On adaptive networks and 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_1

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

document license
Article 25fa Dutch Copyright Act

Chapter 1
On Adaptive Networks and Network Reification

Abstract This chapter is a brief preview of what can be expected in this book, with some pointers to various chapters and sections. First, it is discussed how networks can be adaptive in different ways and according to different orders. A variety of examples of first and second-order adaptation are summarized, and the possibility of adaptation of order higher than two is discussed. After this, the notion of network reification is briefly summarized and how it can be used to model adaptive networks in a transparent and network-oriented manner. It is pointed out how repeated application of network reification can be used to model adaptive networks with the adaptation of multiple orders. Finally, it is discussed how mathematical analysis of emerging behavior of a network not only can be applied to non-adaptive base networks, but also to reified adaptive networks.

1.1 Introduction

To model dynamics in real-world processes, different dynamic modeling approaches have been developed based on some type of dynamical system architecture; e.g., Ashby (1960), Port and van Gelder (1995). Within the dynamic modeling area in general, adaptive behaviour is a nontrivial and interesting challenge. For network-oriented dynamic modeling approaches in particular, the considered model structure is a network structure. It turns out that for many real-world domains, network models often show some form of network adaptation by which some of the network structure characteristics change over time. This can be described by *network adaptation principles* specifying how exactly certain characteristics of a network structure change over time. A well-known example of such an adaptation principle concerns adaptation of the connection weights in Mental Networks by *Hebbian learning* (Hebb 1949), which within Cognitive Neuroscience is considered a form of *plasticity* (see Sect. 1.4.2 below). Another example of an adaptation principle is adaptation of the connection weights in Social Networks by *bonding based on homophily*; e.g., Byrne (1986), McPherson et al. (2001), Pearson et al. (2006), Sharpanskykh and Treur (2014).

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_1

The adaptive behavior itself can also be adaptive, which leads to adaptation of different orders; for example, within Cognitive Neuroscience *metaplasticity* determines under which circumstances and to which extent plasticity occurs in Mental Networks; e.g., Abraham and Bear (1996), Magerl et al. (2018), Sehgal et al. (2013), Schmidt et al. (2013), Zelcer et al. (2006). To model adaptive networks in a neat and easily manageable manner is a nontrivial challenge, and even more so when they are adaptive of higher-order.

The notion of *network reification* as introduced in this book is a means to model adaptive networks in a more transparent manner within a Network-Oriented Modelling perspective. Reification literally means representing something abstract as a material or more real concrete thing (Merriam-Webster and Oxford dictionaries). This concept is used in different scientific areas in which it has been shown to provide substantial advantages in expressivity and transparency of models, and, in particular, within AI; e.g., Davis and Buchanan (1977), Davis (1980), Bowen and Kowalski (1982), Demers and Malenfant (1995), Galton (2006), Hofstadter (1979), Sterling and Shapiro (1996), Sterling and Beer (1989), Weyhrauch (1980). Specific cases of reification from a linguistic or logical perspective are representing relations between objects by objects themselves, or representing more complex statements by objects or numbers. For network models, reification can be applied by reifying the network structure characteristics in the form of additional network states (called reification states) within an extended network. Multilevel reified networks can be used to model networks which are adaptive of different orders. It is also discussed how mathematical analysis can be applied to reified networks.

In this chapter, in Sect. 1.2 various examples of first and second-order adaptation are summarized, and in Sect. 1.3 the possibility of adaptation of order higher than 2 is discussed. In Sect. 1.4 the notion of network reification is briefly discussed and how it can be used to model adaptive networks in a more transparent manner; in Sect. 1.5 it is discussed how repeated application of network reification can be used to model adaptive networks with adaptation of multiple orders. In Sect. 1.6 it is discussed how mathematical analysis of emerging behavior of a network not only can be applied to base networks, but also to reification states in reified networks. Finally, Sect. 1.7 is a discussion.

1.2 First- and Second-Order Adaptation

In this section, a brief overview is given of a variety of known adaptation principles of different orders. It shows the wide range of potential applications for a Network-Oriented Modeling approach based on network reification. First, first-order adaptation is addressed in Sect. 1.2.1; next, adaptation of second-order is addressed in Sect. 1.2.2.

1.2.1 First-Order Adaptation

There are many well-known examples of first-order adaptive networks, for example, related to or inspired by adaptation principles from Cognitive Neuroscience, Cognitive Science or Social Science. Just a few examples are listed below. Although the majority of the first-order network adaptation principles known in the literature consider adaptations of connection weights over time, also other characteristics of the network structure can be considered to be adaptive, for example, the way in which incoming impact is aggregated or the speed of processing, as the last two bullets point out:

- Mental or neural networks equipped with a Hebbian learning adaptation principle (Hebb 1949) to adapt connection weights over time ('neurons that fire together, wire together'); see Sects. 1.4 and 1.6 below, Chap. 3, Sect. 3.6.1, and Chap. 4 in this book.
- Mental networks in which an adaptation principle describes how stress affects the connections ('state-connection modulation'); e.g., (Sousa et al. 2012; Treur and Mohammadi Ziabari 2018), see also and Chap. 3, Sect. 3.6.4, and Chap. 5 in this book.
- Mental networks in which an adaptation principle describes how context factors can affect the excitability of states; e.g., (Chandra and Barkai 2018); see also and Chap. 3, Sect. 3.6.5, and Chap. 4 in this book.
- Social networks equipped with an adaptation principle for bonding based on homophily (Byrne 1986; McPherson et al. 2001; Pearson et al. 2006; Sharpanskykh and Treur 2014; Beukel et al. 2019; Blankendaal et al. 2016; Boomgaard et al. 2018) to adapt connection weights over time ('birds of a feather flock together'); see also Chap. 3, Sect. 3.6.1, and Chap. 6 in this book.
- Social networks equipped with a triadic closure adaptation principle to adapt connection weights over time ('friends of my friends will become my friends'); e.g., Rapoport (1953), Banks and Carley (1996), see also Chap. 3, Sect. 3.6.2 in this book.
- Social networks equipped with a preferential attachment adaptation principle expressing that connections are strengthened preferably to nodes that have more and/or stronger connections ('more becomes more'); e.g., Barabasi and Albert (1999); see also Chap. 3, Sect. 3.6.3 in this book.
- Adaptive social network models and analysis of these networks for a variety of application domains can be found in the work around the toolkit for dynamic network analysis and visualization ORA (Carley 2017; Carley et al. 2013b). Among the many applications are (Carley et al. 2013a; Carley and Pfeffer 2012; Merrill et al. 2015).
- Neural networks equipped with (machine) learning mechanisms such as backpropagation or deep learning to adapt connection weights over time; e.g., LeCun et al. (2015).
- Adaptive functions for aggregation of incoming impact and activation of nodes. For example, as mentioned above, their threshold values to model adaptive

intrinsic properties of neurons such as their excitability; e.g., Chandra and Barkai (2018). As another example, the mechanism for the formation of an opinion based on multiple incoming opinions may change over time from selecting the maximal value of them to using the average instead; e.g., Chap. 3, Sects. 3.6.7 and 3.7 in this book. Yet another example describes how for multicriteria decision-making criteria weight factors are changed over time.
- Adaptive speed of states to model adaptive processing speed, for example, the response time of a person depending on workload, or intake of certain chemicals that affect response time; e.g., Chap. 3, Sects. 3.6.6 and 3.7 in this book.

