adaptive themselves too, according to certain second-order adaptation principles. From recent literature, it has become clear that in real-world domains often these characteristics can still change over time, for example, in the case of metaplasticity. The notion of metaplasticity as already mentioned above, has become a focus of study in empirical literature such as Arnold et al. (2015), Chandra and Barkai (2018), Daimon et al. (2017), Magerl et al. (2018), Parsons (2018), Robinson et al. (2016), Sehgal et al. (2013), Schmidt et al. (2013), Zelcer et al. (2006). This area of higher-order adaptivity is a next challenge to be addressed. To this end, in the current chapter a construction of multilevel reification is illustrated for the Network-Oriented Modeling approach based on temporal-causal networks (Treur 2016, 2019). By an appropriate number of iterations, this multilevel reification construction introduced here can be used to model higher-order adaptivity of any level. The multilevel reification architecture has been implemented by the author in Matlab, as will be discussed in Chap. 9. The homophily context is used in this chapter as the first application of a second-order adaptive Social Network for the Social Science area, and the context of plasticity and metaplasticity as a second application for a second-order adaptive Mental Network in the Cognitive Neuroscience area.

In this chapter in Sect. 4.2, the Network-Oriented Modeling approach based on temporal-causal networks is briefly summarized. Next, in Sect. 4.3 the network reification concept is summarized, and in Sect. 4.4 the more general multilevel network reification construction is introduced. Moreover, it is shown how it can model examples of second-order network adaptivity. This is illustrated by a second-order adaptive network for plasticity and metaplasticity from Cognitive Neuroscience. In Sect. 4.5 example simulations for this multilevel network reification example are presented. Section 4.6 discusses the added complexity in a multilevel reification architecture. Section 4.7 is a discussion.

4.2 Structure and Dynamics of Temporal-Causal Networks

The network structure of a temporal-causal network model can be described conceptually by a graph with nodes and directed connections and a number of labels for such a graph for connectivity, aggregation, and timing:

(a) **Connectivity**

- In terms of connection weights $\boldsymbol{\omega}_{X,Y}$; see Table 4.1, upper part, and Fig. 4.1 for an example of a basic fragment of a network with states X_1, X_2 and Y, and labels $\boldsymbol{\omega}_{X_1,Y}$, $\boldsymbol{\omega}_{X_2,Y}$ for connection weights.

(b) **Aggregation**

- In terms of combination functions $\mathbf{c}_Y(..)$; a library with a number of standard combination functions is available, but also new functions can be added. Such functions are just declarative mathematical objects that relate real numbers to real numbers without any procedural elements involved, i.e., **c**: $\mathrm{R}^k \rightarrow \mathrm{R}$.

(c) **Timing**

- In terms of speed factors $\boldsymbol{\eta}_Y$.

In the lower part of Table 4.1, it is shown how the numerical representation of the network's dynamics is defined in terms of the above labels; see also

Table 4.1 Conceptual and numerical representation of a temporal-causal network structure

Concepts	Notation	Explanation
States and connections	X, Y, $X \rightarrow Y$	Describes the nodes and links of a network structure (e.g., in graphical or matrix format)
Connection weight	$\boldsymbol{\omega}_{X,Y}$	The *connection weight* $\boldsymbol{\omega}_{X,Y} \in [-1, 1]$ represents the strength of the causal impact of state X on state Y with $X \rightarrow Y$
Aggregating multiple impacts	$\mathbf{c}_Y(..)$	For each state Y a *combination function* $\mathbf{c}_Y(..)$ is chosen to combine the causal impacts of other states on state Y
Timing of the causal effect	$\boldsymbol{\eta}_Y$	For each state Y a *speed factor* $\boldsymbol{\eta}_Y \geq 0$ is used to represent how fast a state is changing upon causal impact

Concepts	Numerical representation	Explanation
State values over time t	$Y(t)$	At each time point t each state Y in the model has a real number value in [0, 1]
Single causal impact	$\mathbf{impact}_{X,Y}(t)$ $= \boldsymbol{\omega}_{X,Y} X(t)$	At t state X with connection to state Y has an impact on Y, using weight $\boldsymbol{\omega}_{X,Y}$
Aggregating multiple impacts	$\mathbf{aggimpact}_Y(t)$ $= \mathbf{c}_Y(\mathbf{impact}_{X_1,Y}(t), \ldots, \mathbf{impact}_{X_k,Y}(t))$ $= \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y} X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y} X_k(t))$	The aggregated impact of multiple states X_i on Y at t, is determined using combination function $\mathbf{c}_Y(..)$
Timing of the causal effect	$Y(t + \Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)]\Delta t =$ $Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y} X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y} X_k(t)) - Y(t)]\Delta t$	The causal impact on Y is exerted over time gradually, using speed factor $\boldsymbol{\eta}_Y$

Adopted from Treur (2019)

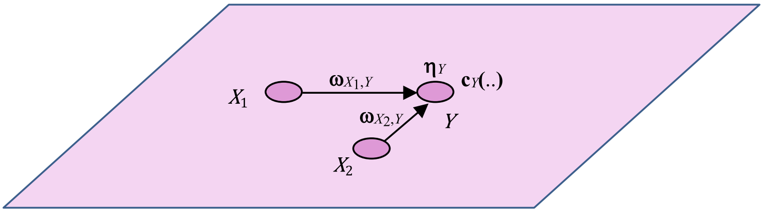


Fig. 4.1 Fragment of a temporal-causal network structure in a labeled graph representation. The basic elements are nodes and their connections, with for each node Y a speed factor $\boldsymbol{\eta}_Y$ and a combination function $\mathbf{c}_Y(..)$, and for each connection from X to Y a connection weight $\boldsymbol{\omega}_{X,Y}$

Treur (2016), Chap. 2. Here X_1, …, X_k are the states from which state Y receives incoming connections. These formulas in the last row in Table 4.1 define the detailed dynamic semantics of a temporal-causal network. They can be used for mathematical analysis and for simulation, and can be written in differential equation format as follows:

$$\mathbf{d}Y(t)/\mathbf{d}t = \boldsymbol{\eta}_Y\left[\mathbf{c}_Y\left(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)\right) - Y(t)\right] \tag{4.1}$$

