Formal Aspects of Self-* in Autonomic Networked Computing Systems

Phan Cong-Vinh

Abstract A new computing paradigm is currently on spot: autonomic computing (AC), which is inspired by the human autonomic nervous system. AC is characterized by its self-* facets such as self-configuration, self-healing, self-optimization, and self-protection. The overarching goal of AC is to realize computer systems, and thus networked computing systems, that can manage themselves without direct human interventions. Meeting this grand challenge of autonomic computing requires a fundamental approach to the notion of self-*. To this end, taking advantage of the categorical approach we establish, in this chapter, a firm formal basis for modeling self-* in autonomic networked computing systems, developing self-* monoid, category of self-* monoids, and series of self-* facets. All of these are to achieve formal aspects of the self-*.

1 Introduction

Networked computing is a characteristic of many modern computing systems and implies an increased complexity in managing the system behavior. *Autonomic computing* (AC) is essential to keep such systems manageable. In fact, the problem is that many networked computing systems make central or global control impossible. For example, the information needed to make decisions cannot be gathered centrally (e.g., the type of mobile ad hoc networks (MANETs)). In such the networked computing systems, AC is only possible when networked computational entities autonomously interact and coordinate with each other to maintain properly the required computations. Therefore, in the networking environments, we denote AC as *autonomic networked computing* (ANC). In other words, when AC mechanism is

P. Cong-Vinh (✉)
Centre for Applied Formal Methods, London South Bank University, Borough Road, London SE1 0AA, United Kingdom
e-mail: phanvc@ieee.org

M.K. Denko et al. (eds.), *Autonomic Computing and Networking*,
DOI 10.1007/978-0-387-89828-5_16, © Springer Science+Business Media, LLC 2009

implemented in the networked computing systems then it defines ANC paradigm. The essence of ANC is to enable the autonomic networked computational entities to govern themselves the set of services and resources delivered at any given time while interacting and coordinating with each other. Hence, for *ANC systems* (ANCSs), one of major challenges is how to support self-governance in the face of changing user needs, environmental conditions, and computation objectives. In other words, how does an ANCS understand relevant contextual data, change to those data and adapt the services and resources, which it provides, in accordance with goal-driven computational mechanisms?

Dealing with this grand challenge of ANCSs requires a well-founded modeling and in-depth analysis on the notion of ANC. With this aim, we develop a firm formal approach in which autonomic networked computational entities are able to detect, diagnose and repair faults, as well as adapt their configuration and optimize their performance in the face of changing user needs and environmental conditions. All of these must be done while protecting and healing themselves in the face of natural problems and malicious attacks.

AC is often described as self-* , but ANC focuses on self-knowledge in preference to build self-governance. However, self-* functionality is still supported, but the emphasis of ANC is on the foundation to realize self-*, not in the different self-* technologies.

In this view, we see that rigorously approaching to ANC requests fundamental research in all aspects of the self-*. As a novel development for the self-*, we consider to formalize aspects of the self-* taking advantage of categorical language, whose content is presented in this chapter.

2 Outline

The chapter is a reference material for readers who already have a basic understanding of ANCS and are now ready to know the novel approach for formalizing self-* in ANCS using categorical language.

Formalization is presented in a straightforward fashion by discussing in detail the necessary components and briefly touching on the more advanced components. Several exercises and notes explaining how to use the formal aspects, including justifications needed in order to achieve the particular results, are presented.

We attempt to make the presentation as self-contained as possible, although familiarity with the notion of self-* in ANCS is assumed. Acquaintance with the algebra and the associated notion of categorical language is useful for recognizing the results, but is almost everywhere not strictly necessary.

The rest of this chapter is organized as follows: Sections 3, 4 and 5 present the notions of AC, ANC, and some categorical terms, respectively. Section 6 presents models of self-* in ANCSs. In Section 7, structures of self-* including self-* monoid, a category of self-* monoids and some algebraic properties are developed. Section 8 is a place to develop series of self-* facets in detail. In Section 9, we briefly discuss an alternative approach and compare it with our development. Finally, a short summary is given in Section 10.

3 Autonomic Computing as Self-*

AC imitates and simulates the natural intelligence possessed by the human autonomic nervous system using generic computers. This indicates that the nature of software in AC is the simulation and embodiment of human behaviors, and the extension of human capability, reachability, persistency, memory, and information processing speed [52]. AC was first proposed by IBM in 2001 where it is defined as

> "Autonomic computing is an approach to self-managed computing systems with a minimum of human interference. The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement" [22].

AC is generally described as self-*. Formally, let self-* be the set of self-_'s. Each self-_ to be an element in self-* is called a *self-* facet*. That is,

$$\text{self-}^* = \{\text{self-}\_ \mid \text{self-}\_ \text{ is a self-}^* \text{ facet}\} \tag{1}$$

We see that self-CHOP is composed of four self-* facets of self-configuration, self-healing, self-optimization, and self-protection. Hence, self-CHOP is a subset of self-*. That is, self-CHOP = {self-configuration, self-healing, self-optimization, self-protection} $\subset$ self-*. Every self-* facet must satisfy some certain criteria, so-called *self-* properties*. In [55], Wolf and Holvoet classified the self-* properties in autonomic networks.

In its AC manifesto, IBM proposed eight facets setting forth an AC system (ACS) known as *self-awareness, self-configuration, self-optimization, self-maintenance, self-protection (security and integrity), self-adaptation, self-resource-allocation*, and *open-standard-based* [22]. Kinsner pointed out that these facets indicate that IBM perceives AC is a mimicry of human nervous systems [27]. In other words, self-awareness (consciousness) and non-imperative (goal-driven) behaviors are the main features of ACSs [52].

