A unified approach to represent network adaptation principles by 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_3

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

document license
Article 25fa Dutch Copyright Act

Chapter 3
A Unified Approach to Represent Network Adaptation Principles by Network Reification

Abstract In this chapter, the notion of network reification is introduced: a construction by which a given (base) network is extended by adding explicit states representing the characteristics defining the base network's structure. This is explained for temporal-causal networks where connection weights, combination functions, and speed factors represent the characteristics for Connectivity, Aggregation, and Timing describing the network structure. Having the network structure represented in an explicit manner within the extended network enables to model the adaptation of the base network by dynamics within the reified network: an adaptive network is represented by a non-adaptive network. It is shown how the approach provides a unified modeling perspective on representing network adaptation principles across different domains. This is illustrated for a number of well-known network adaptation principles such as for Hebbian learning in Mental Networks and for network evolution based on homophily in Social Networks.

3.1 Introduction

Reification is a notion that is known from different scientific areas. Literally, it means representing something abstract as a material or concrete thing (Merriam-Webster dictionary), or making something abstract more concrete or real (Oxford dictionaries). Well known examples in linguistics, logic and knowledge representation domains are representing relations between objects as objects themselves (reified relations); this enables to introduce variables and relations over these reified relations. In this way, the expressivity of a language can be extended substantially. In such a way in logic, statements can be represented by term expressions over which predicates can be defined. This idea of reification has been applied in particular to many modeling and programming languages, for example, logical, functional, and object-oriented languages (e.g., Weyhrauch 1980; Bowen and Kowalski 1982; Bowen 1985; Sterling and Shapiro 1986; Sterling and Beer 1989; Demers and Malenfant 1995; Galton 2006). Also in fundamental research, the notion of reification plays an important role. For example, Gödel's

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_3

famous incompleteness theorems in Mathematical Logic depend on reification of logical statements by representing them by natural numbers over which predicates are used to express, for example, (non)provability of such statements (e.g., Smorynski 1977; Hofstadter 1979).

In this chapter, the general notion of reification is applied to networks in particular, and illustrated for a Network-Oriented Modeling approach based on temporal-causal networks (Treur 2016, 2019). A network (the base network) is extended by adding explicit network states representing characteristics of the network structure. In a temporal-causal network, the network structure is defined by three types of characteristics: connection weights (for Connectivity), combination functions (for Aggregation), and speed factors (for Timing). By reifying these characteristics of the base network as states in the extended network, and defining proper causal relations for them and with the other states, an extended, reified network is obtained which explicitly represents the structure of the base network, and how this network structure evolves over time. This enables to model dynamics *of* the base network by dynamics *within* the reified network: thus an adaptive network is represented as a non-adaptive network.

By the introduced concept of network reification it becomes possible to analyse network adaptation principles from an inherent network modeling perspective. Applying this, a unified framework is obtained to represent and compare network adaptation principles across different domains. To illustrate this, a number of well-known network adaptation principles are analysed and compared, including, for example, adaptation principles for Hebbian learning for Mental Networks, and for bonding based on homophily for Social Networks.

In Sect. 3.2 the Network-Oriented Modeling approach based on temporal-causal networks is briefly summarized. Next, in Sect. 3.3 the idea of reifying the network structure characteristics by additional reification states representing them is introduced. In Sect. 3.4 it is discussed how causal relations for these reified states can be defined by which they contribute to an aggregated causal effect on the states in the base network. In Sect. 3.5 the universal combination function and difference equation for the base states' dynamics is briefly presented, which generalises to reified networks what in Chap. 2 are called basic difference or differential equations. Section 3.6 shows how the obtained reification approach can be applied to analyse and unify many well-known network adaptation principles from a Network-Oriented Modeling perspective. In Sect. 3.7, as an illustration an example simulation within a developed software environment for network reification shows how an adaptive speed factor and an adaptive combination function can be used to model a scenario of a manager who adapts to an organisation. This example illustrates how the role matrices format to specify a non-reified network's structure as introduced in Chap. 2, can be generalised relatively easily to obtain a useful means to specify a reified network's structure. In Sect. 3.8 the (im)possibility of joint reification states for multiple base states or roles is briefly discussed. Section 3.9 presents an analysis of the added complexity of the reification construction, and Sect. 3.10 is a final discussion.

3.2 Temporal-Causal Networks: Structure and Dynamics

In general, a network structure is considered to be defined by nodes (or states) and connections between them. However, this only covers very general aspects of a network structure in which no distinctions can be made, for example, between different strengths of connections, and different ways in which multiple connections to the same node interact and work together. In this sense, in many cases a plain graph structure provides underspecification of a network structure. Also, Pearl (2000) points out this problem of underspecification in the context of causal networks from the (deterministic) Structural Causal Model perspective. In that context functions f_i for nodes V_i are used to specify how multiple impacts on the same node V_i should be combined, but this concept is lacking in a plain graph representation:

> Every causal model *M* can be associated with a directed graph, *G(M)* (…) 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

3.2.1 Conceptual Representation of a Temporal-Causal Network Model

A conceptual representation of the network structure of a temporal-causal network model does involve representing in a declarative manner states and connections between them that represent (causal) impacts of states on each other. This part of the conceptual representation is often depicted in a conceptual picture by a graph with nodes and directed connections. However, a *full conceptual representation* of a temporal-causal network structure also includes a number of labels for such a graph. First, in reality, not all connections are equally strong, so some notion of *strength of a connection* $\boldsymbol{\omega}_{X,Y}$ is used as a label for connections (Connectivity). Second, a combination function $\mathbf{c}_Y(..)$ to *aggregate multiple impacts* on a state is used as a label for states (Aggregation). Third, for each state a notion of *speed factor* $\boldsymbol{\eta}_Y$ of a state is used as a label for timing of the state's processes (Timing). These three notions, called connection weight, combination function, and speed factor, make the graph of states and connections a labeled graph. This labeled graph forms the *defining network structure* of a temporal-causal network model in the form of a conceptual representation; see Table 3.1, adopted from (Treur 2019), and see Fig. 3.1 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, $\mathbf{c}_Y(..)$ for combination function, and $\boldsymbol{\eta}_Y$ for speed factor.

Combination functions can have different forms, as there are many different approaches possible to address the issue of combining multiple impacts. Combination functions provide a way to specify how multiple causal impacts on this state are aggregated. For this aggregation, pre-defined combination functions from a library can be used, or modified according to a pre-designed template.

Table 3.1 Conceptual representation of a temporal-causal network model: the 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 through connection $X \rightarrow Y$
Aggregating multiple impacts on a state	$\mathbf{c}_Y(..)$	For each state Y (a reference to) a *combination function* $\mathbf{c}_Y(..)$ is chosen to combine the causal impacts of other states on state Y
Timing of the effect of impact	$\boldsymbol{\eta}_Y$	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

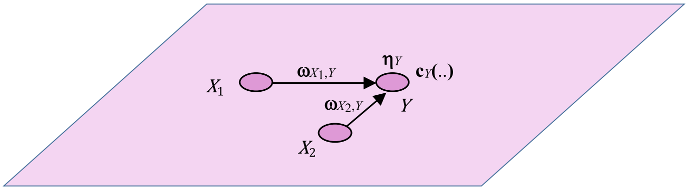

Fig. 3.1 A fragment of a temporal-causal network structure in a conceptual labeled graph representation

3.2.2 *Numerical Representation of a Temporal-Causal Network Model*

Next it is shown how a conceptual representation (based on states and connections enriched with labels for connection weights, combination functions, and speed factors), determines a numerical representation defining the network's intended dynamic semantics (Treur 2016), Chap. 2; see Table 3.2, adopted from (Treur 2019). Note that here X_1, …, X_k are the states from which state Y gets its incoming connections.

The difference equations in the last row in Table 3.2 form the numerical representation of the dynamics of a temporal-causal network model. They can be used for simulation and mathematical analysis, and also be written in differential equation format:

$$\begin{aligned} Y(t + \Delta t) &= Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]\Delta t \\ \mathbf{d}Y(t)/\mathbf{d}t &= \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)] \end{aligned} \tag{3.1}$$

where the X_i are all states from which state Y gets its incoming connections.

Table 3.2 Numerical representation of a temporal-causal network model: the network dynamics

Concept	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, usually in the [0, 1] interval
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 connection weight $\boldsymbol{\omega}_{X,Y}$
Aggregating multiple causal 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 causal 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 $\boldsymbol{\eta}_Y$; here the X_i are all states from which state Y gets its incoming connections

3.2.3 Basic Combination Functions, Their Parameters and Combining Them

Often used examples of combination functions are shown in Table 3.3. As shown in Table 3.2 these functions are used by applying them on the single causal impacts for V_1, …, V_k for the states X_1, …, X_k from which state Y gets its incoming connections. They are the *identity* **id(.)** for states with impact from only one other state, the *scaled sum* combination function $\mathbf{ssum}_{\boldsymbol{\lambda}}$**(..)** with scaling factor $\boldsymbol{\lambda}$, and the *simple logistic sum* combination function $\mathbf{slogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}$**(..)** and *advanced logistic sum* combination function $\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}$**(..)**, both with steepness $\boldsymbol{\sigma}$ and threshold $\boldsymbol{\tau}$; see also (Treur 2016), Chap. 2, Table 2.10.

Other options for combination functions are the *scaled minimum* combination function $\mathbf{smin}_{\boldsymbol{\lambda}}$**(..)**, *scaled maximum* combination function $\mathbf{smax}_{\boldsymbol{\lambda}}$**(..)**, the *Euclidean* combination function of ***n***th-order with scaling factor $\boldsymbol{\lambda}$ (with n any number > 0,

Table 3.3 Often used combination functions

Name	Formula
Identity	$\mathbf{id}(V) = V$
Scaled sum	$\mathbf{ssum}_\lambda(V_1, \ldots, V_k) = \frac{V_1 + \ldots + V_k}{\lambda}$
Simple logistic	$\mathbf{slogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = \frac{1}{1+e^{-\sigma(V_1 + \ldots + V_k - \tau)}}$
Advanced logistic	$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = \left[\frac{1}{1+e^{-\sigma(V_1 + \ldots + V_k - \tau)}} - \frac{1}{1+e^{\sigma\tau})}\right](1+e^{-\sigma\tau})$
Scaled minimum	$\mathbf{smin}_\lambda(V_1, \ldots, V_k) = \frac{\min(V_1, \ldots, V_k)}{\lambda}$
Scaled maximum	$\mathbf{smax}_\lambda(V_1, \ldots, V_k) = \frac{\max(V_1, \ldots, V_k)}{\lambda}$
Euclidean function	$\mathbf{eucl}_{n,\lambda}(V_1, \ldots, V_k) = \sqrt[n]{\frac{V_1^n + \ldots + V_k^n}{\lambda}}$
Scaled geometric mean	$\mathbf{sgeomean}_\lambda(V_1, \ldots, V_k) = \sqrt[k]{\frac{V_1 * \ldots * V_k^n}{\lambda}}$

generalising the scaled sum $\mathbf{ssum}_\lambda(..)$ for $n = 1$), and the *scaled geometric mean* combination function $\mathbf{sgeomean}_\lambda(..)$.

The above examples of combination functions are called *basic combination functions* and in a general format indicated by $\text{bcf}_i(..)$. As also discussed in Chap. 2, Sect. 2.3.2 they can be combined to form more complex combination functions by forming weighted averages of them with *combination function weight* factors $\gamma_1, \ldots, \gamma_m$ as follows

$$\mathbf{c}_Y(V_1, \ldots, V_k) = \frac{\gamma_{1,Y}\,\text{bcf}_1(V_1, \ldots, V_k) + \ldots + \gamma_{m,Y}\,\text{bcf}_m(V_1, \ldots, V_k)}{\gamma_{1,Y} + \ldots + \gamma_{m,Y}} \quad (3.2)$$

This type of representation (with the $\gamma_{j,Y}$ depending on time) for combination functions will also be used for combination function reification in Sect. 3.5. Usually, combination functions have parameters, for example, a scaling factor λ, or steepness σ and threshold τ for logistic functions. These *combination function parameters* can also be used as arguments in the notation $\text{bcf}_i(..)$, and denoted by $\pi_{i,j}$, so that it becomes $\text{bcf}_i(\pi_{1,1}, \pi_{1,2}, V_1, \ldots, V_k)$ and

$$\mathbf{c}_Y(\pi_{1,1}, \pi_{1,2}, \ldots, \pi_{1,m}, \pi_{1,m}, V_1, \ldots, V_k) = \frac{\gamma_{1,Y}\text{bcf}_1(\pi_{1,1,Y}, \pi_{2,1,Y}, V_1, \ldots, V_k) + \ldots + \gamma_{m,Y}\text{bcf}_m(\pi_{1,m,Y}, \pi_{2,m,Y}, V_1, \ldots, V_k)}{\gamma_{1,Y} + \ldots + \gamma_{m,Y}} \quad (3.3)$$

These characteristics γ will also be used in an adaptive manner for combination function reification in the example reified network models described in Sects. 3.6.7 and 3.7.