Examples of combination functions are the *identity* **id**(.) for states with impact from only one other state, the *scaled sum* $\mathbf{ssum}_\lambda$(.) with scaling factor $\boldsymbol{\lambda}$, the scaled minimum function $\mathbf{smin}_\lambda$**(..)** and maximum function $\mathbf{smax}_\lambda$**(..)**, and the *advanced logistic sum* combination function $\mathbf{alogistic}_{\sigma,\tau}$(..) with steepness $\boldsymbol{\sigma}$ and threshold $\boldsymbol{\tau}$; see also Treur (2016, Chap. 2, Table 2.10):

$$\begin{aligned}
&\mathbf{id}(V) = V\\
&\mathbf{ssum}_\lambda(V_1, \ldots, V_k) = \frac{V_1 + \cdots + V_k}{\boldsymbol{\lambda}}\\
&\mathbf{smin}_\lambda(V_1, \ldots, V_k) = \frac{\min(V_1, \ldots, V_k)}{\boldsymbol{\lambda}}\\
&\mathbf{smax}_\lambda(V_1, \ldots, V_k) = \frac{\max(V_1, \ldots, V_k)}{\boldsymbol{\lambda}}\\
&\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = \left[\frac{1}{1+\mathrm{e}^{-\boldsymbol{\sigma}(V_1+\cdots+V_k-\boldsymbol{\tau})}} - \frac{1}{1+\mathrm{e}^{\boldsymbol{\sigma}\boldsymbol{\tau}}}\right](1+\mathrm{e}^{-\boldsymbol{\sigma}\boldsymbol{\tau}})
\end{aligned} \tag{4.2}$$

Note that for basic combination functions, specific parameters are considered. Examples are the scaling factor $\boldsymbol{\lambda}$, the steepness $\boldsymbol{\sigma}$, and the threshold $\boldsymbol{\tau}$ above. These parameters can also be written as arguments in the function, for example, $\mathbf{alogistic}(\boldsymbol{\sigma}, \boldsymbol{\tau}, V_1, \ldots, V_k)$, or in lists as $\mathbf{alogistic}([\boldsymbol{\sigma}, \boldsymbol{\tau}], [V_1, \ldots, V_k])$; this actually is how they are represented in the software environment.

Examples of combination functions applied in particular for reification states in reified adaptive networks (introduced in Chap. 3, Sect. 3.6.1) are the following

- Hebbian learning (see Sect. 4.4)

$$\mathbf{hebb}_{\boldsymbol{\mu}}(V_1, V_2, W) = V_1 V_2(1 - W) + \boldsymbol{\mu} W \tag{4.3}$$

  Here V_1, V_2 refer to the activation levels of two connected states and W to their connection weight; $\boldsymbol{\mu}$ is a parameter for the persistence factor.

- Simple linear homophily (see Chap. 6)

$$\mathbf{slhomo}_{\boldsymbol{\alpha},\boldsymbol{\tau}}(V_1, V_2, W) = W + \boldsymbol{\alpha} W(1 - W)(\boldsymbol{\tau} - |V_1 - V_2|) \tag{4.4}$$

  Here V_1, V_2 refer to the activation levels of the states of two connected persons and W to their connection weight; $\boldsymbol{\tau}$ is a tipping point parameter and $\boldsymbol{\alpha}$ is a homophily modulation parameter. This is applied to model bonding based on homophily; see Chap. 6.

The set of already available combination functions forms a *combination function library* (with at the time of writing 35 functions), which can be chosen as basic combination functions during the design of a network model. These functions are declarative mathematical functions relating real numbers to real numbers without any procedural or process elements.

4.3 Addressing Network Adaptation by Network Reification

Recall from Chap. 3 that network reification is a construction principle by which a base network is extended by extra states that represent the base network's structure. This construction principle is briefly summarized here.

4.3.1 Extending the Network by Reification States

The added states represent specific characteristics of the network structure. They are what are called *reification states* for these characteristics, in other words, the characteristics are reified by these states. More specifically, these reification states represent the labels for *connection weights*, *combination functions*, and *speed factors* shown in Table 4.1. For connection weights $\boldsymbol{\omega}_{X_i,Y}$ for the incoming connections from states X_i to state Y and speed factors $\boldsymbol{\eta}_Y$ for state Y, their reification states $\mathbf{W}_{X_i,Y}$ and $\mathbf{H}_Y$ represent the value of them, and the vector of reification states $\mathbf{C}_Y$ = ($\mathbf{C}_{1,Y}$, $\mathbf{C}_{2,Y}$, ….) represents the weights for the chosen basic combination functions for state Y; moreover, reification states $\mathbf{P}_{i,j,Y}$ represent the adaptive parameters of combination functions. In Fig. 4.2, the reification states are depicted in the upper (blue) plane, whereas the states of the base network are in the lower (pink) plane.

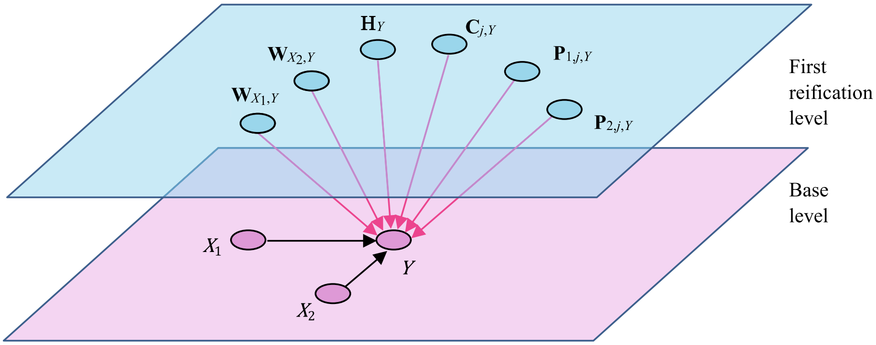


Fig. 4.2 Network reification for a temporal-causal network with in the upper, blue plane reification states $\mathbf{H}_Y$ for the speed factor of base state *Y*, $\mathbf{W}_{X_i,Y}$ for the weights of the connections from X_i to *Y*, $\mathbf{C}_{j,Y}$ for the basic combination function weights of *Y*, and $\mathbf{P}_{i,j,Y}$ for the parameter values of these basic combination functions. The downward connections from these reification states to state *Y* in the base network (in the lower, pink plane) indicate their special causal effect on *Y*

Within the reified network causal relations for the reification states for characteristics of a network can be defined: incoming connections affecting them, and outgoing connections from them to the related base network states. Such connections are the way in which adaptation principles are explicitly represented within the (reified) network; see also the many examples in Chap. 3. The downward pink arrows in Fig. 4.2 define how the reification states contribute their special effect to an aggregated impact on the related base network state.