4 Autonomic Networked Computing

From the notion of AC, an ACS is defined by Wang as

> "An autonomic computing system is an intelligent system that implements nondeterministic, context-dependent, and adaptive behaviors based on goal- and inference-driven mechanisms" [53].

This definition is concerned with three major factors of ACS:

- *Variable events*: AC systems do not rely on instructive and procedural information, but are dependent on variable events of ever-changing external environment and internal status formed by the long-term historical events.
- *Variable behaviors*: AC systems behave in a nondeterministic, context-dependent, and adaptive manner.

- *Goal-driven mechanisms*: AC systems do not rely on imperative and procedural instructions, but are dependent on goal-, perception-, and inference-driven mechanisms.

Consequently, ANC, and thus self-*, is achieved when an ACS is constructed as a group of locally interacting autonomous computational entities that cooperate in order to adaptively maintain the desired system-wide behavior without any external or central control. Such an ACS is viewed as an ANCS.

The topic of AC has seen a number of developments through various research investigations following the IBM initiative such as AC paradigm in [9, 18, 37, 41, 47]; different approaches and infrastructures in [1, 5, 44, 46, 55] for enabling autonomic behaviors [48–51]; core enabling systems, technologies, and services in [15, 16, 45, 3, 21, 31] to support the realization of self-* properties in autonomic systems and applications; specific realizations of self-* properties in autonomic systems and applications in [8, 13, 24, 20, 26, 34, 39]; architectures and modeling strategies of autonomic networks in [17, 33, 32]; middleware and service infrastructure as facilitators of autonomic communications in [11, 35, 19]; approaches in [12, 4, 40] to equipping current networks with autonomic functionality for migrating this type of networks to autonomic networks.

Moreover, AC has also been intensely studied by various areas of engineering including artificial intelligence, control systems, and human-orientated systems [25, 36, 53, 54]. Autonomic computing has been set as an important requirement for systems devised to work in new generation global networked and distributed environments such as wireless networks, P2P networks, Web systems, multi-agent systems, and grids [10, 12, 28, 38, 56]. Such systems pose new challenges for the development and application of autonomic computing techniques, due to their special characteristics including *nondeterminism, context-awareness, and goal- and inference-driven adaptability* [53].

5 Some Categorical Terms

In this section, we recall some concepts from the category theory [2, 6, 7, 29, 30] used in this chapter.

5.1 What is a category?

■ A category $\mathbf{C}$ can be viewed as a graph $(Obj(\mathbf{C}), Arc(\mathbf{C}), s, t)$, where

- $Obj(\mathbf{C})$ is the set of nodes we call *objects*,
- $Arc(\mathbf{C})$ is the set of edges we call *morphisms* and
- $s, t : Arc(\mathbf{C}) \longrightarrow Obj(\mathbf{C})$ are two maps called *source* (or *domain*) and *target* (or *codomain*), respectively.

We write $f : \mathscr{X} \longrightarrow \mathscr{Y}$ when f is in $Arc(\mathbf{C})$ and $s(f) = \mathscr{X}$ and $t(f) = \mathscr{Y}$.

Explanation on terminology: An object in the category is an algebraic structure such as a set. We are probably familiar with some notations for finite sets: {*Student A, Student B, Student C*} is a name for the set whose three elements are *Student A, Student B, Student C*. Note that the order in which the elements are listed is irrelevant.

A morphism f in the category consists of three things: a set $\mathscr{X}$, called the source of the morphism; a set $\mathscr{Y}$, called the target of the morphism; and a rule assigning to each element x in the source an element y in the target. This y is denoted by $f(x)$, read "f of x." Note that the morphism is also called the *map*, *function*, *transformation*, *operator*, or *arrow*. For example, let $\mathscr{X}$ = {*Student A, Student B, Student C*}, $\mathscr{Y}$ = {*Math, Physics, Chemistry, History*} and let f assign each student his or her favorite subject. The following internal diagram is an illustration.

$$\begin{tikzcd}
\{Student\ A \arrow[dr] & Student\ B \arrow[d, "f\text{=favorite subject}"] & Student\ C\} \arrow[d] & \\
\{Math & Chemistry & History & Physics\}
\end{tikzcd} \tag{2}$$

This states that the favorite subject of the *Student C* is *History*, written by f(*Student C*) = *History*, while *Student A* and *Student B* prefer *Chemistry*. There are some important properties of any morphism

- From each element in the source {*Student A, Student B, Student C*}, there is exactly one arrow leaving.
- To an element in the target {*Math, Physics, Chemistry, History*}, there may be zero, one or more arrows arriving.

It is possible that the source and target of the morphism could be the same set. The following internal diagram is an example.

$$\begin{tikzcd}
\{Student\ A \arrow[dr] & Student\ B \arrow[dr, "e\text{=favorite classmate}"] & Student\ C\} \arrow[dl] \\
\{Student\ A & Student\ B & Student\ C\}
\end{tikzcd} \tag{3}$$

and, in the case, the morphism is called an *endomorphism* whose representation is available as in

$$\begin{tikzcd}
\{Student\ A \arrow[r] & Student\ B \arrow[r, bend right] & Student\ C\} \arrow[l]
\end{tikzcd} \tag{4}$$

■ Associated with each object $\mathscr{X}$ in $Obj(\mathbf{C})$, there is a morphism $1_{\mathscr{X}} = \mathscr{X} \longrightarrow \mathscr{X}$, called the *identity* morphism on $\mathscr{X}$, and to each pair of morphisms $f : \mathscr{X} \longrightarrow \mathscr{Y}$ and $g : \mathscr{Y} \longrightarrow \mathscr{Z}$, there is an associated morphism $f; g : \mathscr{X} \longrightarrow \mathscr{Z}$, called the *composition* of f with g. The representations in (5) include the external diagrams of identity morphism and composition of morphisms.