As several real-world examples show, adaptation principles may be adaptive themselves too, according to certain second-order adaptation principles. This will be discussed next.

1.2.2 Second-Order Adaptation

Second-order adaptation can occur in different forms. From recent literature it is apparent that in real world domains characteristics representing adaptation principles often can still change over time, depending on circumstances. The notion of metaplasticity or second-order adaptation has become an important topic within Cognitive Neuroscience and Social Sciences. Some examples are:

- In literature such as Abraham and Bear (1996), Chandra and Barkai (2018), Daimon et al. (2017), Magerl et al. (2018), Parsons (2018), Robinson et al. (2016), Sehgal et al. (2013), Schmidt et al. (2013), Zelcer et al. (2006) various studies are reported which show how adaptation of synapses as described, for example, by first-order adaptation principles based on Hebbian learning can be modulated by a second-order adaptation principle suppressing the first-order adaptation process or amplifying it, thus some form of *metaplasticity* is described. Factors affecting synaptic plasticity as reported are presynaptic or postsynaptic activation, previous (learning) experiences, stress, or intake of certain chemicals or medicine; e.g., (Robinson et al. 2016): 'Adaptation accelerates with increasing stimulus exposure' (p. 2). This is addressed in Chap. 4, Sects. 4.4 and 4.5 in this book.
- From the Social Science area, in an adaptive social network based on a first-order adaptation principle for bonding based on homophily (McPherson et al. 2001) the similarity measure determining how similar two persons are may change over time by a second-order adaptation principle, for example, due to age or other varying circumstances. As an example, for somebody who is very busy or already has a lot of connections the requirements for being similar might become more strict; e.g., see Treur (2018b, 2019b) and Chap. 6 in this book.
- Also in the Social Science area the second-order adaptation concept called 'inhibiting adaptation' can be found Carley (2001, 2002, 2006). The idea is that

networked organisations need to be adaptive in order to survive in a dynamic world. However, some types of circumstances affect this first-order adaptivity in a negative manner, for example, frequent changes of persons or (other) resources. Such circumstances can be considered as inhibiting the adaptation capabilities of the organisation. Especially in Carley (2006) it is described in some detail how such a second-order adaptation principle based on inhibiting the first-order adaptation can be exploited as a strategy to attack organisations that are considered harmful or dangerous such as terrorist networks, by creating circumstances that indeed achieve inhibiting adaptation.

The second item above on adaptive adaptation principles for bonding based on homophily is illustrated in more detail in Chap. 6 in this book; see also (Treur 2018b). For the first item, adaptive adaptation principles for Hebbian learning have been considered in which the adaptation speed (learning rate) and the persistence factor for the first-order Hebbian learning adaptation principle are changing based on a second-order adaptation principle; see Chap. 4, Sects. 4.4 and 4.5 in this book.

In Fessler et al. (2015) some interesting ideas are put forward on first and second-order adaptation for the area of evolutionary adaptive processes.

> For example, the S-curve in the human spine reflects the determinative influence of the original function of the spine as a suspensory beam in a quadrupedal mammal, in contrast to its current function as a load-bearing pillar: whereas the original design functioned efficiently in a horizontal position, the transition to bipedality required the introduction of bends in the spine to position weight over the pelvis (Lovejoy 2005). The resulting configuration makes humans prone to lower-back injury, illustrating how path dependence can both set the stage for kludgy designs and constrain their optimality. Moreover, the combination of bipedality and pressures favoring large brain size in humans exacerbates a conflict between the biomechanics of locomotion (favoring a narrow pelvis) and the need to accommodate a large infant skull during parturition. This increases the importance of higher-order adaptations such as relaxin, a hormone that loosens ligaments during pregnancy, allowing the pelvic bones to separate. (Fessler et al. 2015)

According to this the following types of adaptation can be considered for the human spine:

- First-order adaptation:
  for quadrupedal mammals, a straight horizontal spine is an advantage.
- Second-order adaptation:
  transition to bipedality requires the introduction of bends in the spine to position weight over the pelvis; this makes humans prone to lower-back injury.

Similarly, the following types of adaptation can be considered for the human pelvis:

- First-order adaptation:
  bipedality favors a narrow pelvis.
- Second-order adaptation:
  larger brain size needs a wider pelvis: using relaxin allowing the pelvic bones to separate during giving birth.

1.3 Higher-Order Adaptation

A next question is whether also relevant examples of third- or even higher-order adaptation can be found. First, it will be discussed what orders of adaptation are addressed in the literature. After, a few examples from an evolutionary context are pointed out.

1.3.1 What Orders of Adaptation Are Addressed?

A Google Scholar search in this direction on February 3, 2019 resulted in the following outcomes:

"adaptation"	4 million hits
"second-order adaptation"	360 hits
"third-order adaptation"	14 hits
"fourth-order adaptation"	3 hits.

This shows a very fast decreasing pattern. Further inspection of the left graph in Fig. 1.1 depicting the logarithm of the number of hits against the order clearly shows that the pattern is not just (negatively) exponential. Instead, the right graph in Fig. 1.1 depicting the double logarithm of the number of hits fits much better. The dotted line in that graph is a linear trendline with linear formula in x and y as indicated: the double logarithm seems to allow an almost linear approximation, so the number of hits is in the order of a double (negative) exponential pattern

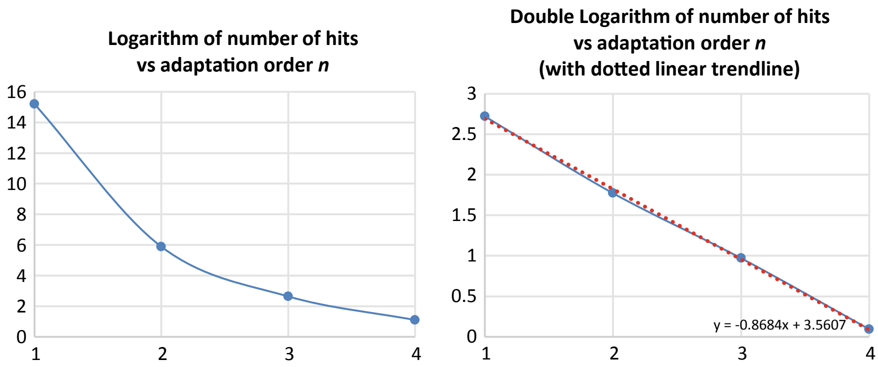

Fig. 1.1 Logarithm (left graph) and double logarithm (right graph) of the number of hits of adaptivity of different orders (vertical axis) in Google Scholar versus the order of adaptation (horizontal axis). The dotted linear trendline shows that the double logarithm of the number of hits has an almost linear dependence of the order of adaptivity

$e^{35.18782e-0.8684n}$ as a function of the order n. This very strongly decreasing pattern may suggest that within a research context, an adaptation of order higher than two is not often considered a very useful or applicable notion.