These downward connections in Fig. 4.2 and the combination functions for the base states are defined in a generic manner. The general pattern is that the reification state roles $\mathbf{W}_{X_i,Y}$, $\mathbf{H}_Y$ and $\mathbf{C}_{\ j,Y}$ and $\mathbf{P}_{i,j,Y}$ for connection weights, speed factors, combination function weights, and combination function parameter values have a role-specific causal connection to state *Y* in the base network, as they all affect *Y* in their own role-dependent way. All depicted downward connections automatically get weight 1, so that there is a one-to-one correspondence between the base characteristic and its reification, and in the reified network the speed factors of the base states are set at 1 too. For the base states, new combination functions are needed that will be defined below (see also Chap. 3, Sect. 3.5). The different components

$$\mathbf{C}_{1,Y}, \mathbf{C}_{2,Y}, \ldots$$

for $\mathbf{C}_Y$ are explained as follows. During modeling a sequence of basic combination functions

$$\text{bcf}_1(..), \ldots, \text{bcf}_m(..)$$

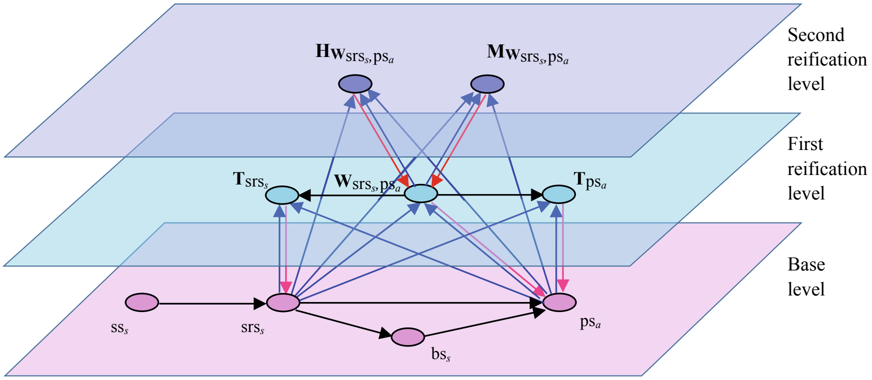
is chosen from the function library discussed in Sect. 4.2 (in which also new functions can be added), to be used in the specific application addressed; for more details, see Chap. 9. For example,

$$\begin{aligned}
\mathrm{bcf}_1(..) &= \mathbf{sum}(..),\\
\mathrm{bcf}_2(..) &= \mathbf{ssum}_{\boldsymbol{\lambda}}(..)\\
\mathrm{bcf}_3(..) &= \mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(..)
\end{aligned}$$

For a given state Y, each of these selected basic combination functions $\mathrm{bcf}_j(..)$ gets a weight $\boldsymbol{\gamma}_{j,Y}$ assigned which is represented by reification state $\mathbf{C}_{j,Y}$. Moreover, each basic combination function $\mathrm{bcf}_j(..)$ is assumed to have two parameters for each state: $\boldsymbol{\pi}_{1,j,Y}$, $\boldsymbol{\pi}_{2,j,Y}$. These *combination function parameters* $\boldsymbol{\pi}_{1,1,Y}$, $\boldsymbol{\pi}_{2,1,Y}$, …, $\boldsymbol{\pi}_{1,m,Y}$, $\boldsymbol{\pi}_{2,m,Y}$ in the m selected combination functions can also be explicitly represented by *parameter reification states*

$$\mathbf{P}_{1,1,Y}, \mathbf{P}_{2,1,Y}, \ldots, \mathbf{P}_{1,m,Y}, \mathbf{P}_{2,m,Y}$$

so that they also can become adaptive. Their values are considered as the first arguments in $\mathrm{bcf}_j(..)$, and also included as arguments in $\mathbf{c}_Y(...)$. Note that for applications, often more informative names are used for these parameters $\boldsymbol{\pi}_{i,j,Y}$ and their reification states $\mathbf{P}_{i,j,Y}$; for example, reification state $\mathbf{H}_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}$ in Fig. 4.3 for the reified speed factor of the connection adaptation, and reification state $\mathbf{M}_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}$ for the persistence parameter $\boldsymbol{\mu}$ for the Hebbian learning; this will be explained in more detail in Sect. 4.4.



$\mathbf{H}_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}$
$\mathbf{M}_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}$
Second reification level
$\mathbf{T}_{\mathrm{srs}_s}$
$\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$
$\mathbf{T}_{\mathrm{ps}_a}$
First reification level
Base level
ss_s
srs_s
ps_a
bs_s

Fig. 4.3 Overview of the reified network architecture for plasticity and metaplasticity with base level (lower plane, pink), first reification level (middle plane, blue) and second reification level (upper plane, purple), and upward causal connections (blue) and downward causal connections (red) defining interlevel relations. The downward causal connections from the two **T**-states affect the excitability of the (presynaptic and postsynaptic) states srs_s and ps_a. The downward causal connections from the **H**-state and **M**-state affect the adaptation speed and the persistence factor of the connection weight reification state $\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$

So, in the base network for each state Y *combination function weights* $\boldsymbol{\gamma}$ are assumed: numbers $\gamma_{1,Y}$, $\gamma_{2,Y}$, … ≥ 0 that may change over time such that the combination function $\mathbf{c}_Y(.)$ for Y is expressed by:

$$\begin{aligned}&\mathbf{c}_Y(t, \boldsymbol{\pi}_{1,1,Y}, \boldsymbol{\pi}_{2,1,Y},..., \boldsymbol{\pi}_{1,m,Y}, \boldsymbol{\pi}_{2,m,Y}, V_1, \ldots, V_k)\\ &\quad = \frac{\gamma_{1,Y}(t)\mathrm{bcf}_1\left(\boldsymbol{\pi}_{1,1,Y}, \boldsymbol{\pi}_{2,1,Y}, V_1, \ldots, V_k\right) + \cdots + \gamma_{m,Y}(t)\,\mathrm{bcf}_m(\boldsymbol{\pi}_{1,m,Y}, \boldsymbol{\pi}_{2,m,Y}, V_1, \ldots, V_k)}{\gamma_{1,Y}(t) + \cdots + \gamma_{m,Y}(t)}\end{aligned} \tag{4.5}$$

The basic combination function weights $\gamma_{i,Y}(..)$ are represented by the reification states $\mathbf{C}_{i,Y}$ for Y. This describes that for Y a weighted average of basic combination functions is used. Note that, if exactly one of the $\mathbf{C}_{i,Y}(t)$ is nonzero, just one basic combination function is selected for $\mathbf{c}_Y(.)$. This approach makes it possible, for example, to gradually switch from one combination function $\mathrm{bcf}_i(..)$ to another one $\mathrm{bcf}_j(..)$ over time by decreasing the value of $\mathbf{C}_{i,Y}(t)$ and increasing the value of $\mathbf{C}_{j,Y}(t)$.