$$\begin{tikzcd} \mathscr{X} \arrow[loop above, "1_{\mathscr{X}}"] & & \mathscr{X} \arrow[r, "f"] \arrow[rr, bend right, "f;g"'] & \mathscr{Y} \arrow[r, "g"] & \mathscr{Z} \end{tikzcd} \tag{5}$$

Explanation on terminology: Here are the corresponding internal diagrams of the identity morphism.

$$\begin{tikzcd} \{Student\ A \arrow[d] & Student\ B \arrow[d, "1_{\mathscr{X}}"] & Student\ C\} \arrow[d] \\ \{Student\ A & Student\ B & Student\ C\} \end{tikzcd} \tag{6}$$

Or

$$\begin{tikzcd} \{Student\ A \arrow[loop above] & Student\ B \arrow[loop below] & Student\ C\} \arrow[loop above] \end{tikzcd} \tag{7}$$

And here, the composition of morphisms is described in the internal diagram

$$\begin{tikzcd} \{Student\ A \arrow[dr] & Student\ B \arrow[dr, "e\text{=favorite classmate}"] & Student\ C\} \arrow[dl] & \\ \{Student\ A \arrow[dr] & Student\ B \arrow[d, "f\text{=favorite subject}"] & Student\ C\} \arrow[d] & \\ \{Math & Chemistry & History & Physics\} \end{tikzcd} \tag{8}$$

Or, in the external diagram $\mathscr{X} \xrightarrow{e} \mathscr{X} \xrightarrow{f} \mathscr{Y}$. By diagram (8), we can obtain answers for the question "What should each student support to his or her favorite classmate for subject?" In fact, the answers are such as " *Student A* likes *Student B*, *Student B* likes *Chemistry*, so *Student A* should support *Chemistry*," "*Student B* likes *Student C*, *Student C* likes *History*, so *Student B* should support *History*" and "*Student C* likes *Student B*, *Student B* likes *Chemistry*, so *Student C* should support *Chemistry*."

The composition of two morphisms e and f means that e and f are combined to obtain a third morphism $\mathscr{X} \xrightarrow{e;f} \mathscr{Y}$. This is represented in the following internal diagram.

$$\begin{tikzcd} \{Student\ A \arrow[dr] & Student\ B \arrow[dr, "e;f"] & Student\ C\} \arrow[dl] & \\ \{Math & Chemistry & History & Physics\} \end{tikzcd} \tag{9}$$

where, for example, $e; f(\mathit{Student\ B}) = \mathit{History}$ is read as “the favorite subject of the favorite classmate of *Student B* is *History*.”
■ The following equation must hold for all objects $\mathscr{X}$, $\mathscr{Y}$ in $Obj(\mathbf{C})$ and morphism $f : \mathscr{X} \longrightarrow \mathscr{Y}$ in $Arc(\mathbf{C})$:

$$\mathit{Identity}: \qquad 1_{\mathscr{X}}; f = f = f; 1_{\mathscr{Y}} \tag{10}$$

$$\begin{tikzcd}
\mathscr{X} \arrow[loop left, "1_{\mathscr{X}}"] \arrow[r, "f"] & \mathscr{Y} & = & \mathscr{X} \arrow[r, "f"] & \mathscr{Y} & = & \mathscr{X} \arrow[r, "f"] & \mathscr{Y} \arrow[loop right, "1_{\mathscr{Y}}"]
\end{tikzcd}$$

The following equation must hold for all objects $\mathscr{X}$, $\mathscr{Y}$ and $\mathscr{Z}$ in $Obj(\mathbf{C})$ and morphisms $f : \mathscr{X} \longrightarrow \mathscr{Y}$, $g : \mathscr{Y} \longrightarrow \mathscr{Z}$ and $h : \mathscr{Z} \longrightarrow \mathscr{T}$ in $Arc(\mathbf{C})$:

$$\mathit{Associativity}: \qquad (f; g); h = f; (g; h) \tag{11}$$

$$\begin{tikzcd}
\mathscr{X} \arrow[r, "f"] \arrow[rr, bend right, "f;g"'] & \mathscr{Y} \arrow[r, "g"] & \mathscr{Z} \arrow[r, "h"] & \mathscr{T} & = & \mathscr{X} \arrow[r, "f"] & \mathscr{Y} \arrow[r, "g"] \arrow[rr, bend right, "g;h"'] & \mathscr{Z} \arrow[r, "h"] & \mathscr{T}
\end{tikzcd}$$

5.2 Isomorphism

A morphism $f : \mathscr{X} \longrightarrow \mathscr{Y}$ in the category $\mathbf{C}$ is an *isomorphism* if there exists a morphism $g : \mathscr{Y} \longrightarrow \mathscr{X}$ in that category such that $f; g = 1_{\mathscr{X}}$ and $g; f = 1_{\mathscr{Y}}$.

$$\begin{tikzcd}
\mathscr{X} \arrow[r, "f"] \arrow[rr, bend right, "f;g=1_{\mathscr{X}}"'] & \mathscr{Y} \arrow[r, "g"] & \mathscr{X} & \text{and} & \mathscr{Y} \arrow[r, "g"] \arrow[rr, bend right, "g;f=1_{\mathscr{Y}}"'] & \mathscr{X} \arrow[r, "f"] & \mathscr{Y}
\end{tikzcd} \tag{12}$$

That is, if the following diagram commutes.