3.2.4 Normalisation, Stationary Points and Equilibria for Temporal-Causal Network Models

Often a combination function is assumed to be normalised by setting a proper scaling factor value. If the scaling factor is too low, an undesirable artificial upward bias may occur, and when the scaling factor is too high an artificial downward bias. Therefore, normalization of the combination functions is important to get a realistic simulation. The notion of normalisation is defined as follows.

Definition 1 (**normalised**) A network is *normalised* if for each state Y it holds $c_Y(\boldsymbol{\omega}_{X_1,Y}, \ldots, \boldsymbol{\omega}_{X_k,Y}) = 1$, where X_1, …, X_k are the states from which Y gets its incoming connections.

As an example, for a Euclidean combination function of $\boldsymbol{n}$th-order the scaling factor value choice

$$\boldsymbol{\lambda}_Y = \boldsymbol{\omega}^{\boldsymbol{n}}_{X_1,Y} + \ldots + \boldsymbol{\omega}^{\boldsymbol{n}}_{X_k,Y}$$

will provide a normalised network. This can be done in general as follows:

Normalising a combination function
If any combination function $\mathbf{c}_Y(..)$ is replaced by $\mathbf{c}'_Y(..)$ defined as

$$\mathbf{c}'_Y(V_1, \ldots, V_k) = \mathbf{c}_Y(V_1, \ldots, V_k)/\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}, .., \boldsymbol{\omega}_{X_k,Y}) \tag{3.4}$$

where X_1, …, X_k are the states with outgoing connections to Y and assuming $\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}, \ldots, \boldsymbol{\omega}_{X_k,Y}) > 0$ for $\boldsymbol{\omega}_{X_i,Y} > 0$, then the network becomes normalised.

For different example functions, following the normalisation step above, their normalised variants are given by Table 3.4.

Next, this section focuses on some tools that allow to analyse emerging behaviour and how it relates to the structure properties. The basic definition is as follows.

Definition 2 (**stationary point and equilibrium**) A state Y has a *stationary point* at t if $\mathbf{d}Y(t)/\mathbf{d}t = 0$. The network is in *equilibrium* at t if every state Y of the model has a stationary point at t.

Applying this definition to the specific differential equation format for a temporal-causal network model, the very simple criterion expressed in Lemma 1 can be formulated in terms of the temporal-causal network structure characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$:

Lemma 1 (**Criterion for a stationary point in a temporal-causal network**) Let Y be a state and X_1,…, X_k the states with outgoing connections to state Y.

Table 3.4 Normalisation of the different examples of combination functions

Combination function	Notation	Normalising scaling factor λ	Normalised combination function
Identity function	$\mathbf{id}(.)$	$\boldsymbol{\omega}_{X,Y}$	$V/\boldsymbol{\omega}_{X,Y}$
Scaled sum	$\mathbf{ssum}_\lambda(V_1,\ldots,V_k)$	$\boldsymbol{\omega}_{X_1,Y}+..+\boldsymbol{\omega}_{X_k,Y}$	$(V_1+\ldots+V_k)/(\boldsymbol{\omega}_{X_1,Y}+..+\boldsymbol{\omega}_{X_k,Y})$
Simple logistic	$\mathbf{slogistic}_{\sigma,\tau}(V_1,\ldots,V_k)$	$\mathbf{slogistic}_{\sigma,\tau}(\boldsymbol{\omega}_{X_1,Y},..,\boldsymbol{\omega}_{X_k,Y})$	$\frac{1+e^{-\sigma(\omega_{X_1,Y}+\ldots+\omega_{X_k,Y}-\tau)}}{1+e^{-\sigma(V_1+\ldots+V_k-\tau)}}$
Advanced logistic	$\mathbf{alogistic}_{\sigma,\tau}(V_1,\ldots,V_k)$	$\mathbf{alogistic}_{\sigma,\tau}(\boldsymbol{\omega}_{X_1,Y},..,\boldsymbol{\omega}_{X_k,Y})$	$\frac{\frac{1}{1+e^{-\sigma(V_1+\ldots+V_k-\tau)}}-\frac{1}{1+e^{\sigma\tau}}}{\frac{1}{1+e^{-\sigma(\omega_{X_1,Y}+\ldots+\omega_{X_k,Y}-\tau)}}-\frac{1}{1+e^{\sigma\tau}}}$
Scaled maximum	$\mathbf{smax}_\lambda(V_1,\ldots,V_k)$	$\max(\boldsymbol{\omega}_{X_1,Y},..,\boldsymbol{\omega}_{X_k,Y})$	$\max(V_1,\ldots,V_k)/\max(\boldsymbol{\omega}_{X_1,Y},..,\boldsymbol{\omega}_{X_k,Y})$
Scaled minimum	$\mathbf{smin}_\lambda(V_1,\ldots,V_k)$	$\min(\boldsymbol{\omega}_{X_1,Y},..,\boldsymbol{\omega}_{X_k,Y})$	$\min(V_1,\ldots,V_k)/\min(\boldsymbol{\omega}_{X_1,Y},..,\boldsymbol{\omega}_{X_k,Y})$
Euclidean	$\mathbf{eucl}_{n,\lambda}(V_1,\ldots,V_k)$	$\boldsymbol{\omega}^n_{X_1,Y}+\ldots+\boldsymbol{\omega}^n_{X_k,Y}$	$\sqrt[n]{\frac{V_1^n+\ldots+V_k^n}{\boldsymbol{\omega}^n_{X_1,Y}+\ldots+\boldsymbol{\omega}^n_{X_k,Y}}}$
Scaled geometric mean	$\mathbf{sgeomean}_\lambda(V_1,\ldots,V_k)$	$\boldsymbol{\omega}_{X_1,Y}*..*\boldsymbol{\omega}_{X_k,Y}$	$\sqrt[k]{\frac{V_1*\ldots*V_k}{\boldsymbol{\omega}_{X_1,Y}*\ldots*\boldsymbol{\omega}_{X_k,Y}}}$

Then Y has a stationary point at t if and only if

$$\boldsymbol{\eta}_Y = 0 \quad \text{or} \quad \mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}X_k(t)) = Y(t) \tag{3.5}$$

The latter equation is called a stationary point or equilibrium equation. This criterion will be used in Sects. 3.6 and 3.7 to determine in a straightforward manner the equilibrium equations for states for the different adaptation principles addressed.

3.3 Modeling Adaptive Networks by Network Reification

In general, the structure of a network is described by certain characteristics, such as connection weights. Usually, these network characteristics are considered static: they are assumed not to change over a period of time. This stands in the way of addressing network evolution, where the network structure does change.

Network evolution is studied usually in a hybrid manner by considering a separate dynamic model for additional variables representing network structure characteristics. Such a dynamic model is, for example, specified by a numerical mathematical form of difference or differential equations and a procedural description to simulate these equations. Such a description is different from and outside the context of the Network-Oriented Modeling perspective on dynamics used within the base network itself. In specific applications, still, this extra-network dynamical model has to interact intensively with the internal network dynamics of the base network. For example, see Chap. 1, Sect. 1.4, and in particular Fig. 1.2.

Network reification provides a way to address this in a more unified manner, staying more genuinely within the Network-Oriented Modeling perspective. Using network reification, the base network is extended by extra network states that represent the characteristics of the base network structure (Connectivity, Aggregation, and Timing). In this way, the whole model is specified by one network, a network extension of the base network. Thus the modeling stays within the network context. The new additional states representing the values for the network structure characteristics are called *reification states* for these characteristics. The network characteristics are *reified* by these states. The reification states are depicted in the upper plane in Fig. 3.2, together with the dashed lines indicating the representation relations with the network characteristics of the base network in the lower plane. What can be reified in temporal-causal networks, in particular, are the following characteristics of the network structure: the connection weights, combination functions, combination function parameters, and speed factors. For connection weights $\boldsymbol{\omega}_{X_i,Y}$ and speed factors $\boldsymbol{\eta}_Y$ their added reification states $\mathbf{W}_{X_i,Y}$ and $\mathbf{H}_Y$ represent the value of them.

For combination functions $\mathbf{c}_Y(..)$ the general idea is that from a theoretical perspective a coding is needed for all options for such functions by numbers; for example, assuming there is a countable number, the set of all of them is numbered by natural numbers n = 1, 2, …, and the reified state $\mathbf{C}_Y$ representing them actually represents that number. This is the general idea for addressing reification of

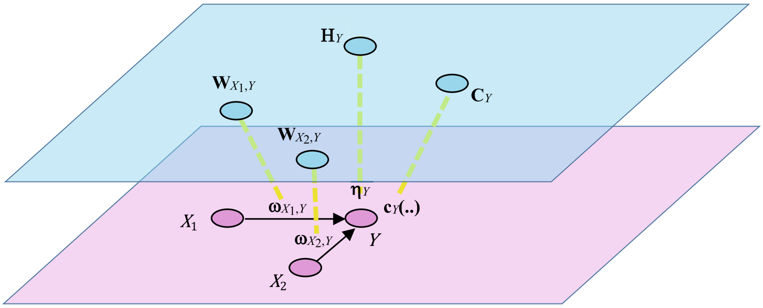

Fig. 3.2 Representation relations (the dashed lines) for connection weight reification states $\mathbf{W}_{X,Y}$, combination function reification states $\mathbf{C}_Y$ and speed factor reification states $\mathbf{H}_Y$: network state $\mathbf{W}_{X,Y}$ represents network characteristic $\boldsymbol{\omega}_{X,Y}$, network state $\mathbf{H}_Y$ represents network characteristic $\boldsymbol{\eta}_Y$, and network state $\mathbf{C}_Y$ represents network characteristic $\mathbf{c}_Y(..)$

combination functions; however, below a more refined approach is shown that is easier to use in practice.

By adding proper causal connections to the reification states (incoming arrows) within the extended network, these states are affected and therefore become adaptive. For many examples of this, see Sects. 3.6 and 3.7. Outward causal connections from reification states (outgoing downward arrows to the related base network states) make their intended special effect happen. This will be addressed in Sect. 3.4; see the pink downward arrows in Fig. 3.3.

3.4 Incorporating the Impact of Downward Causal Connections for Reification States

The added reification states need connections to obtain a well-connected overall network. As always, connections of a state are of two types: (1) outgoing connections, and (2) incoming connections. In the first place outward connections (1) from the reification states to the states in the base network are needed, in order to model how they have their special effect on the adaptive dynamics of the base network. More specifically, it has to be defined how the reification states contribute causally to an aggregated impact on the base network state. In addition to a downward connection, also the combination function has to be (re)defined for the aggregated impact on that base state. Both these downward causal relations and the combination functions are defined in a generic manner, related to the role of a specific network characteristic in the overall dynamics of a state in a temporal-causal network. That will be discussed in the current section and in Sect. 3.5.

In addition, incoming connections (2) of the reification states are added in order to model specific network adaptation principles. These may concern upward connections from the states of the base network to the reification states, or horizontal

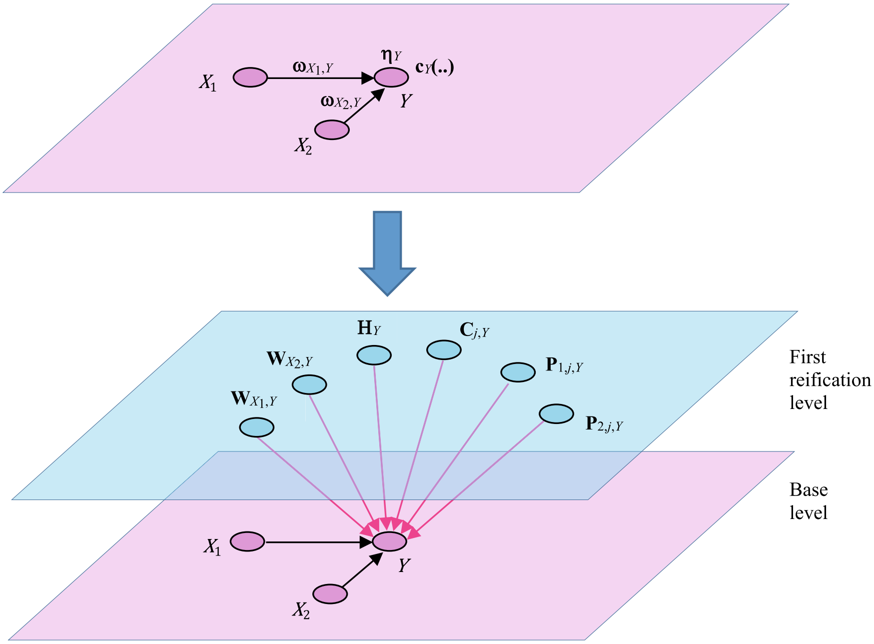

Fig. 3.3 Network reification for temporal-causal networks: downward connections from reification states to base network states

mutual connections between states within the upper plain, or both, depending on the specific network adaptation principles addressed. These connections are not generic but depend on the specific adaptation principle addressed; they will be discussed and illustrated for many cases in Sects. 3.6 and 3.7.