4.3.2 The Universal Combination Function and Universal Difference Equation for Reified Networks

In Chap. 3, Sect. 3.5 a *universal combination function* $\mathbf{c}^*_Y(..)$ has been found for any base state Y in the reified network. In cases of full reification, it has no parameters for network characteristics, only variables. Therefore it can be used in the same form for every base state as shown below; for a more detailed derivation, also see Chap. 10:

$$\begin{aligned}&\mathbf{c}^*_Y(H, C_1, \ldots, C_m, P_{1,1}, P_{2,1}, \ldots, P_{1,m}, P_{2,m}, W_1, \ldots, W_k, V_1, \ldots, V_k, V)\\ &= H\frac{C_1\,\mathrm{bcf}_1\left(P_{1,1,Y}, P_{2,1,Y}, W_1V_1, \ldots, W_kV_k\right) + \cdots + C_m\,\mathrm{bcf}_m(P_{1,m,Y}, P_{2,m,Y}, W_1V_1, \ldots, W_kV_k)}{C_1 + \cdots + C_m} + (1-H)V\\ &= H\left[\frac{C_1\,\mathrm{bcf}_1\left(P_{1,1,Y}, P_{2,1,Y}, W_1V_1, \ldots, W_kV_k\right) + \cdots + C_m\,\mathrm{bcf}_m(P_{1,m,Y}, P_{2,m,Y}, W_1V_1, \ldots, W_kV_k)}{C_1 + \cdots + C_m} - V\right] + V\end{aligned} \tag{4.6}$$

Here

- H refers to the speed factor reification $\mathbf{H}_Y(t)$
- C_i to the combination function weight reification $\mathbf{C}_{i,Y}(t)$
- $P_{i,j}$ to the parameter reification value $\mathbf{P}_{i,j,Y}(t)$ of parameter i = 1, 2 of basic combination function j = 1, …, m
- W_i to the connection weight reification $\mathbf{W}_{X_i,Y}(t)$
- V_i to the state value $X_i(t)$ of base state X_i
- V to the state value $Y(t)$ of base state Y.

This combination function $\mathbf{c}^*{}_Y(..)$ in (4.4) makes that the dynamics of any base state Y within the reified network is described by the following *universal difference equation* in temporal-causal network format:

$$\begin{aligned} Y(t+\Delta t) = Y(t) + \big[&\mathbf{c}^*_Y(\mathbf{H}_Y(t), \mathbf{C}_{1,Y}(t), \ldots, \mathbf{C}_{m,Y}(t),\\ &\mathbf{P}_{1,1,Y}(t), \mathbf{P}_{2,1,Y}(t), \ldots, \mathbf{P}_{1,m,Y}(t),\\ &\mathbf{P}_{2,m,Y}(t), \mathbf{W}_{X_1,Y}(t), \ldots, \mathbf{W}_{X_k,Y}(t),\\ &X_1(t), \ldots, X_k(t), Y(t)) - Y(t)\big]\Delta t \end{aligned} \tag{4.7}$$

For more details, see Chap. 3, Sect. 3.5 and Chap. 10.

Structures added by the reification process are not reified themselves. However, the structure of the reified network can also be reified as another step: providing what is then called *second-order reification*. In the next section it is explored how such second-order reification can be done and how it can be used to model second-order adaptation for adaptive first-order adaptation principles.

4.4 Using Multilevel Network Reification for Higher-Order Adaptive Network Models

In this section, the multilevel reification architecture is introduced that allows modeling of networks with arbitrary orders of adaptation. In this architecture, the base network has its own internal dynamics, but it also evolves through one or more adaptation principles (called *first-order adaptation principles*). Moreover, these first-order adaptation principles themselves can change based on other adaptation principles (called *second-order adaptation principles*). So the architecture offers n reification levels for an arbitrary n where on reification level i adaptation principles are defined for ith-order adaptation. In this chapter, it is shown how the reified temporal-causal network modeling approach can be used to model important developments in empirical science, in particular concerning plasticity and metaplasticity. These are important notions in state of the art research on Cognitive Neuroscience, introduced not from a computational modeling perspective but by purely empirical researchers to clarify what was found empirically. This section shows how these notions can be connected to the reified temporal-network modeling approach described in the current chapter. This particular example shows the essential elements but is kept relatively simple; it can easily be extended by adding more states and connections.

4.4.1 Using Multilevel Network Reification for Plasticity and Metaplasticity from Cognitive Neuroscience

Mental networks equipped with a Hebbian learning mechanism (Hebb 1949) are able to adapt connection weights over time and learn or form memories in this way. Within Neuroscience this is usually called *plasticity*. In some circumstances it is better to learn (and change) fast, but in other circumstances, it is better to stay stable and persist what has been learnt in the past. To control this, by humans a type of (higher-order) adaptation called *metaplasticity* is used. It has become an important focus of study in Cognitive Neuroscience. In literature such as Abraham and Bear (1996), Chandra and Barkai (2018), Magerl et al. (2018), Parsons (2018), Robinson et al. (2016), Sehgal et al. (2013), Schmidt et al. (2013), Sjöström et al. (2008) various studies are reported which show how the adaptation of synapses (as described, for example, by Hebbian learning), is modulated by suppressing the adaptation process or amplifying it. Among the reported factors affecting synaptic plasticity are stimulus exposure, activation, previous experiences, and stress, which can accelerate or decelerate learning, or induce temporarily enhanced excitability of neurons which in turn positively affects learning; e.g., Chandra and Barkai (2018), Oh et al. (2003).

The reified network modeling approach was applied to a case involving both plasticity and metaplasticity, acquired from the literature mentioned above. A network picture of the designed reified network model for plasticity and metaplasticity is shown in Fig. 4.3. Table 4.2 displays the explanations of the states. Section 4.4.2 shows the complete specification. Here the plasticity of the response connection from srs$_s$ to ps$_a$ is considered, modeled by Hebbian learning. Note that the two **T**-states and the **M**-state are combination function parameter states here, respectively for excitability threshold τ of srs$_s$ and ps$_a$ and for persistence parameter μ for the Hebbian learning of the connection from srs$_s$ to ps$_a$. The alternative path via the belief state bs$_s$ supports this learning by contributing to the activation of ps$_a$, thus relating to the original formulation in Hebb (1949):

Table 4.2 State names for the plasticity and metaplasticity model with their explanations

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	bs$_s$	Belief state for stimulus s	
X_4	ps$_a$	Preparation state for response a	
X_5	$\mathbf{W}_{\text{srs}_s,\text{ps}_a}$	Reified representation state for connection weight $\omega_{\text{srs}_s,\text{ps}_a}$	First reification level
X_6	$\mathbf{T}_{\text{srs}_s}$	Reified representation state for threshold parameter τ_{srs_s} of base state srs$_s$	
X_7	$\mathbf{T}_{\text{ps}_a}$	Reified representation state for threshold parameter τ_{ps_a} of base state ps$_a$	
X_8	$\mathbf{H}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$	Reified representation state for speed factor $\eta_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ for reified representation state $\mathbf{W}_{\text{srs}_s,\text{ps}_a}$	Second reification level
X_9	$\mathbf{M}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$	Reified representation state for persistence factor parameter $\mu_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ for reified representation state $\mathbf{W}_{\text{srs}_s,\text{ps}_a}$	

> 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)

In principle, this will start to work when the external stimulus s is sensed through sensor state ss_s. However, as discussed above, whether or not and to which extent learning actually takes place is controlled by a form of metaplasticity; this also relates to factors such as excitability characteristics of the involved states. To model metaplasticity, the model includes a second reification level with states $\mathbf{H}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ representing the speed of the learning (learning rate) of $\boldsymbol{\omega}_{\text{srs}_s,\text{ps}_a}$, and $\mathbf{M}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ representing the persistence $\boldsymbol{\mu}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ of the connection weight $\boldsymbol{\omega}_{\text{srs}_s,\text{ps}_a}$. They have dynamic values depending on the other states. For example, if at some point in time the value of $\mathbf{H}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ is 0, no learning will take place, and if $\mathbf{M}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ has value 0, no learnt effects will persist; the value of these second-order reification states depend on activation of the presynaptic and postsynaptic states srs_s and ps_a, also see Robinson et al. (2016):