$$\begin{tikzcd}
\mathscr{X} \arrow[loop left, "1_{\mathscr{X}}"] \arrow[r, bend right, "f"'] & \mathscr{Y} \arrow[loop right, "1_{\mathscr{Y}}"] \arrow[l, bend right, "g"']
\end{tikzcd} \tag{13}$$

5.3 Element of a set

For any set A, $x \in A$ iff $1 \xrightarrow{x} A$ (or $x : 1 \longrightarrow A$) where 1 denotes a singleton set. Focus on one element of {*Math, Physics, Chemistry, History*}, say {*subject*}, and call this set “1.” Let us see what the morphisms from 1 to {*Math, Physics, Chemistry, History*} are. There are exactly four of them.

$$\begin{tikzcd} \{\mathit{subject}\} \arrow[r, bend right, "\mathit{Math}"'] & \{\mathit{Math} & \mathit{Chemistry} & \mathit{History} & \mathit{Physics}\} \end{tikzcd} \tag{14}$$

$$\begin{tikzcd} \{\mathit{subject}\} \arrow[rr, bend right, "\mathit{Chemistry}"'] & \{\mathit{Math} & \mathit{Chemistry} & \mathit{History} & \mathit{Physics}\} \end{tikzcd} \tag{15}$$

$$\begin{tikzcd} \{\mathit{subject}\} \arrow[rrr, bend right, "\mathit{History}"'] & \{\mathit{Math} & \mathit{Chemistry} & \mathit{History} & \mathit{Physics}\} \end{tikzcd} \tag{16}$$

$$\begin{tikzcd} \{\mathit{subject}\} \arrow[rrrr, bend right, "\mathit{Physics}"'] & \{\mathit{Math} & \mathit{Chemistry} & \mathit{History} & \mathit{Physics}\} \end{tikzcd} \tag{17}$$

By this way, we can write $1 \xrightarrow{2} \mathbb{N}$ (or $2 : 1 \longrightarrow \mathbb{N}$) for $2 \in \mathbb{N}$, $1 \xrightarrow{i} \mathbb{N}$ (or $i : 1 \longrightarrow \mathbb{N}$) for $i \in \mathbb{N}$ and so on.

5.4 Functor

Functor is a special type of mapping between categories. Functor from a category to itself is called an *endofunctor*. Note that the functors are also viewed as morphisms in a category, whose objects are smaller categories.

5.5 T-algebra

Let $\mathbf{C}$ be a category, $\mathscr{A}$ an object in $Obj(\mathbf{C})$, $\mathsf{T} : \mathbf{C} \longrightarrow \mathbf{C}$ an endofunctor and f a morphism $\mathsf{T}(\mathscr{A}) \xrightarrow{f} \mathscr{A}$; *then* T-*algebra* is a pair $\langle \mathscr{A}, f \rangle$. $Obj(\mathbf{C})$ is called a *carrier* of the algebra and T a *signature* of the algebra.

6 Self-* in ANCSs

As known that ANC, and thus self-*, is achieved when ANCSs are constructed. In this way, for forming ANCSs, we start with considering *deterministic autonomic networked computing systems* (DANCSs) and then extend to *nondeterministic autonomic networked computing systems* (NANCSs) by categorical approach in this section.

6.1 Self- in* DANCSs

DANC we want to abstract is intuitionally multiple partial morphism applications, such as

$$s_0 \xrightarrow{\sigma_0} s_1 \xrightarrow{\sigma_1} s_2 \xrightarrow{\sigma_2} s_3 \cdots \tag{18}$$

where

- All indexes $i \in T$ $(= \mathbb{N} \cup \{0\})$ refer to times,
- s is a state of DANCS in the set, denoted by *Sys*, of states. s_i is the state s at the time i,
- σ is a contextual data in the set, denoted by *Context*, of contextual data. σ_i is the contextual data σ at the time i, which makes change of the state s_i to become s_{i+1}.

The meaning of (18) is understood as

$$\ldots\ s_2(s_1(s_0())) = \ldots\ s_2(s_1(\sigma_0)) = \ldots\ s_2(\sigma_1) = \ldots\ \sigma_2 \tag{19}$$

The adaptation process in (18) can also be descriptively drawn as

$$s_0()\ \sigma_0\ \sigma_1\ \sigma_2\ \cdots \longmapsto s_1(\sigma_0)\ \sigma_1\ \sigma_2\ \cdots \longmapsto \sigma_0\ s_2(\sigma_1)\ \sigma_2\ \cdots \tag{20}$$

or, in another representation

$$\xrightarrow{s_0} \sigma_0\ \sigma_1\ \sigma_2\ \cdots \longmapsto \sigma_0 \xrightarrow{s_1} \sigma_1\ \sigma_2\ \cdots \longmapsto \sigma_0\ \sigma_1 \xrightarrow{s_2} \sigma_2\ \cdots \tag{21}$$

Note that in (20) and (21), we want to represent the above-mentioned adaptation process of DANCS based on context where each step of the process is an application of unary partial morphism $1 \xrightarrow{s_i} Sys$ on $1 \xrightarrow{\sigma_{i-1}} Context$, for all i in T.

The adaptation process, in (20) and (21), describes the notion of DANC in DANCSs including the adaptation steps to change *configurations* of the system.