For the downward connections (1), the general pattern is that each of the reification states $\mathbf{W}_{X_i,Y}$, $\mathbf{H}_Y$ and $\mathbf{C}_Y$ for the reified network characteristics (connection weights, speed factors, and combination functions), has a specific causal connection to state Y in the base network according to its own special role. These are the (pink) downward arrows from the reification plane to the base plane in Fig. 3.3. Actually, $\mathbf{C}_Y$ is a vector of states ($\mathbf{C}_{1,Y}$, $\mathbf{C}_{2,Y}$, ...) with a (small) number of different components $\mathbf{C}_{1,Y}$, $\mathbf{C}_{2,Y}$, ... for different basic combination functions that will be explained below. Note that combination functions may contain some parameters, for example, for the scaled sum combination function the scaling factor $\boldsymbol{\lambda}$, and for the advanced logistic function the steepness $\boldsymbol{\sigma}$ and the threshold $\boldsymbol{\tau}$. For these parameters also reification states $\mathbf{P}_{i,j,Y}$ can be added, with the possibility to make them adaptive as well. More specifically, for each basic combination function represented by $\mathbf{C}_{j,Y}$ there are two parameters $\boldsymbol{\pi}_{1,i}$ and $\boldsymbol{\pi}_{2,i}$ that are reified by parameter reification states $\mathbf{P}_{1,j,Y}$ and $\mathbf{P}_{2,j,Y}$.

Note that the 3D layout of these figures and the depicted planes are just for understanding; in a mathematical or computational sense, they are not part of the network specification. However, for each of the reification states, it is crucial to know what it is that they are reifying and for what base state. Therefore the names of the reification states are chosen in such a way that this information is visible. For example, in the name $\mathbf{H}_Y$ the **H** indicates that it concerns speed factor (indicated by $\boldsymbol{\eta}$) reification and the subscript Y that it is for base state Y. So, in general, the bold capital letter **R** in $\mathbf{R}_{subscript}$ indicates the type of reification and the subscript the concerning base state Y, or (for **W**) the pair of states X, Y. This **R** indicates the *role* that is played by this reification state. This role corresponds one to one to the characteristics of the base network structure that is reified: connection weight $\boldsymbol{\omega}$ for Connectivity, speed factor $\boldsymbol{\eta}$ for Timing, basic combination function **c(..)**, and parameter $\boldsymbol{\pi}$ for Aggregation. The role defines which special effect the reification state has on base state Y. By the specific role matrix in which a downward connection is indicated, the downward connection and the special effect for that role is completely determined. The reification state with this connection cannot occur in any other role matrix then. For example, if such a downward link is indicated in role matrix **ms** for the speed factors, the value of the reification state can only be used for the speed factor, not for something else. In this way, there are four roles for reification states:

- the role of connection weight reification **W** reifying connection weights $\boldsymbol{\omega}$
- the role of speed factor reification **H** reifying speed factors $\boldsymbol{\eta}$
- the role of combination function reification $\mathbf{C}_j$ reifying combination functions **c(..)**
- the role of parameter reification $\mathbf{P}_{i,j}$ reifying combination function parameters $\boldsymbol{\pi}_{i,j}$

In accordance with this indicated role information, each reification state has exactly one downward causal connection, which goes to the specified base state Y, and in the reified network this downward connection has its special effect according to its role **R** in the aggregation of the causal impacts on Y by a new, dedicated combination function. Note that to keep a transparent one-to-one relation between a reification state representing one of the base network characteristics, and the actual value used for that characteristic in the dynamics of state Y, the (pink) downward links get automatically standard weight value 1; this cannot be changed.

The general picture is that the base states have more incoming connections now, some of which have specific roles, with special effects according to their role. Therefore, in the reified network new combination functions for the base states are needed to aggregate these special effects. These new combination functions can be expressed in a universal manner based on the original combination functions, and the different reification states, but to define them some work is needed. That will be done in Sect. 3.5, but also in a more extensive manner in Chap. 10.

3.5 The Universal Combination Function and Difference Equation for Reified Networks

In this section, the universal combination function and universal difference (or differential) equation for base states Y within a reified network are introduced. The universal difference (or differential) equation generalises what in Chap. 2, Sects. 2.3.1 and 2.4.2 are called the basic difference (or differential) equations. Recall from Sect. 3.2, that based on the basic combination functions $\mathrm{bcf}_j(..)$ from the library, the general combination function format is expressed in terms of the network structure characteristics and the single impacts from the base states indicated by V_1, …, V_k as follows:

$$\mathbf{c}_Y(\boldsymbol{\pi}_{1,1}, \boldsymbol{\pi}_{1,2}, \ldots, \boldsymbol{\pi}_{1,m}, \boldsymbol{\pi}_{1,m}, V_1, \ldots, V_k) = \frac{\boldsymbol{\gamma}_{1,Y}\,\mathrm{bcf}_1\left(\boldsymbol{\pi}_{1,1,Y}, \boldsymbol{\pi}_{2,1,Y}, V_1, \ldots, V_k\right) + \ldots + \boldsymbol{\gamma}_{m,Y}\mathrm{bcf}_m\left(\boldsymbol{\pi}_{1,m,Y}, \boldsymbol{\pi}_{2,m,Y}, V_1, \ldots, V_k\right)}{\boldsymbol{\gamma}_{1,Y} + \ldots + \boldsymbol{\gamma}_{m,Y}} \quad (3.6)$$

To enable reification, the idea is that all network structure characteristics can become dynamic. This is the main step made here, compared to Chap. 2. In particular, within the combination function this holds for the $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$ characteristics, so that these characteristics can get an argument t for time:

$$\mathbf{c}_Y(t, \boldsymbol{\pi}_{1,1}(t), \boldsymbol{\pi}_{1,2}(t), \ldots, \boldsymbol{\pi}_{1,m}(t), \boldsymbol{\pi}_{1,m}(t), V_1, \ldots, V_k) = \frac{\boldsymbol{\gamma}_{1,Y}(t)\mathrm{bcf}_1\left(\boldsymbol{\pi}_{1,1,Y}(t), \boldsymbol{\pi}_{2,1,Y}(t), V_1, \ldots, V_k\right) + \ldots + \boldsymbol{\gamma}_{m,Y}(t)\mathrm{bcf}_m\left(\boldsymbol{\pi}_{1,m,Y}(t)\boldsymbol{\pi}_{2,m,Y}(t), V_1, \ldots, V_k\right)}{\boldsymbol{\gamma}_{1,Y}(t) + \ldots + \boldsymbol{\gamma}_{m,Y}(t)} \quad (3.7)$$

These combination functions become adaptive if for these dynamic characteristics $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$, reification states **C** and **P** are introduced for their role, as shown in Fig. 3.3. Within the difference equation also the speed factor $\boldsymbol{\eta}_Y$ and the connection weights $\boldsymbol{\omega}_{X_i,Y}$ occur. Also, these network structure characteristics can be made dynamic by adding the argument t, and reification states **H** and **W** can be added for their role (see Fig. 3.3).

$$Y(t + \Delta t) = Y(t) + \boldsymbol{\eta}_Y(t)[\mathbf{c}_Y(t, \boldsymbol{\omega}_{X_1,Y}(t)X_1(t), \ldots, \boldsymbol{\omega}_{X_k,Y}(t)X_k(t)) - Y(t)]\Delta t \quad (3.8)$$

Using the above expressions (3.7) and (3.8) the difference and differential equation for state Y can be found based on the appropriate choice of the *universal combination function* $\mathbf{c}^*_Y(..)$ for Y in the reified network. This combination function in the reified network needs arguments for the reification states for all network structure characteristics as they are dynamic now and have an impact on state Y. So, in addition to the impacts within the original base network the new function $\mathbf{c}^*_Y(..)$ needs additional arguments indicated by variables H, C_1, …., C_m, $P_{1,1}$, $P_{2,1}$, …, $P_{1,m}$, $P_{2,m}$, W_1, …, W_k for the special effects of the different types of reification states on state Y, where:

- H is used for the speed factor reification $\mathbf{H}_Y(t)$ representing $\boldsymbol{\eta}_Y(t)$
- C_j for the combination function weight reification $\mathbf{C}_{j,Y}(t)$ representing $\boldsymbol{\gamma}_{j,Y}(t)$
- $P_{i,j}$ for the combination function parameter reification $\mathbf{P}_{i,j,Y}(t)$ representing $\boldsymbol{\pi}_{i,j,Y}(t)$
- W_i for the connection weight reification $\mathbf{W}_{X_i,Y}(t)$ representing $\boldsymbol{\omega}_{X_i,Y}(t)$.

It has been found out (for more details, see also Chap. 10) that the function $\mathbf{c}^*_Y(H, C_1, \ldots, C_m, P_{1,1}, P_{2,1}, \ldots, P_{1,m}, P_{2,m}, W_1, \ldots, W_k, V_1, \ldots, V_k, V)$ that is needed here can be defined as follows:

$$\mathbf{c}^*_Y(H, C_1, \ldots, C_m, P_{1,1}, P_{2,1,\ldots,} P_{1,m}, P_{2,m}, W_1, \ldots, W_k, V_1, \ldots, V_k, V)$$
$$= H\frac{C_1\mathrm{bcf}_1\left(P_{1,1}, P_{2,1}, W_1V_1, .., W_kV_k\right) + \ldots + C_m\,\mathrm{bcf}_m\left(P_{1,m},\, P_{2,m},\, W_1V_1, .., W_kV_k\right)}{C_1 + \ldots + C_m} + (1-H)V \tag{3.9}$$

where

- V_i for the state values $X_i(t)$ of base states X_i, which are the base states from which Y gets its incoming connections
- V for the state value $Y(t)$ of base state Y

This combination function shows the way in which the special impacts of the downward causal connections from the reification states (their special effects) according to their role are aggregated together with the other impacts within the base level. Then based on this combination function (and using speed factor with default value $\boldsymbol{\eta}^*_Y = 1$ and weights 1 for the incoming connections), within the reified network the *universal difference equation* for base state Y is

$$\begin{aligned} Y(t+\Delta t) = {} & Y(t) \\ & + [\mathbf{c}^*_Y(\mathbf{H}_Y(t), \mathbf{C}_{1,Y}(t), \ldots, \mathbf{C}_{m,Y}(t), \mathbf{P}_{1,1,Y}(t), \mathbf{P}_{2,1,Y}(t), \ldots, \mathbf{P}_{1,m,Y}(t), \\ & \mathbf{P}_{2,m,Y}(t), \mathbf{W}_{X_1,Y}(t), \ldots, \mathbf{W}_{X_k,Y}(t), X_1(t), \ldots, X_k(t), Y(t)) - Y(t)]\Delta t \end{aligned} \tag{3.10}$$

In case of full reification this difference equation does not have any parameter for the network characteristics, it only has variables; therefore it has a universal form for every base state. So this is the way in which the special impact of the downward causal connections from the reification states is incorporated within the temporal-causal network format.

It can be verified using (3.9) by rewriting that this universal difference equation in (3.10) indeed is equivalent to the above difference equation in (3.8). For more explanation and background on this, see Chap. 10. The *universal differential equation* variant is as follows (leaving out the reference to t):

$$\mathbf{d}Y/\mathbf{d}t = \mathbf{H}_Y\left[\frac{\mathbf{C}_{1,Y}\mathrm{bcf}_1\left(\mathbf{P}_{1,1,Y},\mathbf{P}_{2,1,Y},\mathbf{W}_{X_1,Y}X_1,..,\mathbf{W}_{X_k,Y}X_k\right)+\ldots+\mathbf{C}_{m,Y}\mathrm{bcf}_m\left(\mathbf{P}_{1,m,Y},\mathbf{P}_{2,m,Y},\mathbf{W}_{X_1,Y}X_1,..,\mathbf{W}_{X_k,Y}X_k\right)}{\mathbf{C}_{1,Y}+\ldots+\mathbf{C}_{m,Y}}-Y\right] \tag{3.11}$$

The universal combination function was introduced above in (3.9) out of the blue, it may seem. But at least now it was shown that it fulfills what is required. In Chap. 10, it is shown in more detail how this universal combination function can be derived and it is illustrated for some cases. From an abstract point of view now it has been found that the class of temporal-causal network models is closed under the operation of reification.