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

Note that a double level subscript notation for second-order reification states such as $\mathbf{H}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ should be read as $\mathbf{H}_Y$ for a state Y at the first reification level, in this case, $Y = \mathbf{W}_{\text{srs}_s,\text{ps}_a}$. By substituting $\mathbf{W}_{\text{srs}_s,\text{ps}_a}$ for Y in $\mathbf{H}_Y$, this results in the double level subscript notation $\mathbf{H}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$; note that here for the sake of simplicity the subscripts in srs_s and ps_a are considered to be at the same subscript level as srs and ps. So, the subscript of $\mathbf{H}$ is $\mathbf{W}_{\text{srs}_s,\text{ps}_a}$ and this subscript itself has subscripts srs_s and ps_a; the notation should be interpreted as $\mathbf{H}_{(\mathbf{W}_{\text{srs}_s,\text{ps}_a})}$. In this way, the number of reification levels is reflected in the number of subscript levels. This applies to all states at the second reification level, so, for example, also to $\mathbf{M}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$. Up till now no cases of network adaptation of order higher than 2 have been addressed; however, see Chaps. 7 and 8 where more than 2 reification levels show up, and more subscript levels accordingly. From a modeling perspective there is nothing against adding a third reification level for the characteristics that define the second-order adaptation principles by the dynamics of the second-order reification states, for example, adding third-order reification states for their speed factors or their combination functions or the parameters of these functions.

To address dynamic levels of excitability of base states, first-order reification states $\mathbf{T}_{\text{srs}_s}$ and $\mathbf{T}_{\text{ps}_a}$ have been included that model the intrinsic excitability of the presynaptic and postsynaptic state srs_s and ps_a, respectively, by the value of the thresholds $\boldsymbol{\tau}_{\text{srs}_s}$ and $\boldsymbol{\tau}_{\text{ps}_a}$ of their logistic sum combination functions; also see Chandra and Barkai (2018):

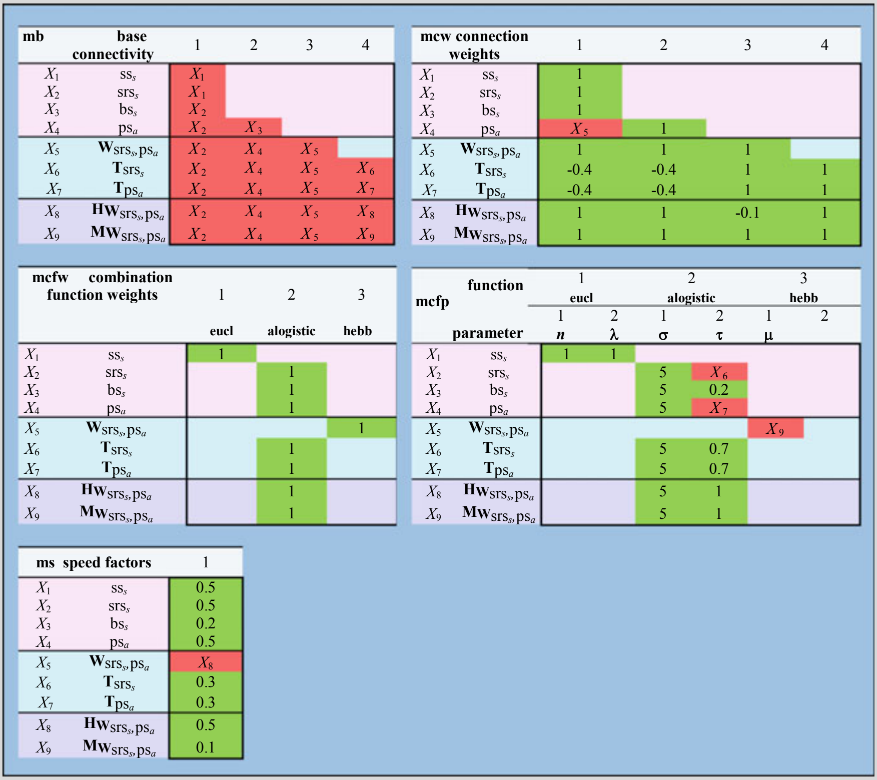
> Learning-related cellular changes can be divided into two general groups: modifications that occur at synapses and modifications in the intrinsic properties of the neurons. While it is commonly agreed that changes in strength of connections between neurons in the relevant networks underlie memory storage,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

For most of the states, the combination function used below is the $\mathbf{alogistic}_{\sigma,\tau}(..)$ function. The only exceptions are the sensor state ss_s which uses the Euclidean combination function $\mathbf{eucl}_{1,\lambda}(..)$ and $\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$ which uses the Hebbian combination function $\mathbf{hebb}_{\mu_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}}(..)$.

4.4.2 Role Matrices Covering Plasticity and Metaplasticity

The multilevel reified network model described in Sect. 4.4.1 by a conceptual graphical representation, is described in the current section by a conceptual role matrices representation. The role matrix **mb** specifies for this network model on each row for a given state which states at the same or a lower level have outgoing connections to that state. This plays the role of *base connectivity*. This matrix contains the information depicted in Fig. 4.3 by upward (blue) or leveled (black) arrows, and includes for each state a numbering of the incoming base connections (the 1–4 in the top row), and for some of the states a connection from the state itself. The latter applies to all (first- and second-order) reification states, as can be seen in **mb**. For example, in the third row, it is indicated that state X_3 (=bs_s) only has one incoming base connection, from state X_2 (=srs_s). As another example, the fifth row indicates that state X_5 (=$\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$) has incoming base connections from X_2 (=srs_s), X_4 (=ps_a), X_5 (=$\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$) 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 second column with more informative state names in each of the role matrices depicted in Box 4.1 is not part of the specification but has just been added for human understanding.