Definition 1 (Configuration of DANCS) *We define a configuration of* DANCS *at an adaptation step to be a member of the set Sys* $\times$ *Context*$^{i \in T}$*, where Context*$^{i \in T}$ *stands for*

$$Context^{i \in T} = \underbrace{Context \times Context \times \ldots \times Context}_{i\ times} \tag{22}$$

Explanation on terminology: As we know, when we combine sets by multiplication, each set is a *factor* and the resulting set is the *product*. Hence, each set *Context* is a factor of the resulting set *Context*$^{i \in T}$, *Sys* and *Context*$^{i \in T}$ are two factors of the set $Sys \times Context^{i \in T}$. The definition of multiplication of sets is very natural. Just remember that a product is not just a set, but a set with two morphisms as in

- When $i = 2$ then $Context^2 = \{< \sigma_1, \sigma_2 > | \sigma_1, \sigma_2 \in Context\}$ is obtained by

$$\begin{tikzcd}
& \sigma_1 \in Context \\
< \sigma_1, \sigma_2 > \in Context^2 \arrow[ur, "f_1"] \arrow[dr, "f_2"'] & \\
& \sigma_2 \in Context
\end{tikzcd}$$

- When $i = 3$ then $Context^3 = \{<< \sigma_1, \sigma_2 >, \sigma_3 > | \sigma_1, \sigma_2, \sigma_3 \in Context\}$ is obtained by

$$\begin{tikzcd}
& < \sigma_1, \sigma_2 > \in Context^2 \\
<< \sigma_1, \sigma_2 >, \sigma_3 > \in Context^3 \arrow[ur, "g_1"] \arrow[dr, "g_2"'] & \\
& \sigma_3 \in Context
\end{tikzcd}$$

Specially, we have

- If $i = 0$ then $Context^0 = \{\}$
- If $i = 1$ then $Context^1 = Context = \{\sigma_1 | \sigma_1 \in Context\}$

We hope that these diagrams seem suggestive to readers. Our aim is to learn to use them as precise tools of understanding and reasoning, not merely as intuitive guides.

The DANC paradigm, which we want to approach to, is based on mapping a configuration to another. Let us see the following examples

Example 1 A specific DANC can be specified by the following morphism:

$$Self\text{-}X : (Sys \times Context) \longrightarrow Sys \tag{23}$$

(i.e., $Self\text{-}X : (Sys \times Context^1) \longrightarrow (Sys \times Context^0)$ or denoted by $Self\text{-}X$ $(Sys \times Context, Sys)$)

Example 2 Another specific DANC can be specified by

$$Self\text{-}X : (Sys \times Context) \longrightarrow (Sys \times Context) \tag{24}$$

(i.e., $Self\text{-}X : (Sys \times Context^1) \longrightarrow (Sys \times Context^1)$ or denoted by $Self\text{-}X$ $(Sys \times Context, Sys \times Context)$)

Example 3 Again, we can also specify another specific DANC as

$$Self\text{-}X : (Sys \times Context^n) \longrightarrow (Sys \times Context) \tag{25}$$

(i.e., $Self\text{-}X : (Sys \times Context^n) \longrightarrow (Sys \times Context^1)$ or denoted by $Self\text{-}X$ $(Sys \times Context^n, Sys \times Context))$
and we can, in the completely same way, do for any other specific DANC.

Definition 2 *Generally, an arbitrary* DANC *is specified by*

$$Self\text{-}X : (Sys \times Context^{i \in T}) \longrightarrow (Sys \times Context^{j \in T}) \tag{26}$$

Now, let us try to do the following exercise.

Exercise 1 (Self-* in DANCSs) Show that the morphism *Self-X* in (26) *defines self-* in* DANCSs

Solution *This stems from* (26) *and the fact that ANC, and thus DANC, is described through self-*.* □

Morphism *Self-X* is called a self-*. Morphism *Self-X* in (26) defines a set $\{Self\text{-}X_{k \in \mathbb{N}}\}$ of mappings such that

$$\{Self\text{-}X_{k \in \mathbb{N}}\} : (Sys \times Context^{i \in T}) \longrightarrow (Sys \times Context^{j \in T}) \tag{27}$$

Hence, let us do the exercise as in

Exercise 2 (Self-* facets in DANCSs) Show that the set $\{Self\text{-}X_{k \in \mathbb{N}}\}$ in (27) defines self-* facets in DANCSs. Each mapping $Self\text{-}X_{k \in \mathbb{N}}$ is called a self-* facet.

Solution This originates as the result of the truth that self-* is the set of self-* facets. □

For further well-founded investigation, we can construct a category of the sets of DANCS configurations and establish *Self-X*-algebras as described in the following exercises.

Exercise 3 (Category of the sets of DANCS configurations) Show that the sets of DANCS configurations as in Definition 1 define a category.

Solution In fact, let **Cat(DANCS)** be such a category of the sets of DANCS configurations, whose structure is constructed as follows:

- Each set of configurations $Sys \times Context^{i \in T}$ defines an object.
  That is, *Obj*(**Cat(DANCS)**) $= \{Sys \times Context^{i \in T}\}$.
- Each $Self\text{-}X$ defines a morphism.
  That is, *Arc*(**Cat(DANCS)**) $= \{Self\text{-}X : (Sys \times Context^{i \in T}) \longrightarrow (Sys \times Context^{j \in T})\}$.

It is easy to check that identity in (10) and associativity in (11) on all $Self$-Xs are satisfied. □

Exercise 4 (*Self-X*-algebra(DANCS)) Show that each morphism *Self-X* in the category **Cat(DANCS)** defines an algebra, so-called *Self-X*-algebra (DANCS).

Solution This stems from definition of T-algebra in Section 5, where functor T is defined such that $\mathsf{T} = \biguplus\{Self\text{-}X\}$. Note that the notation $\biguplus$ stands for *disjoint union* or *coproduct*. □

With the result of Exercise 4, we obtain a compact formal definition of DANCS as in

Definition 3 (DANCS) *Each Self-X-algebra*(DANCS) *defines a* DANCS

Both *Sys* and *Context* may be infinite. If both *Sys* and *Context* are finite, then we have a finite DANCS, otherwise we have an infinite DANCS.