Note that there are two important advantages in keeping the reified network in the form of a temporal-causal network model according to the standard format from Sect. 3.2. First is that now the reified network itself can also be reified again, in order to model second-order adaptation principles (as will be described in Chap. 4). Iteration of the reification step can only be done in a standard manner if every reification step makes a new temporal-causal network model and not an arbitrary complex dynamical system.

A second advantage is that mathematical analysis of equilibria can also be applied in a uniform manner, based on combination functions and the criterion on stationary points and equilibria formulated using them in Lemma 1 in Sect. 3.2.4. This allows such a mathematical analysis to relate emerging behaviour to properties of these combination functions. Chapters 11 to 14 are based on this, where Chaps. 11 and 12 address combination functions for base networks and Chaps. 13 and 14 address combination functions for reification states (for Hebbian learning and for bonding by homophily, respectively) at a first reification level. In all of these chapters, it is explored in general how certain relevant properties of the network structure entail certain properties of the emerging network behaviour. Aggregation characteristics as represented by combination functions are an essential element of the network structure; it turns out that such relevant properties of the network structure most often involve specific properties of the combination functions such as monotonicity and being scalar-free, sometimes in conjunction with some Connectivity properties such as the network being strongly connected.

More specifically, in all of these cases in Chaps. 11–14, analysis results were obtained of the uniform format that certain specific properties of the Aggregation characteristics as expressed by combination functions (sometimes together with certain properties of the network's Connectivity) entail certain properties of the network's emerging behaviour, mostly concerning equilibria that are reached. Typical properties of combination functions that are relevant for a base network are monotonicity and being scalar-free (see Chaps. 11 and 12). For combination functions describing bonding by homophily, a typical relevant property is having a tipping point for similarity where 'being alike' turns into 'not being alike' or conversely (see Chap. 13). Also for combination functions for Hebbian learning, some monotonicity properties turn out relevant for the emerging behaviour (see Chap. 14).

3.6 Using Network Reification for Unified Modeling of Network Adaptation Principles

In Sects. 3.4 and 3.5, it has been explained how the special effects of the downward causal connections for the different roles can be defined in a reified network, and how their contribution to a joint aggregated causal impact on base network states is specified by a generically defined universal combination function. This makes the reified network already work when each of the reification states has a constant value; it will then work just like a nonadaptive base network. However, availability of the reification states for the base network structure as explicit network states, which in principle can change over time, opens the possibility to define network adaptation principles in a Network-Oriented manner. This can be done by specifying

(a) The **connectivity** for reification states:

- proper causal connections to the reification states

(b) The **aggregation** for reification states:

- proper combination functions for them

(c) The **timing** for the reification states:

- proper speed factors for the adaptation of them

This is not just specification by an arbitrary separate set of difference or differential equations or procedural description as for the traditional hybrid approach discussed in Chap. 1, Sect. 1.4.1 and shown in Fig. 1.2. The reification perspective offers a framework to specify network adaptation principles in a more transparent, unified and standardized Network-Oriented manner, and compare them to each other.

This will be illustrated below for a number of examples of well-known network adaptation principles: Hebbian learning in Mental Networks and homophily in Social Networks, triadic closure in Mental and Social Networks, and preferential attachment in Mental and Social Networks. These examples of network adaptation principles in Sects. 3.6.1–3.6.4 all focus on adaptive connection weights. By far most of the network adaptation principles described in the literature only concern Connectivity: they address adaptive connections as adaptive network characteristics. However, in Sects. 3.6.5, 3.6.6 and 3.6.7 it will be shown how other adaptive network characteristics concerning Aggregation and Timing such as adaptive excitability, adaptive speed factors and adaptive combination functions can have interesting applications as well. For example, in a Social Network, response time may depend on external factors such as workload, which varies over time; this can be modeled by a speed factor that all the time adapts to this work load. Also in a Social Network, the way in which someone aggregates opinions from others may also change over time. For example, due to circumstances such as bad experiences or new higher management, a manager may change in how inputs from different employees are incorporated in his or her own opinions and decision making. These applications of speed factor reification and combination function reification will be illustrated in Sect. 3.7 more extensively by an example reified network model and example simulations.

So, next a number of adaptation principles known from the literature are addressed. It is shown how they can be modeled by network reification, and, in particular, their *connectivity* (in terms of their connections) and their *aggregation* (in terms of their combination functions), and the entailed *emerging equilibria* are described and analysed.

3.6.1 Network Reification for Adaptation Principles for Hebbian Learning and Bonding by Homophily

Hebbian learning (Hebb 1949) is based on the principle that strengthening of a connection between neurons over a period of time may take place when both states are often active simultaneously: 'neurons that fire together, wire together'. The principle itself refers to Hebb (1949), and over time has gained more interest in the area of computational modeling due to more extensive empirical support (e.g., Bi and Poo 2001), and more advanced mathematical formulations (e.g., Gerstner and Kistler 2002).

A different principle in a different domain, namely the *bonding based on homophily principle* in Social Science, has exactly the same graphical representation. This principle states that within a social network the more similar two persons are, the stronger their connection will become: 'birds of a feather flock together' (e.g., McPherson et al. 2001).

Connectivity of the reification states for the Hebbian learning and bonding by homophily principles
In Fig. 3.4 it is shown how the Hebbian learning principle can be modeled conceptually in a reified network by upward arrows to the reification states for the connection weights: each connection weight reification state is affected by the two connected states, for the sake of simplicity with connection weights 1. Moreover, a connection of $\mathbf{W}_{X_i,Y}$ to itself is assumed, also with weight 1. For bonding by homophily, the social network connection is similarly affected by the connected states as well with weights 1, like for Hebbian learning, so also in that case Fig. 3.4 applies.

So, both cases share the same connectivity. However, this does not hold for their aggregation: to model these two adaptation principles, the combination functions still are not the same, as the state values have different effects on the connection weights; this is addressed next.

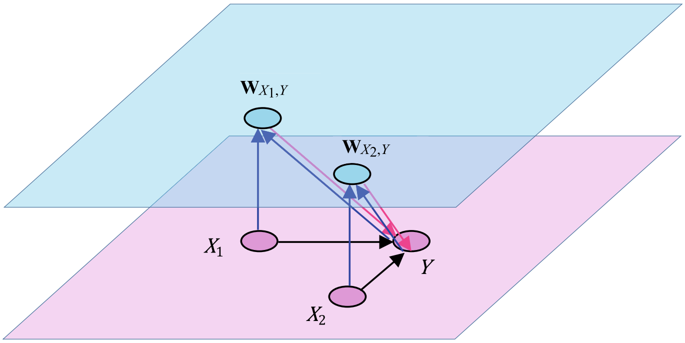

Fig. 3.4 Reified conceptual modeling of Hebbian learning in Mental Networks or Homophily in Social Networks

Aggregation by combination functions for the Hebbian learning and bonding by homophily principles
For Hebbian learning the combination function **hebb(..)** can be chosen, which is defined by

$$\mathbf{hebb}_{\boldsymbol{\mu}}(V_1, V_2, W) = V_1 V_2 (1 - W) + \boldsymbol{\mu} W \tag{3.12}$$

with $\boldsymbol{\mu}$ being the persistence parameter, where V_1 stands for $X_i(t)$, V_2 for $Y(t)$ and W for $\mathbf{W}_{X_i,Y}(t)$. This parameter describes in how far a learnt connection persists over time. Full persistence is indicated by $\boldsymbol{\mu} = 1$. If $\boldsymbol{\mu} < 1$, then some extent of extinction takes place; full extinction takes place for $\boldsymbol{\mu} = 0$. In the first part of the formula, the expression $V_1 V_2$ models the condition 'neurons that fire together', and the factor $(1 - W)$ takes care that the connection weight stays in the [0, 1] interval. For more options for Hebbian learning functions, see Chap. 14. Note that the function uses as third argument the current value W of the connection weight; this assumes that there is a connection from the reification state to itself, although in conceptual pictures such as in Fig. 3.4 such connections usually are not depicted; but they are specified in the role matrices. The same applies to many other reification states and adaptation principles.

For the bonding by homophily principle by (Blankendaal et al. 2016) or (Treur 2016), Chap. 11, Sect. 11.7, an option for the combination function is the *simple linear homophily function* $\mathbf{slhomo}_{\boldsymbol{\sigma},\boldsymbol{\tau}}$**(..)**:

$$\mathbf{slhomo}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1, V_2, W) = W + \boldsymbol{\alpha}(\boldsymbol{\tau} - |V_1 - V_2|)(1 - W)W \tag{3.13}$$

Here $\boldsymbol{\alpha}$ is the homophily modulation factor, and $\boldsymbol{\tau}$ the tipping point. Here the part $(\boldsymbol{\tau} - |V_1 - V_2|)$ models the condition 'birds of a feather': this part is positive if the difference between V_1 and V_2 is less than the tipping point $\boldsymbol{\tau}$ ('birds of a feather' is true) and negative when this difference is more than $\boldsymbol{\tau}$ ('birds of a feather' is false).

The factor $(1 - W)W$ takes care that W stays in the [0, 1] interval. As long as W is not 0 or 1, in the first case by the combination function a positive term is added to W, the combination function provides a value higher than W, so the connection weight will increase; in the second case a negative term is added, so the combination function provides a value lower than W, and the connection weight will decrease. In Chap. 13 more options for homophily functions are discussed.

Box 3.1 and 3.2 show mathematical analysis of emerging behaviour for these two adaptation principles.

Box 3.1 Mathematical analysis of emerging behaviour for the Hebbian learning principle

For a stationary point, applying the criterion (3.5) of Lemma 1 provides the following equilibrium equation for the above Hebbian learning combination function:

$$\begin{aligned} &W = \mathbf{hebb}_{\boldsymbol{\mu}}(V_1,\ V_2, W) = V_1V_2(1 - W) + \boldsymbol{\mu} W && \Leftrightarrow \\ &W = V_1V_2 - V_1V_2W + \boldsymbol{\mu} W && \Leftrightarrow \\ &W(1 + V_1V_2 - \boldsymbol{\mu}) = V_1V_2 && \Leftrightarrow \\ &W = \frac{V_1V_2}{1 - \boldsymbol{\mu} + V_1V_2} \end{aligned}$$

For example, when in an equilibrium both V_1 and V_2 have value 1, then $W = \frac{1}{2 - \boldsymbol{\mu}}$.

Box 3.2 Mathematical analysis of emerging behaviour for the bonding by homophily principle

For the above combination function for bonding based on homophily case, for a stationary point applying the criterion (3.5) of Lemma 1 provides the equilibrium equation:

$$\begin{aligned} &W = \mathbf{slhomo}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(V_1, V_2, W) = W + \boldsymbol{\alpha}(\boldsymbol{\tau} - |V_1 - V_2|)(1 - W)W && \Leftrightarrow \\ &\boldsymbol{\alpha}(\boldsymbol{\tau} - |V_1 - V_2|)(1 - W)W = 0 && \Leftrightarrow \\ &W = 0 \quad \text{or} \quad W = 1 \quad \text{or} \quad |V_1 - V_2| = \boldsymbol{\tau} \end{aligned}$$

As in simulations, for example, with the scaled sum combination function for the base states, the third option here often turns out to be not attracting, this indicates that in an equilibrium a form of clustering is achieved with connection weights 1 between states within one cluster and connection weights 0 between states in different clusters; see also Chap. 13.

3.6.2 Network Reification for the Triadic Closure Adaptation Principle

Another adaptivity principle is the *triadic closure principle* from Social Science (Rapoport 1953; Granovetter 1973; Banks and Carley 1996): If two persons in a social network have a common friend, then there is a higher chance that they will become friends themselves.

Connectivity of the reification states for the triadic closure adaptation principle
The connectivity for this adaptation principle is modeled conceptually in graph form as shown in Fig. 3.5.
Here horizontal arrows between the reification states describe the effect of triadic closure. The weights of these connections from $\mathbf{W}_{X,Y}$ and $\mathbf{W}_{Y,Z}$ to $\mathbf{W}_{X,Z}$ may be 1 for the sake of simplicity, but may also have different values, for example, to express that one of the two has more influence than the other one.