Box 4.1 Role matrices for the second-order reified network for plasticity and metaplasticity



mb	base connectivity	1	2	3	4
X_1	ss_s	X_1			
X_2	srs_s	X_1			
X_3	bs_s	X_2			
X_4	ps_a	X_2	X_3		
X_5	$\mathbf{W}_{srs_s,ps_a}$	X_2	X_4	X_5	
X_6	$\mathbf{T}_{srs_s}$	X_2	X_4	X_5	X_6
X_7	$\mathbf{T}_{ps_a}$	X_2	X_4	X_5	X_7
X_8	$\mathbf{Hw}_{srs_s,ps_a}$	X_2	X_4	X_5	X_8
X_9	$\mathbf{Mw}_{srs_s,ps_a}$	X_2	X_4	X_5	X_9

mcw	connection weights	1	2	3	4
X_1	ss_s	1			
X_2	srs_s	1			
X_3	bs_s	1			
X_4	ps_a	X_5	1		
X_5	$\mathbf{W}_{srs_s,ps_a}$	1	1	1	
X_6	$\mathbf{T}_{srs_s}$	-0.4	-0.4	1	1
X_7	$\mathbf{T}_{ps_a}$	-0.4	-0.4	1	1
X_8	$\mathbf{Hw}_{srs_s,ps_a}$	1	1	-0.1	1
X_9	$\mathbf{Mw}_{srs_s,ps_a}$	1	1	1	1

mcfw	combination function weights	1	2	3
		eucl	alogistic	hebb
X_1	ss_s	1		
X_2	srs_s		1	
X_3	bs_s		1	
X_4	ps_a		1	
X_5	$\mathbf{W}_{srs_s,ps_a}$			1
X_6	$\mathbf{T}_{srs_s}$		1	
X_7	$\mathbf{T}_{ps_a}$		1	
X_8	$\mathbf{Hw}_{srs_s,ps_a}$		1	
X_9	$\mathbf{Mw}_{srs_s,ps_a}$		1	

mcfp	function	1		2		3	
		eucl		alogistic		hebb	
		1	2	1	2	1	2
	parameter	n	λ	σ	τ	μ	
X_1	ss_s	1	1				
X_2	srs_s			5	X_6		
X_3	bs_s			5	0.2		
X_4	ps_a			5	X_7		
X_5	$\mathbf{W}_{srs_s,ps_a}$					X_9	
X_6	$\mathbf{T}_{srs_s}$			5	0.7		
X_7	$\mathbf{T}_{ps_a}$			5	0.7		
X_8	$\mathbf{Hw}_{srs_s,ps_a}$			5	1		
X_9	$\mathbf{Mw}_{srs_s,ps_a}$			5	1		

ms	speed factors	1
X_1	ss_s	0.5
X_2	srs_s	0.5
X_3	bs_s	0.2
X_4	ps_a	0.5
X_5	$\mathbf{W}_{srs_s,ps_a}$	X_8
X_6	$\mathbf{T}_{srs_s}$	0.3
X_7	$\mathbf{T}_{ps_a}$	0.3
X_8	$\mathbf{Hw}_{srs_s,ps_a}$	0.5
X_9	$\mathbf{Mw}_{srs_s,ps_a}$	0.1

In a similar way the four types of role matrices for *non-base roles* (showing either values or reification states to play that role; in the later case the downward arrows in Fig. 4.3 are defined here), were defined; see Box 4.1: role matrices **mcw** for connection weights, **ms** for speed factors, **mcfw** for combination function weights, and **mcfp** for combination function parameters. As before, within each role matrix, cell entries in red indicate a reference to the name of another state that as a form of reification represents in a dynamic manner an adaptive network characteristic, while entries indicating in green indicate fixed values for nonadaptive characteristics. The red cells represent the downward causal connections from the reification states in pictures as shown in Fig. 4.3, 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 actually defines the roles of the reification states so that there is no need to computationally use information from the names **W**, **H**, **C**, **P** of them; they may have any own given names.

For example, in Box 4.1 the name X_5 in the red cell row-column (4, 1) in role matrix **mcw** indicates that the value of the connection weight from srs_s to ps_a is the value of state X_5. In contrast, the 1 in green cell (5, 1) of **mcw** indicates the static value of the connection weight from X_2 (=srs_s) to X_5 (=$\mathbf{W}_{\text{srs}_s,\text{ps}_a}$). Similarly, role matrix **ms** indicates (in red) that X_8 represents the adaptive speed factor of X_5, and (in green) that the speed factors of all other states have fixed values.

For a given application a limited fixed sequence of combination functions is specified by **mcf** = [1 2 3], where the numbers 1, 2, 3 refer to the numbering in the function library which currently contains 35 combination functions, the first three being $\mathbf{eucl}_{n,\lambda}(..)$, $\mathbf{alogistic}_{\sigma,\tau}(..)$, $\mathbf{hebb}_{\mu}(..)$. In Box 4.1 the role matrices **mcfw** and **mcfp** are shown for combination function weights and parameters, respectively. Here the 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.

4.5 Simulation for a Second-Order Reified Network Model for Plasticity and Metaplasticity

Following what is reported in the literature on metaplasticity, a number of simulation experiments have been performed. In particular, a scenario is shown here in which the focus was on the effect of activation of the postsynaptic state ps_a on plasticity; the effect of the presynaptic state srs_s on reification states was blocked (weights of upward links from srs_s were set 0). In Fig. 4.4 the simulation results are shown. For settings, see the specification in Sect. 4.4.2, Box 4.1. The upper graph shows the activation levels of the base states and how the weight of the connection from srs_s to ps_a is learnt. Here the activation levels and the exact shape of the learning curve also depend on controlling factors shown in the lower graph in Fig. 4.4. As can be seen there, following exposure to stimulus s, the threshold values $\mathbf{T}_{\text{srs}_s}$ and $\mathbf{T}_{\text{ps}_a}$ for the activation of srs_s and ps_a are decreasing to low levels. This substantially increases the excitability of srs_s and ps_a conform (Chandra and Barkai 2018) and therefore gives a boost to the activation levels of these base states, which in turn strengthens the Hebbian learning. Also, it is shown that following exposure to stimulus s the learning speed $\mathbf{H}_{\mathbf{W}_{\text{srs}_s,\text{ps}_a}}$ strongly increases, conform (Robinson et al. 2016). These controlling measures together result in a quite steep increase of the connection weight reification state. However, after the learnt level of the weight has become high, the thresholds increase again, and the learning speed decreases again. This makes the excitability of srs_s and ps_a lower and stops the boosts on learning; this has a positive effect on stabilising the situation, in accordance with what, e.g., in Sjöström et al. (2008) is called 'The Plasticity Versus Stability Conundrum' (p. 773).