1.3.2 *Examples of Adaptation of Order Higher Than Two from an Evolutionary Context*

As one of the hits found for third-order adaptation, in (Fessler et al. 2015) the following is put forward referring to second and third-order adaptation:

> Also of relevance here, one form of disgust, pathogen disgust, functions in part as a third-order adaptation, as disease-avoidance responses are up-regulated in a manner that compensates for the increases in vulnerability to pathogens that accompany pregnancy and preparation for implantation – changes that are themselves a second-order adaptation addressing the conflict between maternal immune defenses and the parasitic behavior of the half-foreign conceptus (Fessler et al. 2005; Jones et al. 2005; Fleischman and Fessler 2011). (Fessler et al. 2015)

It could be argued that this domain of evolutionary development is not exactly comparable to the types of application domains considered in the current chapter and book. For example, in evolutionary processes not organisms in their daily life are considered but species on an evolutionary relevant long term time scale. However, at least in a metaphorical sense, this evolutionary domain might provide an interesting source of inspiration.

In Chaps. 7 and 8 the question on adaptive networks of order higher than 2 comes back. Then two application contexts will be addressed for higher-order (higher than 2) adaptive network models: in Chap. 7 one for evolutionary processes as described in the above quote, and in Chap. 8 one for the notion of Strange Loop as put forward by Hofstadter (1979, 2007).

1.4 Using Network Reification to Model Adaptive Networks

In this section, first the often used hybrid approach to adaptive networks is discussed and next the approach based on network reification presented in the current book.

1.4.1 *The Hybrid Approach to Model Adaptive Networks*

Adaptive networks are often modeled in a hybrid manner by considering two types of models that interact with each other (see Fig. 1.2):

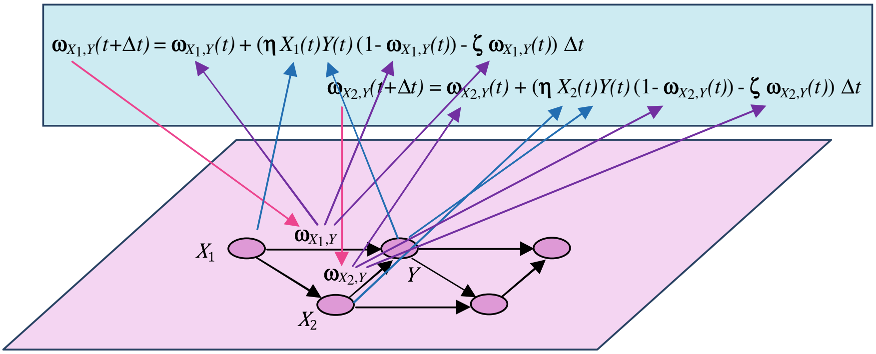

Fig. 1.2 Adaptive network model as a combination of and interaction between a network model and a non-network model

(1) a *network model* for the dynamics within the base network
(2) an *adaptation model* for the dynamics of the network structure characteristics of the base network.

The latter dynamic model is usually specified in a format outside the context of the Network-Oriented Modeling perspective as used for the base network itself. It is most often specified in the form of some procedural or algorithmic programming specification used to run the difference or differential equations underlying the network adaptation process. This non-network dynamic model interacts intensively with the dynamic model for the internal network dynamics of the base network; in Social Network context sometimes this interaction is termed co-evolution; e.g., Holme and Newman (2006), Treur (2019b). An example from the neurocognitive area is modeling *Hebbian learning* (Hebb 1949); this is also what is addressed in Fig. 1.2 in a hybrid manner. Hebbian learning is based on the principle that strengthening of a connection between neurons over time may take place when both states are often active simultaneously: 'neurons that fire together, wire together'. As an illustration, consider the connection from states X_i to state Y. At each point in time t, within the adaptation model (top level rectangle in Fig. 1.2), the change of the weights $\boldsymbol{\omega}_{X_i,Y}(t)$ depends both on the states $X_i(t)$ and $Y(t)$ representing the activation values of the states within the base network (indicated by the blue upward arrows in Fig. 1.2). Note that the blue rectangle only gives a part of the specification needed for the adaptation process in the hybrid situation. Also procedural code is needed to run these equations, which for the sake of simplicity is left out of this picture.

In the picture $\boldsymbol{\eta}$ is the learning rate and $\boldsymbol{\zeta}$ the extinction rate; see also Treur (2016) Chap. 2, p. 93. For this Hebbian learning example, the adaptation principle strengthens a connection when the learning part $\boldsymbol{\eta} X_i(t)Y(t)\;(\mathbf{1}-\boldsymbol{\omega}_{X_i,Y}(t))$ is higher than the extinction part $\boldsymbol{\zeta}\,\boldsymbol{\omega}_{X_i,Y}(t)$ and weakens it when this is opposite. Within the base network, the values for the connection weights as determined by the adaptation model are used all the time (indicated in Fig. 1.2 by the red downward arrows), in order to determine the specific dynamics of each base state. This leaves us with a hybrid model consisting of one network model and one non-network model (see Fig. 1.2), each with their own software components to run them, with interactions between these two different types of models [upward from $\boldsymbol{\omega}_{X_i,Y}(t)$, $X_i(t)$ and $Y(t)$, and downward from $\boldsymbol{\omega}_{X_i,Y}(t+\Delta t)$]. The hybrid approach for adaptive networks was also followed in (Treur 2016) and (Treur and Mohammadi Ziabari 2018). For each new adaptation principle, a new piece of software had to be added. This experience led to the motivation to develop the alternative approach described in this book; for a preview see Sect. 1.4.2.