Aggregation by a combination function for the triadic closure principle
The combination function for $\mathbf{W}_{XZ}(t)$ can, for example, be a scaled sum:

$$\mathbf{ssum}_{\boldsymbol{\lambda}}(W_1, W_2) = \frac{W_1 + W_2}{\boldsymbol{\lambda}} \tag{3.14}$$

where W_1 indicates $\mathbf{W}_{X,Y}(t)$ and W_2 indicates $\mathbf{W}_{Y,Z}(t)$ and when the black horizontal arrows from $\mathbf{W}_{X,Y}$ and $\mathbf{W}_{Y,Z}$ to $\mathbf{W}_{X,Z}$ are assumed to have weight 1, the scaling factor $\boldsymbol{\lambda}$ can be normalised at 2. Alternatively, a higher-order Euclidean or logistic sum combination function might be used. For a $\boldsymbol{n}$th-order Euclidean combination function it becomes:

$$\mathbf{eucl}_{\boldsymbol{n},\boldsymbol{\lambda}}(W_1,\ W_2) = \sqrt[\boldsymbol{n}]{\frac{W_1^{\boldsymbol{n}} + W_2^{\boldsymbol{n}}}{\boldsymbol{\lambda}}} \tag{3.15}$$

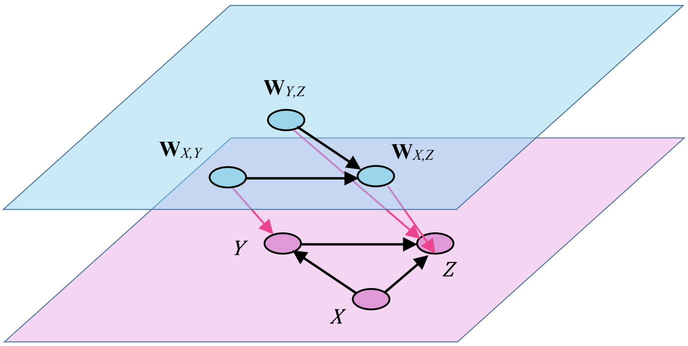

Fig. 3.5 Reified conceptual modeling of the triadic closure principle in Mental and Social Networks

There is also a counterpart of this principle in Mental Networks. It is a form of transitive closure which is implied indirectly by the Hebbian learning principle: strong connections from X to Y and from Y to Z will make more often X and Z active at the same time, and therefore their connection will become stronger by Hebbian learning. Box 3.3 shows mathematical analysis of emerging behaviour for this adaptation principle.

Box 3.3 Mathematical analysis of emerging behaviour for the triadic closure principle

For this case for a stationary point applying the criterion (3.5) of Lemma 1 provides the following linear equilibrium equation for $W = \mathbf{W}_{X,Z}(t)$:

$$W = \mathbf{ssum}_\lambda(W_1, W_2) = (W_1 + W_2)/\boldsymbol{\lambda} \Leftrightarrow$$

$$W = (W_1 + W_2)/\boldsymbol{\lambda}$$

For the Euclidean combination function, for a stationary point in the normalised case it holds

$$W = \sqrt{\frac{W_1^n + W_2^n}{\boldsymbol{\lambda}}}$$

3.6.3 *Network Reification for a Preferential Attachment Adaptation Principle*

Another principle from Social Science is the principle of *preferential attachment* (Barabasi and Albert 1999). This principle states that connections to states that already have more or stronger connections will become more strong.

Connectivity of the reification states for the preferential attachment adaptation principle

This can be modeled by horizontal connections between the reification states, which can be applied to multiple other connection weight reification states $\mathbf{W}_{X_i,Y}$; see Fig. 3.6. The weights of these horizontal connections can be 1 for the sake of simplicity, or any other values.

Aggregation by a combination function for the preferential attachment principle

The combination function $\mathbf{c}_{\mathbf{W}_{X_i,Y}}(..)$ for the considered reification state $\mathbf{W}_{X_i,Y}$ can, for example, be a scaled sum function.

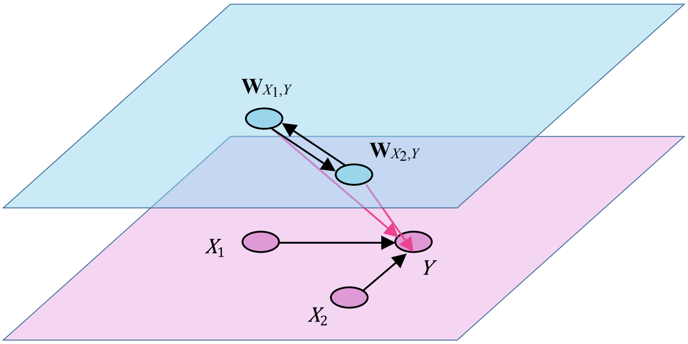

Fig. 3.6 Reified conceptual modeling of preferential attachment in Mental Networks and Social Networks

$$\mathbf{ssum}_{\boldsymbol{\lambda}}(W_1, \ldots, W_k) = \frac{W_1 + \ldots + W_k}{\boldsymbol{\lambda}} \tag{3.16}$$

where W_j is used for $\mathbf{W}_{X_j,Y}(t)$ and $\boldsymbol{\lambda}$ can be normalised at k. These $\mathbf{W}_{X_j,Y}$ represent the weights of all connections of base states X_j to Y, from which the considered X_i is one. Alternatively, a higher order Euclidean or logistic sum combination function might be used here. For a nth order Euclidean combination function it becomes

$$\mathbf{eucl}_{\boldsymbol{n},\boldsymbol{\lambda}}(W_1, \ldots, W_k) = \sqrt[n]{\frac{W_1^{\boldsymbol{n}} + \ldots + W_k^{\boldsymbol{n}}}{\boldsymbol{\lambda}}} \tag{3.17}$$

Also a logistic sum combination function can be used, in which case a higher number k of connections to Y more clearly leads to a higher weight $\mathbf{W}_{X_i,Y}$.

This principle has a counterpart in Mental Networks: for cases that X_1 and X_2 are conceptually related so that they often are activated in the same situations, a stronger connection from X_1 to Y leads to more activation of Y and by Hebbian learning also to a stronger connection from X_2 to Y.

Box 3.4 shows a mathematical analysis of emerging behaviour for this adaptation principle.

Box 3.4 Mathematical analysis of emerging behaviour for the preferential attachment principle
For the scaled sum case, for a stationary point applying the criterion (3.5) of Lemma 1 provides the following linear equation for $W = \mathbf{W}_{X_i,Y}(t)$:

$$W = \mathbf{ssum}_{\boldsymbol{\lambda}}(W_1, \ldots, W_k) = \frac{W_1 + \ldots + W_k}{\boldsymbol{\lambda}} \quad \Leftrightarrow$$
$$W = \frac{W_1 + \ldots + W_k}{k}$$

So the connection weight $\mathbf{W}_{X_i,Y}$ gets the average value of the weights $\mathbf{W}_{X_j,Y}$ for all j.
For the Euclidean case, a stationary point in the normalised case it holds

$$W = \sqrt[n]{\frac{W_1^n + \ldots + W_k^n}{k}}$$

3.6.4 Network Reification for the State-Connection Modulation Adaptation Principle

Yet another adaptation principle that applies both to Mental and Social Networks is the principle of *state- connection modulation*.

Connectivity of the reification states for the state-connection modulation adaptation principle
This can be modeled conceptually by upward arrows from control states Z_i in the base network to the reification states of connection weights; see Fig. 3.7. The weights of these upward connections may be 1 for the sake of simplicity, or any other value.

For a Mental Network Z_i can be a state of extreme stress (Sousa et al. 2012) or a chemical or medicine (e.g., a neurotransmitter). For a counterpart in a Social Network, Z_i can be a measure for the intensity of the actual interaction (e.g., taking into account frequency and emotional charge); this can be called the *interaction connects* principle (e.g., Treur 2016), Chap. 11.

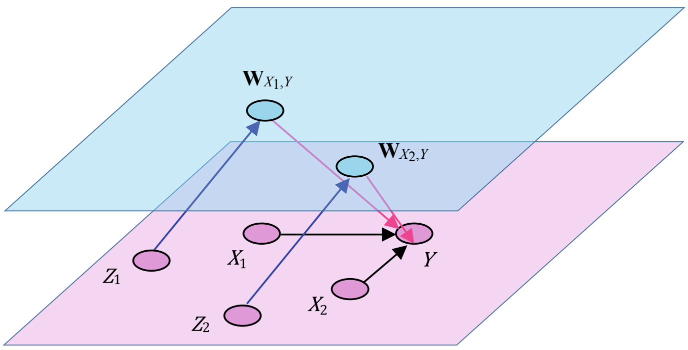

Fig. 3.7 Reified conceptual modeling of state-connection modulation by control state Z_i in a Mental Network or of the interaction connects principle in Social Network with state Z_i the intensity of the interaction

Aggregation by a combination function for the state-connection modulation principle
In this case the combination function $\mathbf{c}_{\mathbf{W}_{X_i,Y}}(..)$ for $\mathbf{W}_{X_i,Y}$ can, for example, be the state-connection modulation function $\mathbf{scm}_{\alpha}(..)$:

$$\mathbf{scm}_{\alpha}(W, V) = W + \alpha VW(1 - W) \tag{3.18}$$

with α a modulation factor, which can be positive (amplifying effect) or negative (suppressing effect), where V is used for for $Z_i(t)$ and W for $\mathbf{W}_{X_i,Y}(t)$; for an application of this, see also Chap. 5 or (Treur and Mohammadi Ziabari 2018). Note that when this state-connection modulation function $\mathbf{scm}_{\alpha}(W, V)$ is used in combination with Hebbian learning, auxiliary variables V_1, V_2 (which are not actually used by the function) are included in this function, making it $\mathbf{scm}_{\alpha}(V_1, V_2, W, V)$ to get one shared sequence of values used by both functions. Box 3.5 shows a mathematical analysis of emerging behaviour for this adaptation principle.

Box 3.5 Mathematical analysis of emerging behavior for the state-connection modulation principle
For this case for a stationary point applying the criterion (3.5) of Lemma 1 provides the following quadratic equation for $W = \mathbf{W}_{X_i,Y}(t)$:

$$\begin{aligned} &W = \mathbf{scm}_{\alpha}(W, V) = W + \alpha VW(1 - W) \quad \Leftrightarrow \\ &\alpha VW(1 - W) = 0 \quad \Leftrightarrow \\ &V = 0 \quad \text{or} \quad W = 0 \quad \text{or} \quad W = 1 \end{aligned}$$

3.6.5 *Network Reification for Excitability Adaptation Principles*

Next, a number of adaptation principles are addressed that are not related to connection weights. The first concerns excitation adaptation principles. In Chandra and Barkai 2018) this is explained as follows:

> 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)

To address dynamic levels of excitability of base states, for a base state Y a logistic sum combination function is assumed, which has a threshold parameter τ. Decreasing the value of τ is increasing excitability of Y, as a lower threshold value will make that Y becomes more activated. Then a reification state $\mathbf{T}_Y$ can be included that represents the intrinsic excitability of Y, by the value of the threshold parameter τ_Y of its logistic sum combination function.

Connectivity of the reification state for the excitability adaptation principle
In Fig. 3.8 the basic connectivity pattern is shown that can be used to model excitability adaptation principles by network reification. Here c is a context factor that affects the excitability. Such a pattern is used in Chap. 4 as part of a more complex (multilevel) reified network model. The weights of the upward connections may be 1 for the sake of simplicity, or any other value.

Aggregation by a combination function for adaptive excitability
For this reification state $\mathbf{T}_Y$, a logistic sum combination function can be used, or an Euclidean function, for example, or any other specific form. In Chap. 4 the advanced logistic sum combination function is used for $\mathbf{T}_Y$; see that chapter for more details.

3.6.6 Network Reification for Response Speed Adaptation Principles

A person will not always respond to inputs with the same speed. Examples of factors affecting the response speed are workload (negative influence) or the availability of support staff (positive influence). A slightly different application is the presence of certain chemicals in the brain to stimulate or slow down transfer between neurons (for example, the effect of alcohol or a stress-suppressing medicine on reaction time).