6.2 Self- in* NANCSs

In NANC we want to model is intuitionally multiple partial morphism applications, such as

$$s_0 \xrightarrow{\sigma_0|x_0} s_1 \xrightarrow{\sigma_1|x_1} s_2 \xrightarrow{\sigma_2|x_2} s_3 \cdots \tag{28}$$

where

- All indexes i in T, s_i, and σ_i are similar in meaning to the ones mentioned in (18)
- x_i is a real number that can be thought of as the *multiplicity* (or *weight*) with which the adaptation from s_i to s_{i+1} occurs.

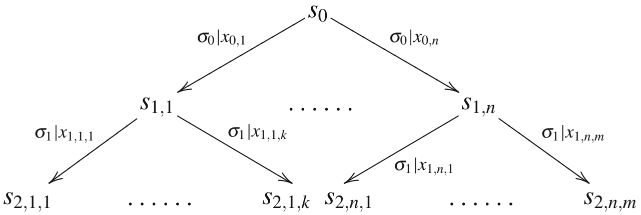
Adaptation process of NANC in diagram (28) can be separated into two complementary parts as follows:

$$s_0 \xrightarrow{\sigma_0} s_1 \xrightarrow{\sigma_1} s_2 \xrightarrow{\sigma_2} s_3 \ldots \tag{29}$$

and

$$s_0 \xrightarrow{x_0} s_1 \xrightarrow{x_1} s_2 \xrightarrow{x_2} s_3 \ldots \tag{30}$$

On the one hand, diagram (29) emphasizes $1 \xrightarrow{\sigma_i} Context$, for all i in T, in the adaptation process. This allows us to discover conveniently sequence of σ_i as series of contextual data. On the other hand, diagram (30) gives rise to $1 \xrightarrow{x_i} \mathbb{R}$, for all i in T, as weights of the series of contextual data in the adaptation process to support an evaluation of weight-based quantitative behaviors of the series of contextual data. Some first steps of the adaptation process in (28) can also be descriptively drawn as



(31)

Diagram (31) is thought of as

- For the first step,

  $s_1 \in \{s_{1,1},\ldots,s_{1,n}\} \subset Sys$
  and
  $x_0 \in \{x_{0,1},\ldots,x_{0,n}\} \subset \mathbb{R}$
- For the second step,
  $s_2 \in \{s_{2,1,1},\ldots,s_{2,1,k}\} \cup \ldots \cup \{s_{2,n,1},\ldots,s_{2,n,m}\} \subset Sys$
  and
  $x_1 \in \{x_{1,1,1},\ldots,x_{1,1,k}\} \cup \ldots \cup \{x_{1,n,1},\ldots,x_{1,n,m}\} \subset \mathbb{R}$

and the meaning of (28) is viewed as the following morphism.

$$Self\text{-}X : (Sys \times Context) \longrightarrow (Sys \longrightarrow \mathbb{R}) \tag{32}$$

Explanation on terminology: The adaptation morphism *Self-X* in (32) is nondeterministic and this can be explained as follows: *Self-X* assigns to each configuration in $Sys \times Context$ a morphism $Sys \longrightarrow \mathbb{R}$ that can be seen as a kind of *nondeterministic configuration* (or so-called *distributed configuration*) and specifies for every state s' in Sys a multiplicity (or weight) $Self\text{-}X(< s,\sigma >)(s')$ in $\mathbb{R}$.

This nondeterminism of NANC makes extension in representation of the categorical models mentioned in Section 6.1. Let us see the following examples

Example 4 A specific NANC, which is specified by the following morphism, is an extension of (23):

$$Self\text{-}X : (Sys \times Context) \longrightarrow (Sys \longrightarrow \mathbb{R}) \tag{33}$$

(i.e., $Self\text{-}X : (Sys \times Context^1) \longrightarrow ((Sys \times Context^0) \longrightarrow \mathbb{R})$ or denoted by $Self\text{-}X\ ((Sys \times Context), (Sys \longrightarrow \mathbb{R})))$

Example 5 The model in (24) extended for NANC is specified by

$$Self\text{-}X : (Sys \times Context) \longrightarrow ((Sys \times Context) \longrightarrow \mathbb{R}) \tag{34}$$

(i.e., $Self\text{-}X : (Sys \times Context^1) \longrightarrow ((Sys \times Context^1) \longrightarrow \mathbb{R})$ or denoted by $Self\text{-}X\ ((Sys \times Context), ((Sys \times Context) \longrightarrow \mathbb{R})))$

Example 6 Again, we specify another specific NANC as an extension of (25) in

$$Self\text{-}X : (Sys \times Context^{n}) \longrightarrow ((Sys \times Context) \longrightarrow \mathbb{R}) \qquad (35)$$

(i.e., $Self\text{-}X : (Sys \times Context^{n}) \longrightarrow ((Sys \times Context^{1}) \longrightarrow \mathbb{R})$ or denoted by $Self\text{-}X\ ((Sys \times Context^{n}), ((Sys \times Context) \longrightarrow \mathbb{R})))$

and, in the completely same way, we do for an arbitrary NANC as in

Definition 4 *Generally, an arbitrary* NANC *is specified by*

$$Self\text{-}X : (Sys \times Context^{i \in T}) \longrightarrow ((Sys \times Context^{j \in T}) \longrightarrow \mathbb{R}) \qquad (36)$$