1.4.2 Modeling Adaptive Networks Based on Network Reification

One class of dynamic modeling approaches is referred to by Network-Oriented Modeling; for this class, some form of network structure is used as basic architecture. In particular, for the Network-Oriented Modeling approach as addressed in Treur (2016, 2019a) the basic architecture chosen is a temporal-causal network architecture defined by three network structure characteristics (for more details, see Chap. 2, or the above references):

(a) **Connectivity of the network**

- *connection weights* $\boldsymbol{\omega}_{X,Y}$ for each connection from a state (or node) X to a state Y.

(b) **Aggregation of multiple connections in the network**

- a *combination function* $\mathbf{c}_Y$**(..)** for each state Y to determining the aggregation of incoming causal impacts.

(c) **Timing in the network**

- a *speed factor* $\boldsymbol{\eta}_Y$ for each state Y.

Such a network architecture can be used to model in a dynamic manner a wide variety of natural processes and human mental and social processes, based on causal relations that are identified in various empirical scientific disciplines, as has been shown in Treur (2016, 2017).

Recently it has been found out how adaptive networks can be modeled differently from the hybrid approach discussed in Sect. 1.4.1, thereby modeling the whole process in a more transparent Network-Oriented Modeling manner by one overall network extending the base network (Treur 2018a); this will be discussed in more detail in Chap. 3. This process of extending the base network by reification states has been called *network reification*. Reification literally means representing something abstract as a material or concrete thing, or making something abstract more concrete or real (Merriam-Webster and Oxford dictionaries). It is used in different scientific areas in which it has been shown to provide substantial advantages in expressivity and transparency of models; e.g., Davis and Buchanan (1977), Davis (1980), Bowen and Kowalski (1982), Demers and Malenfant (1995), Galton (2006), Hofstadter (1979), Sterling and Shapiro (1996), Sterling and Beer (1989), Weyhrauch (1980). For example, strongly enhanced expressive power and more support for modeling adaptivity in a transparent manner are achieved.

Network reification provides similar advantages; it will be shown in Chap. 3 how network reification can be used to explicitly represent all kinds of well-known (e.g., from Cognitive Neuroscience and Social Science) adaptation principles for networks in a more declarative, transparent and unified manner. Examples of such adaptation principles include, among others, principles for Hebbian learning (to model plasticity in the brain), as already mentioned above, and for bonding based on homophily (to model adaptive social networks). Writing procedural or algorithmic specifications and programming code as usually applied for network adaptation in the hybrid approach is not needed anymore. Both the dynamics of the states within the base network and the dynamics of the network structure are run, not by two different interacting software components as in the hybrid case, but by one and the same generic computational reified network engine based on one universal difference equation as described in detail in Chaps. 9 and 10.

Basically, network reification for a temporal-causal network means that for the adaptive network structure characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$ for each state Y of the base network, additional network states $\mathbf{W}_{X,Y}$, $\mathbf{C}_Y$, $\mathbf{H}_Y$ (called *reification states*) are introduced respectively; see the blue upper plane in Fig. 1.3. Here for practical reasons the combination function reification $\mathbf{C}_Y$ actually is a vector ($\mathbf{C}_{1,Y}$, $\mathbf{C}_{2,Y}$, $\mathbf{C}_{3,Y}$,..) of a number of reification states representing weights for a weighted average of basic combination functions from the *combination function library* (see Chap. 9 for more details). Moreover, combination functions usually have some parameters that also can be reified by reification states $\mathbf{P}_{i,j,Y}$, but for the current chapter for the sake of simplicity, these will be left out of consideration. Including reification states for (some of) the characteristics of the base network structure in an extended network is one step. As a next step, the dynamics of the reification states themselves and their impact on base

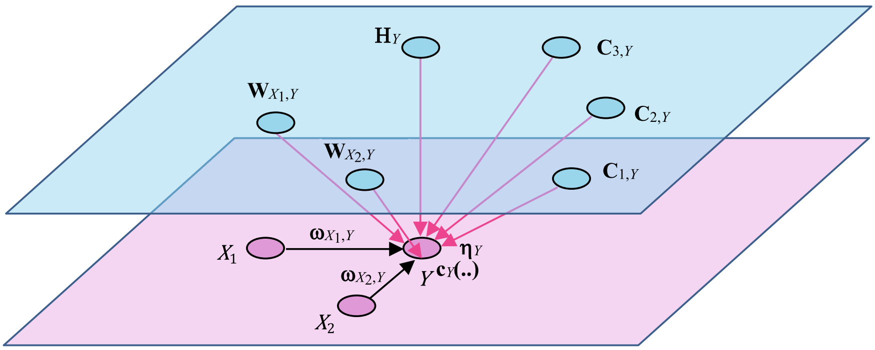

Fig. 1.3 Extending the base network (pink plane) by reification states (blue plane)

state Y are described by additional network structure of the extended network; this additional network structure is also in temporal-causal network format and replaces the blue non-network-like part of the hybrid model and the upward and downward arrows from Fig. 1.2. As a result, an extended network (called a *reified network*) is obtained that embeds the base network in it. This extended network is still a temporal-causal network, as will be pointed out in Chap. 3, Sect. 3.5, and illustrated and proven in more detail in Chap. 10.

Note that the combination function of base state Y has to incorporate these downward causal connections; for details about this, see Chap. 3 or, in more detail, Chaps. 9 and 10. Then a reified network structure is obtained that explicitly represents the characteristics of the (adaptive) base network structure by some of its states, and, moreover, it represents how exactly this base network evolves over time based on adaptation principles that change the base network structure. This construction as described in more detail in Chap. 3 provides an extended temporal-causal network that is called a *reified network architecture*. Like any other state, reification states are defined by three general network structure characteristics connectivity (a), aggregation (b), and timing (c), mentioned above:

(a) For the reification states their *connectivity* in terms of their incoming and outgoing connections has different functions:

- The *outgoing downward causal connections* (the pink downward arrows in Figs. 1.3 or 1.4) from the reification states $\mathbf{W}_{X,Y}$, $\mathbf{C}_Y$, $\mathbf{H}_Y$ to state Y represent the specific causal impact (their special effect) each of these reification states have on Y. These downward causal impacts are standard per type of reification state, and make that the original network characteristics $\boldsymbol{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\boldsymbol{\eta}_Y$ need not be used anymore, as $\mathbf{W}_{X,Y}$, $\mathbf{C}_Y$, $\mathbf{H}_Y$ are used in their place.
- The *upward (or leveled) causal connections* (blue arrows) to the reification states give them the dynamics as desired. They are used to specify, together with the combination function that is chosen and the downward connection, the particular adaptation principle that is addressed.