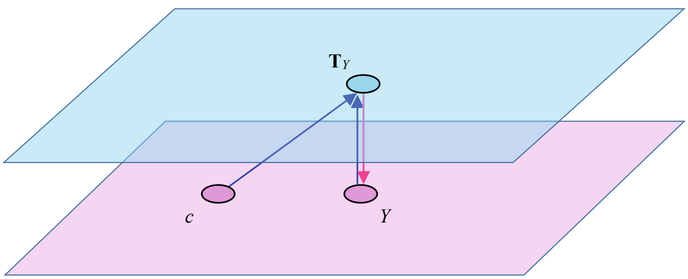

Fig. 3.8 Reified network model for excitability adaptation principles

Connectivity of the reification state for speed adaptation
The connectivity is modeled in Fig. 3.9 by two conditions Z_1 and Z_2 and their positive and negative connections to the speed factor reification $\mathbf{H}_Y$. The weights of the upward connections may be 1 for the sake of simplicity, or any other value.
Aggregation by a combination function for speed adaptation
For this, the combination function $\mathbf{c}_{\mathbf{H}_Y}(..)$ can be modeled by the scaled sum combination function

$$\mathbf{ssum}_{\lambda}(V_1, V_2) = \frac{V_1 + V_2}{\boldsymbol{\lambda}} \tag{3.19}$$

where V_1 stands for $Z_1(t)$, and V_2 for $Z_2(t)$, and assuming the upward connections have weights 1, $\boldsymbol{\lambda}$ can be normalised at 2. Also here alternative options can be used such as ***n***th-order Euclidean combination functions or logistic combination functions. Box 3.6 shows mathematical analysis of emerging behaviour for this adaptation principle.

Box 3.6 Mathematical analysis of emerging behavior for adaptive speed factors
For the normalised case for a stationary point applying the criterion (3.5) of Lemma 1 provides the following linear equation for $H = \mathbf{H}_Y(t)$ in relation to the state values V_1 for Z_1 and V_2 for Z_2

$$H = 1/2(V_1 + V_2)$$

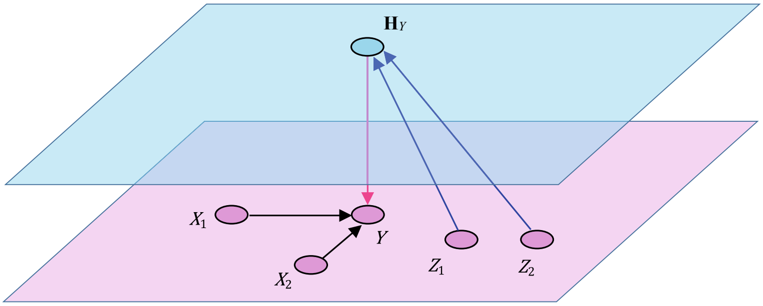

Fig. 3.9 Reified conceptual modeling of an adaptive speed factor under influence of two states Z_1 and Z_2

3.6.7 Network Reification for Aggregation Adaptation Principles

The following is an example of adaptive aggregation by adaptive combination functions. Suppose a manager wants to represent the opinions of her employees well within the organisation. She initially supports any proposal of any single individual employee about a certain issue. This can be modeled by the $\mathbf{smax}_\lambda$**(..)** combination function: if one of the input opinions is high, also the manager's opinion will become high. After bad experiences within the organisation, she may gradually move to a different way, based on averaging over the opinions of her group of employees, which can be modeled by a normalised scaled sum $\mathbf{ssum}_\lambda$**(..)** with $\boldsymbol{\lambda}$ as the sum of the weights of the incoming connections. Or eventually, she can decide only to support an idea when all employees share that opinion. This can be modeled by the $\mathbf{smin}_\lambda$**(..)** combination function. So, the following transitions can take place over time (gradually):

$$\mathbf{smax}_\lambda(..) \rightarrow \mathbf{ssum}_\lambda(..) \rightarrow \mathbf{smin}_\lambda(..)$$

This may look a bit extreme, but at least makes the idea clear. A more realistic version may be the following. First the manager aggregates the incoming opinions by averaging over the group, using the normalised scaled sum $\mathbf{ssum}_\lambda$**(..)**, but later on she gradually moves to using a logistic sum combination function $\mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}$**(..)** where she applies a certain threshold $\boldsymbol{\tau}$ before she supports the opinion:

$$\mathbf{ssum}_\lambda(..) \rightarrow \mathbf{alogistic}_{\boldsymbol{\sigma},\boldsymbol{\tau}}(..)$$

Connectivity of the reification state for the aggregation adaptation principle
A picture of a graphical representation of a network model for this is shown in Fig. 3.10. An example simulation for this scenario can be found in Sect. 3.7; see

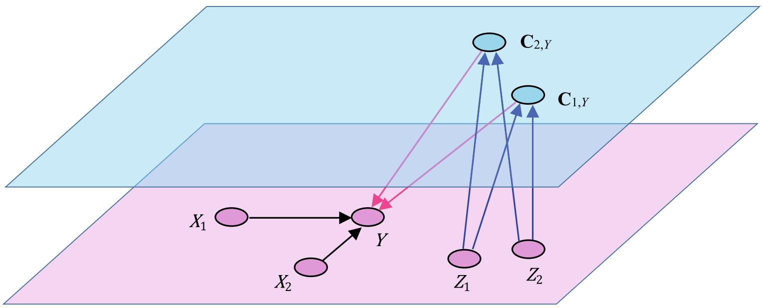

Fig. 3.10 Reified conceptual modeling of adaptive combination functions under influence of states Z_i

Figs. 3.12 and 3.13. Here the basic combination function $\text{bcf}_1(..)$ relating to $\mathbf{C}_{1,Y}$ is $\mathbf{ssum}_\lambda\mathbf{(..)}$ and $\text{bcf}_2(..)$ relating to $\mathbf{C}_{2,Y}$ is $\mathbf{alogistic}_{\sigma,\tau}\mathbf{(..)}$. Some condition Z_1 in the organisation starts to affect $\mathbf{C}_{1,Y}$ and $\mathbf{C}_{2,Y}$ by suppressing the former (negative connection weight) and increasing the latter (positive connection weight). This will make the above transition from $\mathbf{ssum}_\lambda\mathbf{(..)}$ to $\mathbf{alogistic}_{\sigma,\tau}\mathbf{(..)}$ happen. This scenario will be addressed in more detail in Sect. 3.7.

An example of adaptive combination functions for a Mental Network can be found by considering multicriteria decision-making. The valuations for the different criteria are aggregated according to some function. A person may adapt this function over time due to learning. For example, first a $\mathbf{smin}_\lambda\mathbf{(..)}$ function is used to model that all criteria should be fulfilled to get a high overall score, but later, after it was experienced that based on that function almost no decisions were made, a $\mathbf{ssum}_\lambda\mathbf{(..)}$ function is used to model that the decision should be based on the average of the valuations for the different criteria.

Aggregation by a combination function for aggregation adaptation
Suppose $Z_1, \ldots, Z_k$ are the states affecting the combination function reification states $\mathbf{C}_{i,Y}$. Note that some of their impacts can be positive and some can be negative according to the sign of their connection weight $\boldsymbol{\omega}_{Z_j,\mathbf{C}_{i,Y}}$ to $\mathbf{C}_{i,Y}$, and from the aggregation of their impacts it depends whether $\mathbf{C}_{i,Y}(t)$ will increase or decrease with t. A first option for the combination function $\mathbf{c}_{\mathbf{C}_{i,Y}}\mathbf{(..)}$ is a scaled sum combination function to aggregate the impacts of $Z_1, \ldots, Z_k$ on $\mathbf{C}_{i,Y}$:

$$\mathbf{ssum}_\lambda(V_1,\ldots,V_k) = \frac{V_1 + \ldots + V_k}{\lambda} \tag{3.20}$$

where each V_j indicates $\boldsymbol{\omega}_{Z_j,\mathbf{C}_{i,Y}}\, Z_j(t)$. Alternatively, a higher order Euclidean or logistic sum combination function might be used here. For an nth-order Euclidean combination function it becomes

$$\mathbf{eucl}_{n,\lambda}(V_1,\ldots,V_k) = \sqrt[n]{\frac{V_1^n + \ldots + V_k^n}{\lambda}} \tag{3.21}$$

Box 3.7 shows a mathematical analysis of emerging behaviour for this adaptation principle. In Sect. 3.7 a more extensive example illustrates this.

Box 3.7 Mathematical analysis of emerging behavior for combination function adaptation
For a scaled sum combination function, by the criterion (3.5) of Lemma 1 the following equation for a stationary point is obtained (where $C = \mathbf{C}_{i,Y}(t)$):

$$C = \frac{V_1 + \ldots + V_k}{\boldsymbol{\lambda}}$$

where each V_j indicates the value $\boldsymbol{\omega}_{Z_j,\mathbf{C}_{i,Y}}\ Z_j(t)$. Similarly for an $\boldsymbol{n}$th order Euclidean combination function:

$$C = \sqrt[n]{\frac{V_1^{\boldsymbol{n}} + \ldots + V_k^{\boldsymbol{n}}}{\boldsymbol{\lambda}}}$$

3.7 A Reified Network Model for Response Speed Adaptation and Aggregation Adaptation

In this section, following Sects. 3.6.6 and 3.6.7 above, by an example scenario, the use of an adaptive speed factor and an adaptive combination function in a reified network is illustrated. Also, the network's emerging behaviour based on equilibrium values will be analysed. Consider, as also discussed in Sect. 3.6.6, within an organisation a manager of a group of 7 members with their opinions X_1, …, X_7. The adaptation focuses on the manager opinion; the manager adapts to the organization over time. She wants to represent the opinions of the group members well within the organization and therefore she initially uses a (normalized) scaled sum function $\mathbf{ssum}_\lambda$**(..)** to aggregate the opinions to some average. However, later on based on disappointing experiences within the organization, she decides to use a threshold τ through the logistic sum combination function $\mathbf{alogistic}_{\sigma,\tau}$**(..)**. Moreover, initially she is busy with other things and only later she gets more time to respond faster on the input she gets from her group members, so that her speed factor increases from that time point on.

3.7.1 Conceptual Graphical Representation of the Example Reified Network Model

In Fig. 3.11 an overview of this reified network is shown; see also Table 3.5 for the states and their explanation. Note that also group members X_1, …, X_7 have mutual connections, but this is not shown in the graph in Fig. 3.11 to keep the picture simple; see Box 3.8 for these connections. For the same reason the upward connections from all group members to combination function reification state $\mathbf{C}_{1,Y}$ have

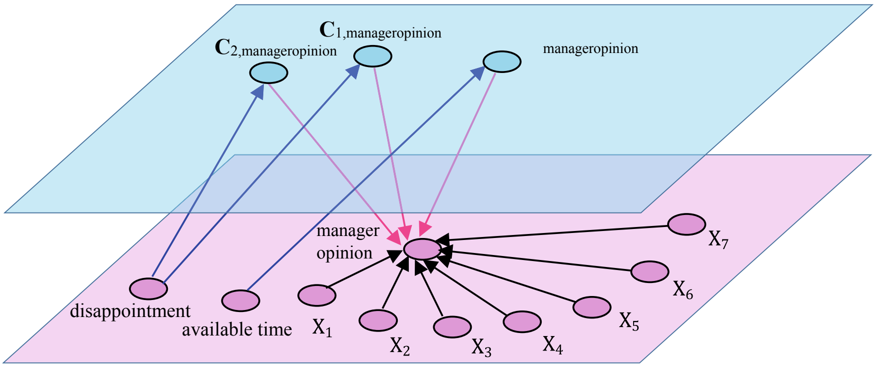

Fig. 3.11 The considered example adaptive reified network model

Table 3.5 States and their explanation

State		Explanation	Level
X_1			
X_2			
X_3			
X_4		These 7 states X_1 to X_7 represent the opinions of the 7 group members	
X_5			Base level
X_6			
X_7			
X_8	managеropionion	The opinion of the manager	
X_9	availabletime	The available time of the manager	
X_{10}	disappointment	The level of disappointment of the manager	
X_{11}	$\mathbf{H}$manageropinion	The reified representation state for the speed of the manager's opinion	
X_{12}	$\mathbf{C}$1,manageropinion	The reified representation state for the weight for the scaled sum combination function to aggregate the group members opinions for the manager's opinion	First reification level
X_{13}	$\mathbf{C}$2,manageropinion	The reified representation state for the weight for the advanced logistic sum combination function to aggregate the group members opinions for the manager's opinion	

been left out; see Box 3.8 for them. Also, the connection from the independent state 'available time' to itself has been left out, and the same for independent state 'disappointment' representing the disappointing experiences; also see Box 3.8 for that. The reification states with their speed factors and combination functions, and connection weights for the incoming connections used to model this scenario are specified as shown in Box 3.8. Note that it is assumed:

$$\mathrm{bcf}_1(..) = \mathbf{ssum}_{\lambda}(..)$$
$$\mathrm{bcf}_2(..) = \mathbf{alogistic}_{\sigma,\tau}(..)$$

So, in the scenario the combination function used for the manager opinion state will change over time from basic combination function $\mathrm{bcf}_1(..)$ to $\mathrm{bcf}_2(..)$, using the reification states $\mathbf{C}_{1,\mathrm{manageropinion}}$ and $\mathbf{C}_{2,\mathrm{manageropinion}}$.