Let us do the exercise as described in

Exercise 5 (Self-* in NANCSs) Show that the morphism *Self-X* in (36) defines self-* in NANCSs

Solution This stems from (36) and the fact that ANC, and thus NANC, is described through self-*. □

Morphism *Self-X* in (36) defines a set $\{Self\text{-}X_{k \in \mathbb{N}}\}$ of mappings such that

$$\{Self\text{-}X_{k \in \mathbb{N}}\} : (Sys \times Context^{i \in T}) \longrightarrow ((Sys \times Context^{j \in T}) \longrightarrow \mathbb{R}) \qquad (37)$$

Thus, let us do the following exercises

Exercise 6 (Self-* facets in NANCSs) Show that the set $\{Self\text{-}X_{k \in \mathbb{N}}\}$ in (37) defines self-* facets in NANCSs. Each mapping $Self\text{-}X_{k \in \mathbb{N}}$ is called a self-* facet.

Solution This originates as the result of the truth that self-* is the set of self-* facets. □

Exercise 7 (Category of the sets of NANCS configurations) Show that the category **Cat(DANCS)** equipped with structure $(Sys \times Context^{i \in T}) \longrightarrow ((Sys \times Context^{j \in T}) \longrightarrow \mathbb{R})$ defines a category **Cat(NANCS)** of the sets of NANCS configurations.

Solution This result comes immediately from Exercise 3. □

Exercise 8 (*Self-X*-algebra(NANCS)) Show that the structure $(Sys \times Context^{i \in T}) \longrightarrow ((Sys \times Context^{j \in T}) \longrightarrow \mathbb{R})$ in the category **Cat(NANCS)** defines an algebra, so-called *Self-X*-algebra (NANCS).

Solution This originates from definition on T-algebra in Section 5, where functor T is defined such that $\mathsf{T} = \biguplus\{Self\text{-}X\}$ (similar to Exercise 4) with *Self-X* defined in (36). □

With this result of Exercise 8, we obtain a compact formal definition of NANCS as in

Definition 5 (NANCS) *Each Self-X-algebra* (NANCS) *defines a* NANCS

Moreover, let us do the following exercise to obtain a significant relationship between DANCSs and NANCSs.

Exercise 9 (Relationship between DANCSs and NANCSs) Show that DANCSs are just of specific NANCSs. In other words, using categorical language, DANCSs $\xrightarrow{\subset}$ NANCSs

Solution In fact, by the adaptation morphism in (36) of NANCSs, let f be the morphism $f : (Sys \times Context^{j \in T}) \longrightarrow \mathbb{R}$, $Conf$ be $Sys \times Context^{j \in T}$ and the finite set $\mathbb{R}(Conf) = \{1 \xrightarrow{c} Conf | f(c) \neq 0\} \xrightarrow{\subseteq} Conf$. Hence it follows that when $\exists!\ 1 \xrightarrow{c} Conf : f(c) = 1$ but $\forall c' \neq c : f(c') = 0$ (i.e., the set $\mathbb{R}(Conf)$ is a singleton set of configuration with weight of 1. Note that the notation $\exists!$ is read as "exist only") then (36) becomes the adaptation morphism of DANCSs as in (26). In other words, in the case, NANCSs will become DANCSs. □

7 Structures of Self-*

In this section, we construct self-* monoid and then a category of self-* monoids in order to consider the significant properties of the self-*.

7.1 *Self-* Monoid*

We know that self-* is specified by the morphism $Self\text{-}X : (Sys \times Context^{n \in T}) \longrightarrow (Sys \times Context^{n \in T})$, which defines the set $\{Self\text{-}X_{i \in \mathbb{N}}(Sys \times Context^{n \in T}, Sys \times Context^{n \in T})\}$ of self-* facets. Let $\mathbf{Self\text{-}X}^{n \in T}$ be the set of such self-* facets, then

$$\mathbf{Self\text{-}X}^{n \in T} = \{Self\text{-}X_{i \in \mathbb{N}}(Sys \times Context^{n \in T}, Sys \times Context^{n \in T})\} \tag{38}$$

Note that, in the case, we write $Self\text{-}X^{n \in T}_{i \in \mathbb{N}}$ to stand for $Self\text{-}X_{i \in \mathbb{N}}(Sys \times Context^{n \in T}, Sys \times Context^{n \in T})$. Thus, we have

$$\mathbf{Self\text{-}X}^{n \in T} = \{Self\text{-}X^{n \in T}_{i \in \mathbb{N}}\} \tag{39}$$

This set with the composition operation ";" satisfies two following properties.

7.1.1 Composition of Self-* Facets

Let f and g be members of $\mathbf{Self\text{-}X}^{n \in T}$, then the composition of self-* facets $f;g : (Sys \times Context^{n \in T}) \longrightarrow (Sys \times Context^{n \in T})$ is as $g : (f : (Sys \times Context^{n \in T})$

$\longrightarrow (Sys \times Context^{n\in T})) \longrightarrow (Sys \times Context^{n\in T})$. In other words, let $f = Self\text{-}X^{n\in T}_{i\in\mathbb{N}}$ and $g = Self\text{-}X^{n\in T}_{j\in\mathbb{N}}$ then

$$(Self\text{-}X^{n\in T}_{i\in\mathbb{N}}\ ;\ Self\text{-}X^{n\in T}_{j\in\mathbb{N}}) = Self\text{-}X_{j\in\mathbb{N}}(Self\text{-}X^{n\in T}_{i\in\mathbb{N}}, Sys \times Context^{n\in T}) \tag{40}$$