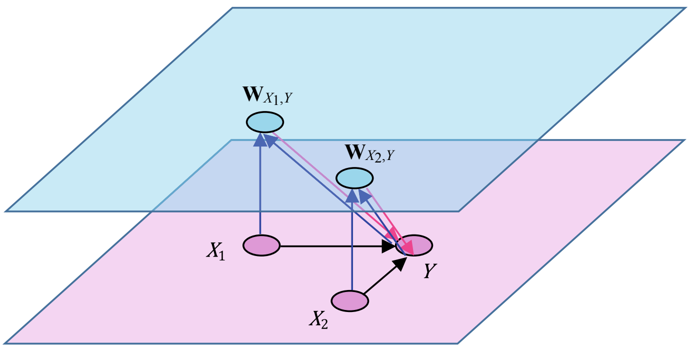

Fig. 1.4 Graphical conceptual representation of a reified network for Hebbian learning in mental networks

This is illustrated in more detail in Fig. 1.4 where network reification for Hebbian learning is modeled. The upward causal connections (blue arrows) to the reification states $\mathbf{W}_{X_1,Y}$ and $\mathbf{W}_{X_2,Y}$ are from the states X_1, X_2, and Y from the base network. Hebbian learning can be described in a simple manner by 'neurons that fire together, wire together'. To incorporate the 'firing together' part, the upward causal connections to $\mathbf{W}_{X_i,Y}$ from the states X_i and Y are needed to express a Hebbian learning adaptation principle. The upward connections are usually assumed to have weight 1. Note that usually also connections from each reification state to itself with weight 1 are applied, but in pictures as in Fig. 1.4 they are often left out.

(b) Concerning the *aggregation* of their incoming causal impacts, in Sect. 1.6 below a combination function that can be used for the reification states $\mathbf{W}_{X_i,Y}$ for Hebbian Learning will be discussed.

(c) Finally, like any other state in a temporal-causal network, reification states have their own *timing* in terms of speed factors. In this case these speed factors represent the *adaptation speed*.

The network reification idea is illustrated for one specific adaptation principle, namely Hebbian learning. In Chap. 3 it is shown how this works for many other well-known adaptation principles.

1.5 Modeling Higher-Order Adaptive Networks by Multilevel Network Reification

What is discussed in Sect. 1.4 is not the end of the story. Reified networks form again a basic temporal-causal network structure defined by certain network structure characteristics. Adaptation principles represented by that reified network structure may themselves be adaptive too, according to certain *second-order adaptation principles*. For example, for real-world processes *plasticity* in Mental Networks as described by Hebbian learning is not a constant feature, but usually varies over time, according to what in Cognitive Neuroscience has been called *metaplasticity* (or *second-order plasticity*); e.g., Abraham and Bear (1996), Magerl et al. (2018), Parsons (2018), Schmidt et al. (2013), Sehgal et al. (2013), Zelcer et al. (2006). To model such multilevel network adaptation processes, it is useful to have some generic architecture in which the different types of adaptation can be modeled in a principled and transparent manner. Such an architecture should be able to distinguish and describe:

(1) *Base network dynamics for base states*
The dynamics within the base network.

(2) *First-order adaptation principles for dynamics of the base network structure*
The dynamics of the base network structure by first-order network adaptation principles.

(3) *Second-order adaptation principles for dynamics of the first-order adaptation principles*
The adaptation of these first-order adaptation principles by second-order adaptation principles.
(4) *Higher-order adaptation principles for dynamics of the second-order adaptation principles*
Maybe still more levels of (higher-order) adaptation.
(5) *Interlevel interactions*
The interactions between these levels.

Such distinctions indeed can be made within a Network-Oriented Modeling framework using the notion of *multilevel reified network architecture*. This type of architecture is obtained by subsequently applying network reification as pointed out in Sect. 1.4 on the reified network structures as well. Repeating multiple times this construction of a reified network architecture provides a multilevel reified network architecture, which will be discussed and illustrated in more detail in Chap. 4.

This multilevel reified (temporal-causal) network architecture is the basis of the implementation of a *dedicated software environment* developed by the author in Matlab, which is discussed in Chap. 9. This environment takes as input so called *role matrices* specifying the network structure characteristics for the different types of states in a designed network model and can just run the model based on them, using a generic *computational reified network engine* included in the environment.

For a (nonadaptive) base network, role matrices are nothing more than a neatly structured way to show in table format (e.g., in Word or in Excel) all values for the characteristics of the model, for example as used in a given simulation scenario; see Chap. 2, Sect. 2.4 and Box 2.1. Each role matrix groups together (for all states), the data of a specific type. In this way, there are five different types of role matrices:

mb for the base connectivity of the network
mcw for the specific connection weights
ms for the speed factors
mcfw for the combination functions used with their weights
mcfp for the parameters of these combination functions.

Again, in case of a nonadaptive network, these five matrices just contain all relevant data (the values) in a standardly structured way as shown in Box 2.1 in Chap. 2. In a reified network, the role matrices are only slightly different, as some of the entries of the role matrices are no values anymore (as now they represent adaptive characteristics), but instead just specify the name of the reification state for this (dynamic) value. This specifies the standard downward causal connection for that reification state: the pink downward arrow in a picture such as Fig. 1.4. The notion of role matrix will be introduced for a base network in detail in Chap. 2 and it will be shown how to apply them in reified networks in Chap. 3.

1.6 Mathematical Analysis of Reified Networks

In this section it will be briefly discussed how the specific structure of reified temporal-causal network models supports mathematical analysis and verification of them.

1.6.1 Mathematical Analysis of a Base Network

As indicated above in Sect. 1.4.2, the choice for temporal-causal networks as basic network architecture comes with three basic concepts for the network structure used for modeling and analysis of the dynamics within a network. It uses state values *X(t)* (usually within the interval [0, 1]) over time for each node *X* and is based on the following *network structure characteristics*:

(a) **Connectivity**

- a *connection weight* $\mathbf{\omega}_{X,Y}$ for each connection from a state *X* to a state *Y*, together with state value *X(t)* defining the single impact $\mathbf{\omega}_{X,Y}X(t)$ of *X* on *Y*.

(b) **Aggregation**

- a *combination function* $\mathbf{c}_Y$**(..)** for each state *Y* to determine the aggregated impact $\mathbf{c}_Y(\mathbf{\omega}_{X_1,Y}X(t), \ldots, \mathbf{\omega}_{X_k,Y}X(t)$ for the incoming single impacts $\mathbf{\omega}_{X_i,Y}X(t)$ of the states $X_1, \ldots, X_k$ with outgoing connections to *Y*

(c) **Timing**

- a *speed factor* $\mathbf{\eta}_Y$ for each state *Y* by which the effect of the aggregated impact on state *Y* is given a proper timing.