The concept of a *role matrix* as introduced in Chap. 2, Sect. 2.4 is used in a generalised form here to describe a reified network's structure. For the role matrices for this example network model, see Box 3.8. As can be seen, the main difference with the role matrices in Chap. 2 is that now in the cells in all role matrices there can be either a value (shaded in green) or the name of a state (shaded in red), whereas in Chap. 2 only values were used in the role matrices **mcw**, **ms**, **mcfw**, and **mcfp**. This is a relatively small step in terms of writing the role matrices, but it drastically changes the possibilities to model adaptive networks, as it makes that a non-reified network becomes a reified network. Indicating not a value but a state name in one of the cells, makes that the network characteristic described by this cell becomes adaptive and the indicated state becomes a reification state for this characteristic. For example by writing X_{11} in the cell for manager opinion in the speed factor role matrix **ms** in Box 3.8, the speed of change of the manager's opinion becomes adaptive according to the (dynamic) value of state X_{11}, and this makes X_{11} a reification state for this adaptive speed factor (and therefore X_{11} also is given the more informative name $\mathbf{H}_{\text{manageropinion}}$).

For the example reified network in the simulation the group members have mutual connections as specified by the role matrices **mb** and **mcw** in Box 3.8. This models a form of social contagion in a strongly connected network from which it is known that it eventually leads to a joint opinion; e.g., see Chap. 11. Note that in role matrices **mb** and **mcw** the cells correspond to all connections that in the picture in Fig. 3.11 are depicted by *upward or horizontal arrows*. There are no cells for the (pink) downward arrows in these two matrices, as these arrows concern special effects, and automatically get weight value 1 to keep the relation between a base network characteristic (as used in the dynamics) and its reification one-to-one. Therefore, the downward arrows from reification states are described in a different way in one of the other role matrices depending on which role these reification states describe, and in the cell of the concept they are reifying; see Table 3.6.

3.7.2 *Conceptual Role Matrices Representation of the Example Reified Network Model*

The role matrices for the example reified network model are shown in Box 3.8 together with the initial values in **iv**. Note that the independent dynamics of each of the states available time and disappointment, which serve as an external input to the model, were modeled by a logistic sum combination function applied to the connection of the state to itself with specific settings (steepness 18 and threshold 0.2) shown in role matrices **mcfw** and **mcfp**.

Table 3.6 Downward causal connections and role matrices: how and where specify what

In model picture	State name	State number	Role	In role matrix
Downward arrow from a reification state for an adaptive connection weight from state X to state Y	$\mathbf{W}_{X,Y}$	X_i	Connection weight reification state for $\boldsymbol{\omega}_{X,Y}$	**mcw** as notation X_i in the cell for the weight $\boldsymbol{\omega}_{X,Y}$ of the connection from X to Y
Downward arrow from a reification state for an adaptive speed factor for state Y	$\mathbf{H}_Y$	X_j	Speed factor reification state for $\boldsymbol{\eta}_Y$	**ms** as notation X_j in the cell for the value $\boldsymbol{\eta}_Y$ of the speed factor of Y
Downward arrow from a reification state for an adaptive combination function weight for state Y	$\mathbf{C}_{i,Y}$	X_k	Combination function weight reification state for $\boldsymbol{\gamma}_{i,Y}$	**mcfw** as notation X_k in the cell for the weight $\boldsymbol{\gamma}_{i,Y}$ of combination function i for state Y
Downward arrow from a reification state for an adaptive combination function parameter for state Y	$\mathbf{P}_{i,j,Y}$	X_l	Combination function parameter reification state for $\boldsymbol{\pi}_{i,j,Y}$	**mcfp** as notation X_l in the cell for the value $\boldsymbol{\pi}_{i,,j,Y}$ of parameter j of combination function i for state Y

3.7.3 Simulation Outcomes for the Example Reified Network Model

In Fig. 3.12 it is shown how in the simulation the manager's speed factor and combination function weights adapt over time, and in Fig. 3.13 the base states are shown: the group member opinions, the manager's opinion, and the change in

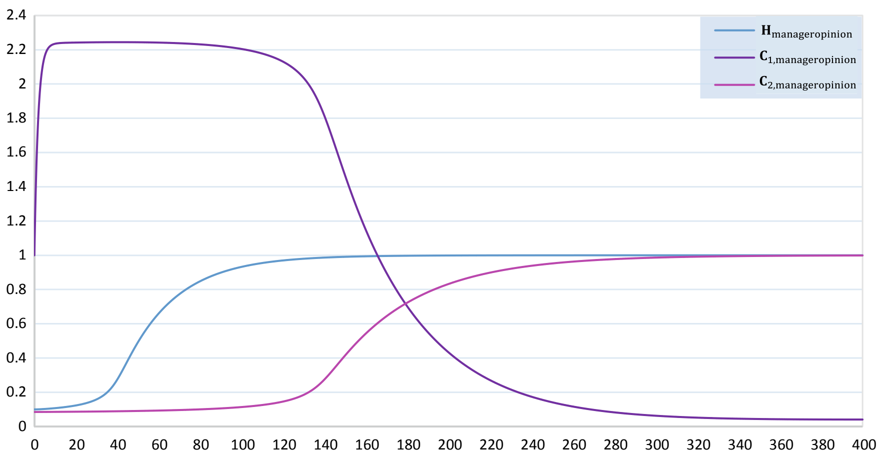

Fig. 3.12 Reified adaptive speed factor and combination function weights for the manager opinion state

available time and in disappointment. Note that this is one of those less usual cases in which state values can be outside the [0, 1] interval. That can also be modelled; in particular, in case of reification states for some role where values are used only for the special effects for that role. In this case, within the combination function a division by the sum of the weights takes place so that finally everything still comes in the [0, 1] interval. As an alternative, for reification state $\mathbf{C}_{1,\text{manageropinion}}$ a logistic sum combination function could have been used instead of a Euclidean function. Then the values would have stayed in the [0, 1] interval all the time.

Box 3.8 Role matrices for the example reified network model

mb	base connectivity	1	2	3	4	5	6	7	8
X_1		X_2	X_4	X_7					
X_2		X_1	X_3	X_5					
X_3		X_2	X_6						
X_4		X_2	X_7						
X_5		X_4	X_7						
X_6		X_1	X_3						
X_7		X_1	X_4						
X_8	manageropionion	X_1	X_2	X_3	X_4	X_5	X_6	X_7	
X_9	availabletime	X_9							
X_{10}	disappointment	X_{10}							
X_{11}	**H**manageropinion	X_9							
X_{12}	**C**1,manageropinion	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_{10}
X_{13}	**C**2,manageropinion	X_{10}							

mcw	connection weights	1	2	3	4	5	6	7	8
X_1		0.5	0.6	0.2					
X_2		0.5	0.6	0.5					
X_3		0.6	0.7						
X_4		0.3	0.7						
X_5		0.4	0.6						
X_6		0.6	0.4						
X_7		0.3	0.7						
X_8	manageropionion	1	1	1	1	1	1	1	
X_9	availabletime	1							
X_{10}	disappointment	1							
X_{11}	**H**manageropinion	1							
X_{12}	**C**1,manageropinion	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.05
X_{13}	**C**2,manageropinion	1							

mcfw	combination function weights	1 eucl	2 alogistic
X_1		1	
X_2		1	
X_3		1	
X_4		1	
X_5		1	
X_6		1	
X_7		1	
X_8	manageropionion	X_{12}	X_{13}
X_9	availabletime		1
X_{10}	disappointment		1
X_{11}	**H**manageropinion	1	
X_{12}	**C**1,manageropinion	1	
X_{13}	**C**2,manageropinion	1	

mcfp	combination function	1 eucl		2 alogistic	
	parameter	1 n	2 λ	1 σ	2 τ
X_1		1	1.3		
X_2		1	1.6		
X_3		1	1.3		
X_4		1	1		
X_5		1	1		
X_6		1	1		
X_7		1	1		
X_8	manageropionion	1	7	5	5.5
X_9	availabletime			18	0.2
X_{10}	disappointment			18	0.2
X_{11}	**H**manageropinion	1	1		
X_{12}	**C**1,manageropinion	1	0.02		
X_{13}	**C**2,manageropinion	1	1		

ms	speed factors	1
X_1		0.05
X_2		0.05
X_3		0.05
X_4		0.05
X_5		0.05
X_6		0.05
X_7		0.05
X_8	manageropionion	X_{11}
X_9	availabletime	0.04
X_{10}	disappointment	0.025
X_{11}	**H**manageropinion	0.5
X_{12}	**C**1,manageropinion	0.5
X_{13}	**C**2,manageropinion	0.5

iv	initial values	1
X_1		1
X_2		0.9
X_3		0.5
X_4		0.6
X_5		0.6
X_6		0.6
X_7		0.7
X_8	manageropionion	0.5
X_9	availabletime	0.1
X_{10}	disappointment	0.085
X_{11}	**H**manageropinion	0.1
X_{12}	**C**1,manageropinion	1
X_{13}	**C**2,manageropinion	0.085

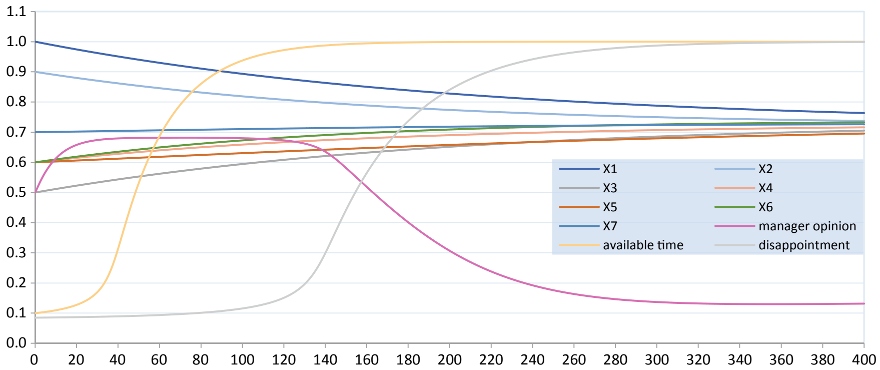

Fig. 3.13 The effect of the adaptive combination function of the manager on her opinion

It can be seen in Fig. 3.12 that after time point 40 the manager's speed factor increases (blue line; with as effect a shorter response time), due to more availability (the purple line in Fig. 3.12). After time point 140 in Fig. 3.12 a switch is shown from a dominant weight for the scaled sum function $\mathbf{ssum}_7\mathbf{(..)}$ (purple line) to a dominant weight for the logistic combination function $\mathbf{alogistic}_{5,5.5}(\mathrm{x})$ (red line), due to increasing disappointment (green line in Fig. 3.12). In Fig. 3.13 it is also shown how the manager's opinion is affected by the opinions of the group members. Here it can be seen that after time point 140 the manager's opinion becomes much lower due to the switch of combination function, which is resulting from the increase in disappointment (green line).

3.7.4 Analysis of the Equilibria for the Example Reified Network Model

Here the equilibrium equations for the different states are considered. First, the independent base states, next the reification states which depend on the independent states.

Equilibrium equations for the independent base states

Both independent base states have a circular causal relation and use the same combination function $\mathbf{alogistic}_{18,0.2}(..)$. From the criterion (3.5) in Lemma 1 it is derived that their equilibrium equations are:

$$\begin{aligned} \mathrm{availabletime} &= \mathbf{alogistic}_{18,0.2}(\mathrm{availabletime}) \\ \mathrm{disappointment} &= \mathbf{alogistic}_{18,0.2}(\mathrm{disapppointment}) \end{aligned} \quad (3.22)$$

So both have an equation of form x = $\mathbf{alogistic}_{18,0.2}(x)$. In Fig. 3.14 the graphs of both the functions x and $\mathbf{alogistic}_{18,0.2}(x)$ are shown. Given that the equilibrium values are not close to 0, it can be seen that they cross indeed very close to 1; the value can be approximately calculated in 12 digits as: x = 0.999999427374. This differs from 1 less than 10^{-6}. This applies both to available time and to disappointment. Indeed in Fig. 3.14 it is shown that in the simulation they end up very close to 1.