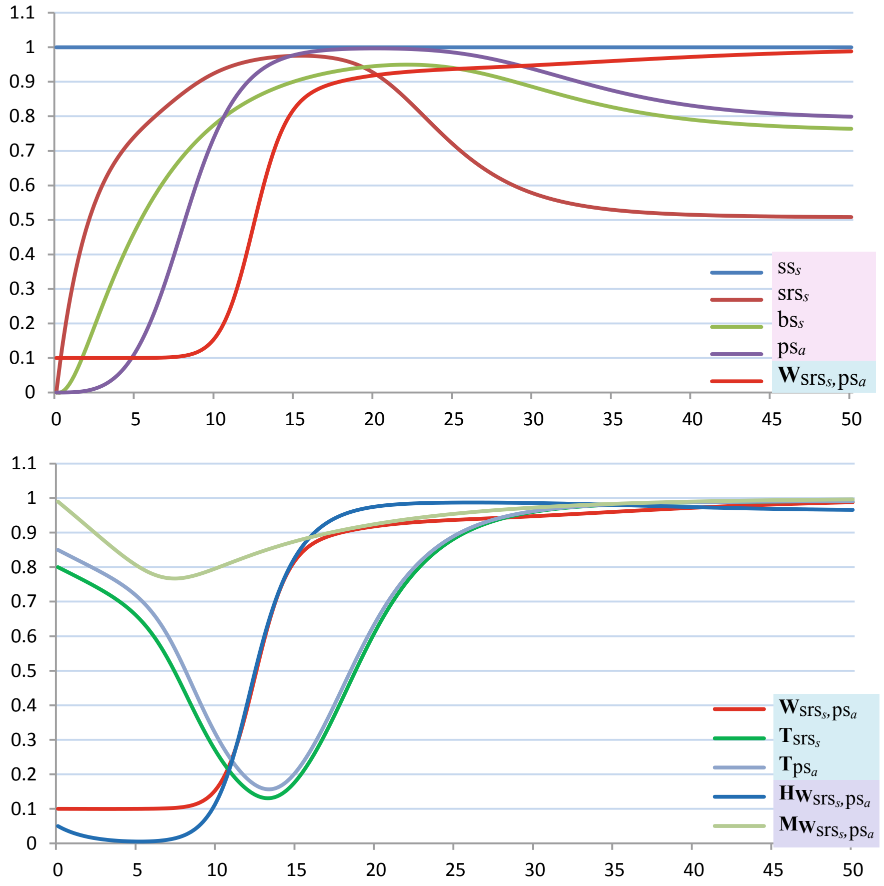


Fig. 4.4 Upper graph: dynamics of base states and the adaptive connection weight represented by $\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$. Lower graph: dynamics of the reification states including the first-order reification state $\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$ for the adaptive connection weight, and $\mathbf{T}_{\mathrm{srs}_s}$ and $\mathbf{T}_{\mathrm{ps}_a}$ for the activation threshold for the presynaptic and postsynaptic states srs$_s$ and ps$_a$, and the second-order reification states $\mathbf{H}_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}$ and $\mathbf{M}_{\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}}$ for the adaptation speed and persistence factor of the connection weight reification state $\mathbf{W}_{\mathrm{srs}_s,\mathrm{ps}_a}$

4.6 On the Added Complexity for Higher-Order Network Reification

Note that, as for any dynamical system, by adding adaptivity to a network always complexity is added. In this section, it is discussed how complexity of a network increases when reification is applied. The added complexity for first-order network reification was addressed in Chap. 3, Sect. 3.9. The outcome will be briefly summarized and next, the step to higher-order reification is made. To start with the outcome, network reification will increase complexity, but this will at most be quadratic in the number of nodes N and linear in the number of connections M of the original network. More specifically, if m the number of basic combination functions considered, then the number of nodes in the reified network is at most $(2 + m + N)N$. If not all connections are used but only a number M of them, the outcome is $(2 + m)N + M$. This is linear in the number of nodes and connections. The number of connections in the reified network is $(m + 1)N + 2M$. Again this is linear in the number of nodes and connections.

If this analysis is applied in an iterative manner for second-order network reification, then the increase in complexity is still polynomial: at most in the fourth power of the number of nodes:

$$\left(N^2\right)^2 = N^4 \tag{4.8}$$

Can this iteration still be continued further, thus obtaining nth-order reification for any n? Yes, theoretically there is no end in this. But also practically, for example, in the case used as illustration in the current chapter, the parameter $\mathbf{v}_{\mathrm{T}_{\Omega X_i,X_j}}$ for the norm of the average connection weight for the tipping point adaptation used as characteristic at the second reification level still could be made adaptive (e.g., related to how busy someone is) and reified at a third reification level. For third-order reification, the increase in complexity is still polynomial: at most in the order of

$$\left(\left(N^2\right)^2\right)^2 = N^8 \tag{4.9}$$

If n reification levels are added, then it is in the order of

$$N^{(2^n)} \tag{4.10}$$

which is still polynomial in N, but double exponential in n. The latter may suggest limiting the number of reification levels in practical applications to just a few, or, alternatively, in each reification step add only a few new reification states: for each step reification can be done in a partial manner as well. For example, if only speed factors are reified, the number of states will only increase in a linear way: one extra state for each existing state. Recall the double negative exponential pattern of hits in the order of

$$\mathrm{e}^{35.19\,\mathrm{e}^{-0.8684n}} \tag{4.11}$$

discussed in Chap. 1, Sect. 1.3. In the current literature, an adaptation of order higher than 2 is extremely rare and of order higher than 3 practically absent. This supports the idea that for now adaptation of order >3 is not considered interesting enough to be addressed. As shown above, for adaptation of order 3 the added complexity is in the order of an 8th degree polynomial in n, and for order 2 a 4th degree polynomial. In this context, note Chap. 10, Sect. 10.7 pointing out how efficient simulation of large-scale reified networks of thousands or even millions of states can be achieved by applying a form of compilation.

4.7 Discussion

The multilevel network reification architecture described here has advantages similar to those found for reification in modeling and programming languages in other areas of AI and Computer Science; e.g., Bowen and Kowalski (1982), Demers and Malenfant (1995), Galton (2006), Hofstadter (1979), Sterling and Shapiro (1996), Sterling and Beer (1989), Weyhrauch (1980). Some parts of this chapter were adopted from Treur (2018). A reified network enables to model dynamics of the original network by dynamics within the reified network, thus representing an adaptive network by a non-adaptive network. Network reification provides a unified manner of modelling adaptation principles, and allows comparison of such principles across different domains, as has been illustrated in Chap. 3. In the current chapter it was shown how a multilevel reified network architecture enables a structured and transparent manner to model network adaptation of any order, illustrated for second-order adaptive networks.

In this chapter, the introduced modeling environment for reified temporal-causal networks was applied to model a second-order adaptive Mental Network showing plasticity and metaplasticity as known from the empirical neuroscientific literature. Although some specific computational models for metaplasticity have been put forward with interesting perspectives for artificial neural networks, for example in Marcano-Cedeno et al. (2011), Andina et al. (2007, 2009), Fombellida et al. (2017), the modeling environment proposed here provides a more general architecture. Applications may extend well beyond the neuro-inspired area (as will be shown in Chap. 6 for a second-order adaptive Social Network).

The causal modeling area has a long history in AI; e.g., Kuipers and Kassirer (1983), Kuipers (1984). The current chapter can be considered a new branch in this causal modeling area. It adds dynamics to causal models, making them temporal, but the main contribution in the current chapter is that it adds a way to specify (multi-order) adaptivity in causal models, thereby conceptually using ideas on meta-level architectures that also have a long history in AI; e.g., Weyhrauch (1980), Bowen and Kowalski (1982), Sterling and Beer (1989). So the modeling approach connects two different areas with a long tradition in AI, thereby strongly extending the applicability of causal modeling to dynamic and adaptive notions such as plasticity and metaplasticity of any order, which otherwise are out of reach of causal modeling.