These notions are very helpful in designing a dynamic model of a network as they allow the modeler to concentrate on the choices for these values and functions (which are just declarative mathematical objects), and make that he or she will not feel any need to consider procedural specifications or even to do programming. This is enabled by dedicated software environments developed in Matlab that take these basic concepts as input and just run simulations based on them: (Mohammadi Ziabari and Treur 2019) for temporal-causal networks and Treur (2019c) for reified temporal-causal networks, the latter of which is described in Chap. 9. To understand how this can happen, within these software environments based on the network structure characteristics input $\mathbf{\omega}_{X,Y}$, $\mathbf{c}_Y$**(..)**, $\mathbf{\eta}_Y$, for each state *Y* the following difference equation is formed and executed; for a more detailed explanation and motivation, see Chap. 2 or Treur (2016), Chap. 2:

$$Y(t+\Delta t) = Y(t) + \mathbf{\eta}_Y[\mathbf{c}_Y(\mathbf{\omega}_{X_1,Y}X_1(t),\ldots,\mathbf{\omega}_{X_k,Y}X_k(t)) - Y(t)]\Delta t \tag{1.1}$$

As reified temporal-causal networks are themselves also temporal-causal networks, these three concepts and the above difference equation format basically also are applied to the states in a reified network. The only difference is that in case of adaptive network structure characteristics $\mathbf{\omega}_{X_i,Y}$, $\mathbf{\eta}_Y$, $\mathbf{c}_Y(..)$ the values of the reification states for them are incorporated, in the sense that their dynamic values are used in Eq. (1.1) instead of static values.

For mathematical analysis, based on the above equation, a simple criterion in terms of the network structure characteristics $\mathbf{\omega}_{X,Y}$, $\mathbf{c}_Y(..)$, $\mathbf{\eta}_Y$ can be formulated for the network reaching an *equilibrium* which is a situation in which no state changes anymore (Chap. 2, Sect. 2.5):

Criterion for an equilibrium of a temporal-causal network model

For all Y it holds

$$\mathbf{\eta}_Y = 0 \quad \text{or} \quad \mathbf{c}_Y(\mathbf{\omega}_{X_1,Y}X_1(t),\ldots,\mathbf{\omega}_{X_k,Y}X_k(t)) = Y(t) \tag{1.2}$$

The equation is also called an *equilibrium equation*. This criterion can easily be used for analysis of the equilibria of the network model with a central role of the combination functions together with connection weights and speed factors. It will be used for analysis of many of the examples in this book, in order to find out what emerging behaviour can be expected when performing simulations. Such results of mathematical analysis can be used for verification of an implemented network model. If the outcomes of a simulation contradict some of these analysis results, then there is an error that has to be addressed.

Following this line one step further, in Chaps. 11 and 12 a more extensive analysis is presented of possible combination functions together with the network connectivity to model convergence of social contagion to one common value. It is shown that such analysis often results in theorems of the form that if the combination function used and the network connectivity have certain properties (for example, strictly monotonous and scalar-free combination functions, and a strongly connected network), then the network behaviour has certain properties (e.g., when the network ends up in an equilibrium state, all state values are equal).

As reified networks inherit the structure of a temporal-causal network, it turns out that a similar line can be followed for the reification states in a reified network. This is addressed in Chap. 13 (for bonding by homophily) and Chap. 14 (for Hebbian learning). A brief preview (for a Hebbian learning adaptation principle) of this can be found in Sect. 1.6.2 below, after the following preparations.

For each reification state a specific combination function is needed; for the Hebbian learning case considered here, this combination function $\mathbf{c}_{\mathbf{W}_{X_i,Y}}(..)$ for reification state $\mathbf{W}_{X_i,Y}$ is called $\mathbf{hebb}_{\mathbf{\mu}}(..)$ and can simply be specified by a formula of not even one full line; see also Chap. 3, Sect. 3.6.1:

$$\mathbf{hebb}_{\boldsymbol{\mu}}(V_1, V_2, W) = V_1 V_2 (1 - W) + \boldsymbol{\mu} W \quad (1.3)$$

Here V_1, V_2 are variable used for the activation levels $X_i(t)$ and $Y(t)$ of two connected states (with connection weight 1) X_i and Y, and W for the value $\mathbf{W}_{X_i,Y}(t)$ of connection weight reification state $\mathbf{W}_{X_i,Y}$ (also connected to itself with weight 1) $\boldsymbol{\mu}$ is a parameter for the persistence factor. Such a simple declarative one line specification as shown in (1.3) which just specifies the core of the adaptation principle, is in strong contrast with a larger number of lines of procedural programming code as usually needed in a hybrid approach as discussed in Sect. 1.4.1. Note that the extinction parameter $\boldsymbol{\zeta}$ used in Fig. 1.2, and also in (Treur 2016) relates to the persistence parameter $\boldsymbol{\mu}$ as follows:

$$\boldsymbol{\mu} = 1 - \boldsymbol{\zeta}/\boldsymbol{\eta} \quad \text{or} \quad \boldsymbol{\zeta} = (1 - \boldsymbol{\mu})\boldsymbol{\eta} \quad (1.4)$$

In Chap. 15, Sect. 15.2 it is shown that based on this, the two ways of modeling are actually mathematically equivalent. However, the latter way based on (1.1) and (1.3) is more transparent and more uniform with the rest of the model, so that no hybrid form of modeling is needed anymore.

1.6.2 *Mathematical Analysis Applied to Reification States*

The criterion for an equilibrium formulated above can not only be applied to base states, but also to reification states. This enables to find out what emerging behaviour is possible for a reified adaptive network, in the sense of equilibria. For example, as discussed in Sects. 1.4 and 1.6.1 for Hebbian learning (see Fig. 1.4), combination function (1.3) can be used for the reification state $\mathbf{W}_{X,Y}$ for an adaptive connection weight $\boldsymbol{\omega}_{X,Y}$. The connections to the reification state $\mathbf{W}_{X,Y}$ are given weight 1. Assuming nonzero learning speed, the above criterion for an equilibrium applied to the reification state $\mathbf{W}_{X,Y}$ provides the *equilibrium equation* (see also Chap. 3, Sect. 3.6.1, and Chap. 14)

$$V_1 V_2 (1 - W) + \boldsymbol{\mu} W = W \quad (1.5)$$

where W is the value of the reification state $\mathbf{W}_{X,Y}$ and V_1 and V_2 of the connected base states X and Y. In Box 1.1 it is shown how this equation can be rewritten by elementary mathematical rules into an equation of the form $W = \ldots V_1 \ldots V_2 \ldots \boldsymbol{\mu} \ldots$ (an expression in terms of V_1, V_2, and $\boldsymbol{\mu}$).