Equilibrium equations for the reification states

Applying the criterion (3.5) in Lemma 1 to the above specifications in Box 3.8, directly provides the equilibrium equations for the reification states by the following relations where H stands for the value of $\mathbf{H}_{\text{manageropinion}}$, C_1 for the value of $\mathbf{C}_{1,\text{manageropinion}}$ and C_2 for the value of $\mathbf{C}_{2,\text{manageropinion}}$:

$$\begin{aligned} H &= \text{availabletime} \\ C_1 &= 0.5\,X_1 + 0.5\,X_2 + 0.5\,X_3 + 0.5X_4 + 0.5\,X_5 \\ &\quad + 0.5X_6 + 0.5X_7 - 2.5\text{disapppointment} \\ C_2 &= \text{disapppointment} \end{aligned} \tag{3.23}$$

Assuming that in the equilibrium it holds $X_i(t)$ = 0.714 for all i = 1, …, 7 (this depends on the specification of the group members) the second equation can be rewritten as:

$$C_1 = 2.5 - 2.5\ \text{disapppointment} = 2.5\,(1 - \text{disapppointment}) \tag{3.24}$$

These equilibrium equations show how the equilibrium values of the reification states depend on the equilibrium values of the independent base states available time and disappointment. As discussed above, the equilibrium value for available time and disappointment are very close to 1 (difference from 1 less than 10^{-6}),

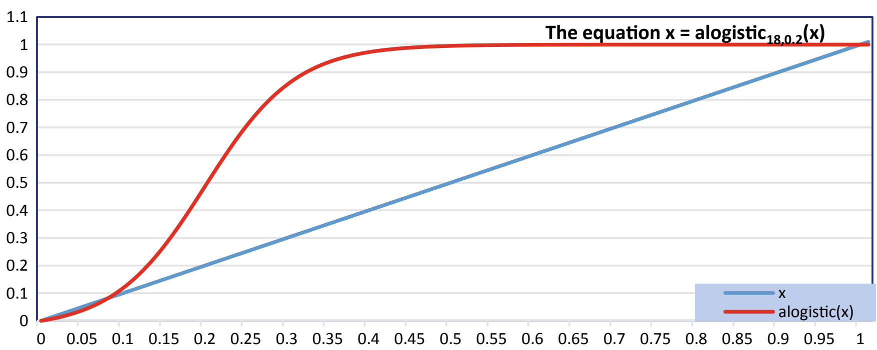

Fig. 3.14 Solving the equilibrium equations for the two independent states availble time and disappointment

therefore H and C_2 will also be very close to 1, which is confirmed in the simulation as shown in Fig. 3.12. Moreover, C_1 will be very close to 0 (by (3.24); difference from 0 less than 10^{-5}). This shows that the combination function for manager opinion indeed switched from the initial scaled sum **ssum**$_7$**(..)** to the logistic **alogistic**$_{5,5.5}$**(..)**; see how this is also confirmed in Figs. 3.12 and 3.13. Note that in Chap. 15, Sect. 15.3 more details are given of the difference and differential equations for this reified network model.

3.8 On Using a Joint Reification State for Multiple Base States or Multiple Roles

Usually a reification state has one role and is associated to one base state. This also supports transparency. However there may be specific cases in which one state can play the role of reification state for multiple states. There could even be situations in which one state plays multiple roles with respect to a given base state or multiple base states, but better be very careful to model like this. Be aware that the value of a reification state is where the characteristic related to its role is maintained, and this can only be one value. Therefore having a joint reification state for multiple base states only makes sense as all these base states should keep a joint value for that characteristic, for example, all have the same speed factor. As soon as there are differences in these values, the reification state has to split according to these differences. This will in general also stand in the way to use one reification state for multiple roles. For example, then the value of this reification state represents both a speed factor and a connection weight. As these are totally different concepts, it will, in general, not make sense to have the same value for them. Keeping this in mind, some examples in which joint reification states might make sense are as follows:

Mental Networks
Chemicals in the brain such as neurotransmitters or drugs or hormones or alcohol or stress related elements with a global effect on the brain:

- Slowing down or speeding up activation of brain states
  (adaptive joint speed factor)
- Lowering thresholds of brain states
  (adaptive joint threshold value for a logistic combination function)

Social networks
Social events that affect multiple persons at the same time:

- Due to a joint meeting no-one responds fast to messages or mail
  (adaptive joint speed factor)
- Within the meeting the threshold to respond publicly is higher
  (adaptive joint threshold value for a logistic combination function)

- Due to positive atmosphere in a meeting the scaling factor is lower, due to negative atmosphere scaling factor is higher
  (adaptive joint threshold value for a logistic combination function)
- Due to positive atmosphere the threshold to accept/assimilate opinions is lower, due to negative atmosphere the threshold to accept is higher`
  (adaptive joint threshold value for a logistic combination function).

But, as indicated, normally there are differences between individuals and between states, and to represent these differences, multiple reification states are needed.

3.9 On the Added Complexity for Reification

Note that, as for any dynamical system, by adding adaptivity to a network model, always complexity is added. In this section, it is discussed how the complexity of a network increases when reification is applied. It will at most be quadratic in the number of nodes N and linear in the number of connections M of the original network, as shown here. More specifically, if m the number of basic combination functions used in the given network model, then the number of *nodes* in the reified network is at most:

$$\begin{array}{ll} N & \text{(original nodes)} \\ +N & \text{(nodes for speed factors)} \\ +N^2 & \text{(nodes for connection weights)} \\ +mN & \text{(nodes for combination functions)} \end{array}$$

which adds to

$$(2+m+N)N \tag{3.25}$$

This is quadratic in the number of nodes.

If more general, not all N^2 connections are used for reification, but only a number M of them (which maximally can be N^2), the outcome is

$$(2+m)N+M \tag{3.26}$$

additional nodes. This is linear in the numbers of nodes and connections. Then the number of *connections* in the reified network is

$$\begin{array}{ll} M & \text{(original connection weights)} \\ +N & \text{(speed factors to their base states)} \\ +\sum_Y \text{indegree}(Y) = M & \text{(connection weights to their base states)} \\ +mN & \text{(combination function weights to their base states)} \end{array}$$

which adds to

$$(m+1)N + 2M \tag{3.27}$$

Also, this is linear in number of nodes and connections. Note, however, that also connections to the reification states will be needed to get them adaptive. But these depend on the specific application. If at least one inward connection per reification state is assumed, this adds at least the number of additional nodes $(2 + m)N + M$ to the number of added connections. So, then the number of additional connections becomes

$$(2m+3)N + 3M \tag{3.28}$$

which again is linear in numbers of nodes and connections.

3.10 Discussion

In this chapter, it was shown how network structure can be reified in a network by adding explicit network states representing the characteristics of the network structure, such as connection weights, combination functions and speed factors. Parts of this chapter were adopted from (Treur 2018a). This construction of network reification can provide advantages similar to those found for reification in modeling and programming languages in other areas of AI and Computer Science, in particular, substantially enhanced expressiveness (e.g., Weyhrauch 1980; Bowen and Kowalski 1982; Bowen 1985; Sterling and Shapiro 1986; Sterling and Beer 1989; Demers and Malenfant 1995; Galton 2006).

A reified network including an explicit representation of the network structure enables to model dynamics *of* the original network by dynamics *within* the reified network. In this way, an adaptive network can be represented by a non-adaptive network. The approach is applicable to the whole variety of adaptive processes as described in Chap. 1, Sect. 1.2.1. It was shown how the approach provides a unified manner of modeling network adaptation principles, and allows comparison of such principles across different domains. This was illustrated for known adaptation principles for Mental Networks and for Social Networks. Note that this approach to model network adaptation principles can be applied successfully to any adaptation principle that is described by (first-order) difference or differential equations (as usually is their format), as in (Treur 2017, Sect. 3.3), it is shown how any difference or differential equation can be modeled in the temporal-causal network format.

Note that in the description in this chapter the structure of the base network is reified but not the structure of the reified network as a whole. In a reification process always new structures are added which are themselves not reified. As the next step in Chap. 4 it will be shown how the structure of the reified network also can be reified by socalled second-order reification; see also (Treur 2018b). Then the network reification approach becomes applicable to all examples of second- or higher-order adaptation described in Chap. 1, Sects. 1.2.2 and 1.3. Structures in the first-order reified network used to model adaptation principles are not reified themselves in a first-order reification process. In a second-order reified network, they are reified as well: their structure is explicitly represented by second-order reification states and their connections, which allows modeling adaptive adaptation principles. It is possible for any n to repeat the construction n times and obtain nth-order *reification*. But still, there will be structures introduced in the step from $n - 1$ to n that have no reification. From a theoretical perspective it can also be considered to repeat the construction infinitely many times, for all natural numbers: ω-*order reification*, where ω is the ordinal for the natural numbers. Then an infinite network is obtained, which is theoretically well-defined; all structures in this network are reified within the network itself, but it may not be clear whether it can be applied in practice, or for theoretical questions. This also might be a subject for future research.

References

Banks, D.L., Carley, K.M.: Models for network evolution. J. Math. Sociol. **21**, 173–196 (1996)
Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
Bi, G., Poo, M.: Synaptic modification by correlated activity: Hebb's postulate revisited. Annu. Rev. Neurosci. **24**, 139–166 (2001)
Blankendaal, R., Parinussa, S., Treur, J.: A temporal-causal modelling approach to integrated contagion and network change in social networks. In: Proceedings of the 22nd European Conference on Artificial Intelligence, ECAI'16, pp. 1388–1396. IOS Press (2016)
Bowen, K.A., Kowalski, R.: Amalgamating language and meta-language in logic programming. In: Clark, K., Tarnlund, S. (eds.) Logic Programming, pp. 153–172. Academic Press, New York (1982)
Bowen, K.A.: Meta-level programming and knowledge representation. New Gener. Comput. **3**, 359–383 (1985)
Chandra, N., Barkai, E.: A non-synaptic mechanism of complex learning: Modulation of intrinsic neuronal excitability. Neurobiol. Learn. Mem. **154**(2018), 30–36 (2018)
Demers, F.N., Malenfant, J.: Reflection in logic, functional and objectoriented programming: a short comparative study. In: IJCAI'95Workshop on Reflection and Meta-Level Architecture and their Application in AI, pp. 29–38 (1995)
Galton, A.: Operators vs. arguments: the ins and outs of reification. Synthese **150**, 415–441 (2006)
Gerstner, W., Kistler, W.M.: Mathematical formulations of Hebbian learning. Biol. Cybern. **87**, 404–415 (2002)
Granovetter, M.S.: The strength of weak ties. Amer. J. Sociol. **78**(6), 1360–1380 (1973)
Hebb, D.: The organisation of behavior. Wiley (1949)
Hofstadter, D.R.: Gödel, Escher, Bach. Basic Books, New York (1979)
McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: homophily in social networks. Annu. Rev. Sociol. **27**, 415–444 (2001)
Pearl, J.: Causality. Cambridge University Press (2000)

Rapoport, A.: Spread of Information through a population with socio-structural bias: i. Assumption of transitivity. Bull. Math. Biophys. **15**, 523–533 (1953)

Smorynski, C.: The incompleteness theorems. In: Barwise, J. (ed.) Handbook of Mathematical Logic, vol. 4, pp. 821–865. North-Holland, Amsterdam (1977)

Sousa, N., Almeida, O.F.X.: Disconnection and reconnection: the morphological basis of (mal) adaptation to stress. Trends Neurosci. **35**(12), 742–751 (2012)

Sterling, L., Shapiro, E.: The Art of Prolog. MIT Press, (1986) (Ch 17, pp. 319–356)

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

Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer, Berlin (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.: Network reification as a unified approach to represent network adaptation principles within a network. In: Proceedings of the 7th International Conference on Theory and Practice of Natural Computing, TPNC'18. Lecture Notes in Computer Science, vol 11324, pp. 344–358. Springer, Berlin (2018a)

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, ComplexNetworks'18, vol. 1. Studies in Computational Intelligence, vol. 812, 635–651, Springer, Berlin (2018b)

Treur, J.: The ins and outs of network-oriented modeling: from biological networks and mental networks to social networks and beyond. In: LNCS Transactions on Computational Collective Intelligence. Paper on Keynote lecture at the 10th International Conference on Computational Collective Intelligence, ICCCI'18 vol. 32, pp. 120–139 (2019)

Treur, J., Mohammadi Ziabari, S.S.: An adaptive temporal-causal network model for decision making under acute stress. In: Proceedings of the 10th International Conference on Computational Collective Intelligence, ICCCI'18. Lecture Notes in Computer Science, Springer, Berlin (2018)

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