In the modeling approach, combination functions play a crucial role. They are declarative mathematical functions relating real numbers to real numbers. The functionality of an overall reified network is determined mainly by the choice of these functions and their use within a reified network architecture. In this sense, they are the powerful building blocks that enable to model in an easy manner dynamic processes which are adaptive of any order.

This construction can be continued to obtain a network architecture which is adaptive up to any order *n*. In Chap. 1, Sect. 1.3 it was discussed in how far adaptation principles of order 3 or higher are considered to be useful in the current literature, and a double negative exponential pattern was found for the number of hits in Google Scholar against the order of adaptation. However, in Chap. 7 an example network for evolutionary processes will be described of order higher than 2, and in Chap. 8 one or two inspired by ideas from Hofstadter (1979, 2007).

In an *n*th-order reified network there still will be network structures introduced in the last step from $n - 1$ to n that have no reification within the *n*th-order reified network. From a theoretical perspective, the construction can be repeated (countable) infinitely many times, for all natural numbers *n*; then $\boldsymbol{\omega}$-order reification is obtained, where $\boldsymbol{\omega}$ is the ordinal for the natural numbers. This is theoretically well-defined as a mathematical structure. All network structures in this $\boldsymbol{\omega}$-order reified network are reified within the network itself, so it is closed under reification. Whether or not such an $\boldsymbol{\omega}$-order construction has a useful application in practice, or can be used to explore theoretical research questions is still an open question, another subject for future research.

References

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

Andina, D., Jevtic, A., Marcano, A., Adame, J.M.B.: Error weighting in artificial neural networks learning interpreted as a metaplasticity model. In: Proceedings of IWINAC'07, Part I. Lecture Notes in Computer Springer Science, pp. 244–252 (2007)

Andina, D., Alvarez-Vellisco, A., Jevtic, A., Fombellida, J.: Artificial metaplasticity can improve artificial neural network learning. Intell. Autom. Soft Comput. **15**(4), 681–694 (2009)

Arnold, S., Suzuki, R., Arita, T.: Selection for representation in higher-order adaptation. Mind. Mach. **25**(1), 73–95 (2015)

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

Byrne, D.: The attraction hypothesis: do similar attitudes affect anything? J. Pers. Soc. Psychol. **51** (6), 1167–1170 (1986)

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

Daimon, K., Arnold, S., Suzuki, R., Arita, T.: The emergence of executive functions by the evolution of second-order learning. Artif. Life Rob. **22**, 483–489 (2017)

Demers, F.N., Malenfant, J.: Reflection in logic, functional and object-oriented programming: a short comparative study. In IJCAI'95 Workshop on Reflection and Meta-Level Architecture and their Application in AI, pp. 29–38 (1995)

Fombellida, J., Ropero-Pelaez, F.J., Andina, D.: Koniocortex-like network unsupervised learning surpasses supervised results on WBCD breast cancer database. In: Proceedings of IWINAC'17, Part II, LNCS, vol. 10338, pp. 32–41. Springer Publishers (2017)

Galton, A.: Operators vs. Arguments: The Ins and Outs of Reification. Synthese **150**, 415–441 (2006)

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

Helbing, D., Brockmann, D., Chadefaux, T., Donnay, K., Blanke, U., Woolley-Meza, O., Moussaid, M., Johansson, A., Krause, J., Schutte, S., Perc, M.: Saving human lives: what complexity science and information systems can contribute. J. Stat. Phys. **158**, 735–781 (2015)

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

Hofstadter, D.R.: I Am a Strange Loop. Basic Books, New York (2007)

Kuipers, B.J.: Commonsense reasoning about causality: deriving behavior from structure. Artif. Intell. **24**, 169–203 (1984)

Kuipers, B.J., Kassirer, J.P.: How to discover a knowledge representation for causal reasoning by studying an expert physician. In: Proceedings of the Eighth International Joint Conference on Artificial Intelligence, IJCAI'83. William Kaufman, Los Altos, CA (1983)

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)

Marcano-Cedeno, A., Marin-De-La-Barcena, A., Jimenez-Trillo, J., Pinuela, J.A., Andina, D.: Artificial metaplasticity neural network applied to credit scoring. Int. J. Neural Syst. **21**(4), 311–317 (2011)

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

Oh, M.M., Kuo, A.G., Wu, W.W., Sametsky, E.A., Disterhoft, J.F.: Watermaze learning enhances excitability of CA1 pyramidal neurons. J. Neurophysiol. **90**(4), 2171–2179 (2003)

Parsons, R.G.: Behavioral and neural mechanisms by which prior experience impacts subsequent learning. Neurobiol. Learn. Mem. **154**, 22–29 (2018)

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

Perc, M., Szolnoki, A.: Coevolutionary games—a mini review. BioSystems **99**, 109–125 (2010)

Robinson, B.L., Harper, N.S., McAlpine, D.: Meta-adaptation in the auditory midbrain under cortical influence. Nat. Commun. **7**, 13442 (2016)

Sehgal, M., Song, C., Ehlers, V.L., Moyer Jr., J.R.: Learning to learn—intrinsic plasticity as a metaplasticity mechanism for memory formation. Neurobiol. Learn. Mem. **105**, 186–199 (2013)

Schmidt, M.V., Abraham, W.C., Maroun, M., Stork, O., Richter-Levin, G.: Stress-induced metaplasticity: from synapses to behavior. Neuroscience **250**, 112–120 (2013)

Sharpanskykh, A., Treur, J.: Modelling and analysis of social contagion in dynamic networks. Neurocomputing **146**, 140–150 (2014)

Sjöström, P.J., Rancz, E.A., Roth, A., Hausser, M.: Dendritic excitability and synaptic plasticity. Physiol. Rev. **88**(769–840), 2008 (2008)

Sterling, L., Beer, R.: Metainterpreters for expert system construction. J. Logic Program. **6**, 163–178 (1989)

Sterling, L., Shapiro, E.: The Art of Prolog, Chap. 17, pp. 319–356. MIT Press (1996)

Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer Publishers (2016)

Treur, J.: Multilevel network reification: representing higher-order adaptivity in a network. In: Proceedings of the 7th International Conference on Complex Networks and their Applications, Complex Networks'18, vol. 1. Studies in Computational Intelligence, vol. 812, 635–651. Springer (2018)

Treur, J.: The ins and outs of network-oriented modeling: from biological networks and mental networks to social networks and beyond. In: Transactions on Computational Collective Intelligence. Contents of Keynote Lecture at ICCCI'18, vol. 32, pp. 120–139. Springer Publishers (2019a)

Treur, J.: Design of a software architecture for multilevel reified temporal-causal networks (2019b). https://doi.org/10.13140/rg.2.2.23492.07045. https://www.researchgate.net/publication/333662169

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

Zelcer, I., Cohen, H., Richter-Levin, G., Lebiosn, T., Grossberger, T., Barkai, E.: A cellular correlate of learning-induced metaplasticity in the hippocampus. Cereb. Cortex **16**, 460–468 (2006)