Box 1.1 Rewriting the equilibrium equation for Hebbian learning for a reified connection weight W

$V_1V_2(1-W) + \mu W = W$	*equilibrium equation*
$V_1V_2 - V_1V_2W + \mu W = W$	*distribution for* V_1V_2
$V_1V_2 = W + V_1V_2W - \mu W$	*rearranging sides*
$V_1V_2 = (1 + V_1V_2 - \mu)W$	*(anti)distribution for* W
$V_1V_2 = (1 - \mu + V_1V_2)W$	*commutation of* $-\mu$ *and* V_1V_2
$W = \frac{V_1V_2}{(1-\mu) + V_1V_2}$	*dividing by* $(1 - \mu + V_1V_2)$

So, the equation

$$W = \frac{V_1V_2}{(1-\mu) + V_1V_2} \tag{1.6}$$

for W is obtained. In case $V_1V_2 = 1$, the outcome is

$$W = \frac{1}{2-\mu} \tag{1.7}$$

This is the maximal value that can be achieved for the connection weight. In case of full persistence $\mu = 1$, the equilibrium Eq. (1.7) results in

$$V_1V_2 = 0 \text{ or } W = 1 \tag{1.8}$$

In that case, the connection weight will always reach the value 1 unless one of the states gets value 0.

This illustrates how the central role of combination functions together with connection weights and speed factors does not only support design and simulation, but also mathematical analysis of a reified network model. In this way, in Chap. 14 a more extensive analysis is made of possible functions (and their properties) to model Hebbian learning, and in Chap. 13 a similar analysis for different functions (and their properties) that can be used for bonding by homophily. It can be seen in these two chapters, similar to what is found in Chaps. 11 and 12, that such an analysis can result in theorems of the form that if the combination function used has certain properties (and the network has a certain type of connectivity), then the network behaviour has certain properties.

1.7 Discussion

In this chapter, it was briefly previewed how networks can be adaptive in different ways and according to different orders. A variety of examples of first- and second-order adaptation were summarized, and pointers were given to chapters in this book where they are addressed in more detail. Also, the possibility of adaptation of order higher than 2 was discussed, which for now can be considered an open question to which Chaps. 7 and 8 will return. The modeling approach presented in this book allows treatment of such adaptation of order higher than two, but real-world examples of it cannot be found easily, as was illustrated by numbers of hits via Google Scholar. The notion of network reification was briefly introduced (as a preview of Chap. 3) and it was shown how it can be used to model adaptive networks in a more transparent and Network-Oriented manner than the usual hybrid approaches. It was discussed how repeated application of network reification can be used to model adaptive networks with adaptation of multiple orders (as a preview of Chap. 4). For this multilevel reified temporal-causal network architecture a dedicated software environment was developed by the author in Matlab. This can be used for modeling and simulation of adaptive networks of any order; it is described in Chap. 9, and more background on the format used in Chap. 10. Finally, it was discussed here how mathematical analysis of emerging behavior of a network not only can be applied to base networks (as addressed in more detail in Chaps. 11 and 12), but also to reified networks (as addressed in more detail in Chaps. 13 and 14).

The basic elements used in Network-Oriented Modeling based on reified temporal-causal networks are declarative: connection weights, combination functions, speed factors are declarative mathematical objects. Together these elements assemble in a standard manner a set of first-order difference or differential equations (1.1), which are declarative temporal specifications. The model's behavior is fully determined by these declarative specifications, given some initial values. The modeling process is strongly supported by using these declarative building blocks. Very complex adaptive patterns can be modeled easily, and in (temporal) declarative form. As mentioned, a dedicated software environment including a generic computational reified network engine (see Chap. 9) takes care of running these high level specifications.

Therefore in many chapters, especially Chaps. 1–9, mathematical and procedural details are kept at a minimum thus obtaining optimal readability for a wide group of readers with diverse multidisciplinary backgrounds. As the Network-Oriented Modeling approach based on reified temporal-causal networks presented in this book abstracts from specific implementation details, making use of the dedicated software environment, modeling can be done without having to design procedural or algorithmic specifications. Moreover, a modeler does not even need to explicitly specify difference or differential equations to get a simulation done, as these are already taken care for by the software environment, based on the modeler's input in the form of the conceptual representation of the network model (the so-called role matrices). Therefore, in Chaps. 1–9 all underlying specific procedural elements and difference or differential equations will usually not be discussed although sometimes the underlying universal difference and differential equation is briefly mentioned; it will be discussed more extensively in Chap. 10. The only mathematical details that will be addressed in Chaps. 1–9 for design of a network model concern the combination functions used, most of which are already given in the available combination function library. For analysis of the emerging behaviour of a network model also these combination functions are central, as the equilibrium equations are based on them, as shown in Sect. 1.6 above and, for example, Eq. (1.5). However, for those readers who still want to see more mathematical details that are covered in the software environment, Chap. 15 presents these in more depth in different sections, as a kind of appendices to many of the chapters.

The causal modeling area has a long history in AI; e.g., Kuipers (1984), Kuipers and Kassirer (1983). Reified temporal-causal network models are part of a newer branch in this causal modeling area. In this branch dynamics is added to causal models, making them temporal as in Treur (2016), but the main contribution in the current book is that a way to specify (multi-order) adaptivity in causal models is added, thereby conceptually using ideas on meta-level architectures that also have a long history in AI; e.g., Davis and Buchanan (1977), Davis (1980), Bowen and Kowalski (1982), Demers and Malenfant (1995), Galton (2006), Hofstadter (1979), Hofstadter (2007), Sterling and Shapiro (1996), Sterling and Beer (1989), Weyhrauch (1980). So, this new Reified Network-Oriented Modeling approach connects two different areas with a long tradition in AI, thereby strongly extending the applicability of causal modeling to dynamic and adaptive notions such as plasticity and metaplasticity of any order, which otherwise would be out of reach of causal modeling.

References

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

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

Banks, D.L., Carley, K.M.: Models for network evolution. J. Math. Soc. **21**, 173–196 (1996)

Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)

Beukel, S.V.D., Goos, S.H., Treur, J.: An adaptive temporal-causal network model for social networks based on the homophily and more- becomes-more principle. Neurocomputing **338**, 361–371 (2019)

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. Frontiers in Artificial Intelligence and Applications, vol. 285, pp. 1388–1396. IOS Press (2016)

Boomgaard, G., Lavitt, F., Treur, J.: Computational analysis of social contagion and homophily based on an adaptive social network model. In: Koltsova, O., Ignatov, D.I., Staab, S. (eds.) Social Informatics: Proceedings of the 10th International Conference on Social Informatics,

SocInfo'18, vol. 1. Lecture Notes in Computer Science vol. 11185, pp. 86–101. Springer Publishers (2018)
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)
Byrne, D.: The attraction hypothesis: do similar attitudes affect anything? J. Pers. Soc. Psychol. **51** (6), 1167–1170 (1986)
Carley, K.M.: Inhibiting adaptation. In: Proceedings of the 2002 Command and Control Research and Technology Symposium, pp. 1–10. Naval Postgraduate School, Monterey, CA (2002)
Carley, K.M.: Destabilization of covert networks. Comput. Math. Organ. Theor. **12**, 51–66 (2006)
Carley, K.M.: ORA: a toolkit for dynamic network analysis and visualization. In: Alhajj, R., Rokne, J. (eds.) Encyclopedia of Social Network Analysis and Mining. Springer (2017). https://doi.org/10.1007/978-1-4614-7163-9_309-1
Carley, K.M., Lee, J.-S., Krackhardt, D.: Destabilizing networks. Connections **24**(3), 31–34 (2001)
Carley, K.M., Pfeffer, J.: Dynamic network analysis (DNA) and ORA. In: Schmorrow, D.D., Nicholson, D.M. (eds.) Advances in Design for Cross-Cultural Activities Part I, pp. 265–274. CRC, Boca Raton (2012)
Carley, K.M., Pfeffer, J., Liu, H., Morstatter, F., Goolsby, R.: Near real time assessment of social media using geo-temporal network analytics. In: Proceedings of 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Niagara Falls, 25–28 Aug 2013 (2013a)
Carley, K.M., Reminga, J., Storrick, J., Pfeffer, J., Columbus, D.: ORA user's guide 2013. Carnegie Mellon University, School of Computer Science, Institute for Software Research. Technical report, CMU-ISR-13–108 (2013b)
Chandra, N., Barkai, E.: A non-synaptic mechanism of complex learning: modulation of intrinsic neuronal excitability. Neurobiol. Learn. Mem. **154**, 30–36 (2018)
Daimon, K., Arnold, S., Suzuki, R., Arita, T.: The emergence of executive functions by the evolution of second–order learning. Artif. Life Robot. **22**, 483–489 (2017)
Davis, R.: Meta-rules: reasoning about control. Artif. Intell. **15**, 179–222 (1980)
Davis, R., Buchanan, B.G.: Meta-level knowledge: overview and applications. In: Proceedings of the 5th International Joint Conference on AI, IJCAI'77, pp. 920–927 (1977)
Demers, F.N., Malenfant, J.: Reflection in logic, functional and object oriented programming: a short comparative study. In: IJCAI'95 Workshop on Reflection and Meta-Level Architecture and their Application in AI, pp. 29–38 (1995)
Fessler, D.M.T., Clark, J.A., Clint, E.K.: Evolutionary psychology and evolutionary anthropology. In: Buss, D.M. (ed.) The Handbook of Evolutionary Psychology, pp. 1029–1046. Wiley, New York (2015)
Fessler, D.M.T., Eng, S.J., Navarrete, C.D.: Elevated disgust sensitivity in the first trimester of pregnancy: evidence supporting the compensatory prophylaxis hypothesis. Evol. Hum. Behav. **26**(4), 344–351 (2005)
Fleischman, D.S., Fessler, D.M.T.: Progesterone's effects on the psychology of disease avoidance: support for the compensatory behavioral prophylaxis hypothesis. Horm. Behav. **59**(2), 271–275 (2011)
Galton, A.: Operators vs. arguments: the ins and outs of reification. Synthese **150**, 415–441 (2006)
Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory (1949)
Hofstadter, D.R.: Gödel, Escher, Bach. Basic Books, New York (1979)
Hofstadter, D.R.: I Am a Strange Loop. Basic Books, New York (2007)
Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Phys. Rev. E **74**(5), 056108 (2006)
Jones, B.C., Perrett, D.I., Little, A.C., Boothroyd, L., Cornwell, R.E., Feinberg, D.R., Tiddeman, B.P., Whiten, S., Pitman, R.M., Hillier, S.G., Burt, D.M., Stirrat, M.R., Law Smith, M.J., Moore, F.R.: Menstrual cycle, pregnancy and oral contraceptive use alter attraction to apparent health in faces. Proc. R. Soc. B **5**(272), 347–354 (2005)

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

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

LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)

Lovejoy, C.O.: The natural history of human gait and posture. Part 2. Hip and thigh. Gait Posture **21**(1), 113–124 (2005)

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

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

Merrill, J.A., Sheehan, B., Carley, K.M., Stetson, P.D.: Transition networks in a cohort of patients with congestive heart failure. a novel application of informatics methods to inform care coordination. Appl. Clin. Inform. **6**(3), 548–564 (2015). https://doi.org/10.4338/aci-2015-02-ra-0021

Mohammadi Ziabari, S.S., Treur, J.: A modeling environment for dynamic and adaptive network models implemented in Matlab. In: Proceedings of the 4th International Congress on Information and Communication Technology (ICICT2019). Springer Publishers (2019)

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

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

Port, R.F., van Gelder, T.: Mind as motion: explorations in the dynamics of cognition. MIT Press, Cambridge, MA (1995)

Rapoport, A.: Spread of Information through a Population with Socio-structural Bias: I. Assumption of transitivity. Bull. Math. Biophys. **15**, 523–533 (1953)

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

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

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

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

Sousa, N., Almeida, O.F.X.: Disconnection and reconnection: the morphological basis of (mal) adaptation to stress. Trends Neurosci. **35**(12), 742–751 (2012). https://doi.org/10.1016/j.tins.2012.08.006. Epub 2012 Sep 21 (2012)

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

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

Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer Publishers (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: Proceeding 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 Publishers (2018a)

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

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

Treur, J.: Mathematical analysis of the emergence of communities based on coevolution of social contagion and bonding by homophily. In: Applied Network Science, vol. 4, p. 39. https://doi-org.vu-nl.idm.oclc.org/10.1007/s41109-019-0130-7 (2019b)

Treur, J.: Design of a Software Architecture for Multilevel Reified Temporal-Causal Networks. https://doi.org/10.13140/rg.2.2.23492.07045. URL: https://www.researchgate.net/publication/333662169 (2019c)

Treur, J., Mohammadi Ziabari, S.S.: An adaptive temporal-causal network model for decision making under acute stress. In: Nguyen, N.T., Trawinski, B., Pimenidis, E., Khan, Z. (eds.) Computational Collective Intelligence: Proceeding of the 10th International Conference, ICCCI 2018, vol. 2. Lecture Notes in Computer Science, vol. 11056, pp. 13–25. Springer Publishers (2018)

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

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