7.1.2 Identity of Self-* Facets

There exist identities $1_{n\in T} : (Sys \times Context^{n\in T}) \longrightarrow (Sys \times Context^{n\in T})$ of self-* facets in $\mathbf{Self\text{-}X}^{n\in T}$ such that, for every f in $\mathbf{Self\text{-}X}^{n\in T}$, $1_{n\in T}; f = f; 1_{n\in T} = f$ to be held. In other words, this can be specified by

$$\begin{aligned} Self\text{-}X^{n\in T}_{i\in\mathbb{N}} &= Self\text{-}X_{i\in\mathbb{N}}(1_{n\in T}, Sys \times Context^{n\in T}) \\ &= Self\text{-}X_{i\in\mathbb{N}}(Sys \times Context^{n\in T}, 1_{n\in T}) \\ &= Self\text{-}X_{i\in\mathbb{N}}(Sys \times Context^{n\in T}, Sys \times Context^{n\in T}) \end{aligned} \tag{41}$$

Thus, $\mathbf{Self\text{-}X}^{n\in T}$ with the composition operation "; " is called *self-* monoid*. Moreover, the monoid $\mathbf{Self\text{-}X}^{n\in T}$ is also a monoid category including only one object to be the set $\{Self\text{-}X^{n\in T}_{i\in\mathbb{N}}\}$, each of whose members is a self-* facet, and by the composition operation as a morphism, then the associativity and identity on the morphisms are completely satisfied.

7.2 A Category of Self-* Monoids

By the self-* monoids $\mathbf{Self\text{-}X}^{i\in T}$, we can construct $\mathbf{Cat}(\mathbf{Self\text{-}X})$ to be a category of self-* monoids. In fact, $\mathbf{Cat}(\mathbf{Self\text{-}X})$ is constructed as follows:

- *Objects*: $Obj(\mathbf{Cat}(\mathbf{Self\text{-}X}))$ is the set of self-* monoids $\mathbf{Self\text{-}X}^{i\in T}$. That is,

$$Obj(\mathbf{Cat}(\mathbf{Self\text{-}X})) = \{\mathbf{Self\text{-}X}^{i\in T}\} \tag{42}$$

- *Morphisms*: Associated with each object $\mathbf{Self\text{-}X}^{i\in T}$ in $Obj(\mathbf{Cat}(\mathbf{Self\text{-}X}))$, we define a morphism $\mathbf{Self\text{-}X}^{i\in T} \xrightarrow{1_{\mathbf{Self\text{-}X}^{i\in T}}} \mathbf{Self\text{-}X}^{i\in T}$, the identity morphism on $\mathbf{Self\text{-}X}^{i\in T}$ such that

$$\mathbf{Self\text{-}X}^{i\in T} \xrightarrow{1_{\mathbf{Self\text{-}X}^{i\in T}} \overset{def}{=} 1_{i\in T}} \mathbf{Self\text{-}X}^{i\in T} \tag{43}$$

or

$$\{Self\text{-}X^{i\in T}_{k\in\mathbb{N}}\} \xrightarrow{1_{\mathbf{Self\text{-}X}^{i\in T}} \overset{def}{=} 1_{i\in T}} \{Self\text{-}X^{i\in T}_{k\in\mathbb{N}}\} \tag{44}$$

and to each pair of morphisms $\textbf{Self-X}^{i\in T} \xrightarrow{f} \textbf{Self-X}^{j\in T}$ and $\textbf{Self-X}^{j\in T} \xrightarrow{g} \textbf{Self-X}^{j\in T}$ such that

$$\textbf{Self-X}^{i\in T} \xrightarrow{f \overset{def}{=} 1_{i\in T} \times Context^{j-i}} \textbf{Self-X}^{j\in T} \tag{45}$$

and

$$\textbf{Self-X}^{j\in T} \xrightarrow{g \overset{def}{=} 1_{j\in T} \times Context^{k-j}} \textbf{Self-X}^{k\in T} \tag{46}$$

there is an associated morphism $\textbf{Self-X}^{i\in T} \xrightarrow{f;g} \textbf{Self-X}^{k\in T}$, the composition of f with g, such that

$$\textbf{Self-X}^{i\in T} \xrightarrow{f;g = 1_{i\in T} \times Context^{k-i}} \textbf{Self-X}^{k\in T} \tag{47}$$

For every object in $Obj(\textbf{Cat}(\textbf{Self-X}))$ and the morphisms

$$\textbf{Self-X}^{i\in T} \xrightarrow{f \overset{def}{=} 1_{i\in T} \times Context^{j-i}} \textbf{Self-X}^{j\in T} \tag{48}$$

$$\textbf{Self-X}^{j\in T} \xrightarrow{g \overset{def}{=} 1_{j\in T} \times Context^{k-j}} \textbf{Self-X}^{k\in T} \tag{49}$$

and

$$\textbf{Self-X}^{k\in T} \xrightarrow{h \overset{def}{=} 1_{k\in T} \times Context^{m-k}} \textbf{Self-X}^{m\in T} \tag{50}$$

in $Arc(\textbf{Cat}(\textbf{Self-X}))$, the following equations hold:

Associativity: $(f;g);h = f;(g;h) = 1_{i\in T} \times Context^{m-i}$
Identity: $1_{\textbf{Self-X}^{i\in T}}; f = f = f; 1_{\textbf{Self-X}^{j\in T}}$
(i.e., $1_{i\in T}; 1_{i\in T} \times Context^{j-i} = 1_{i\in T} \times Context^{j-i} = 1_{i\in T} \times Context^{j-i}; 1_{j\in T}$)

As a result, the above-mentioned monoid morphisms can be diagrammatically drawn such as

$$\textbf{Self-X}^{i\in T} \xrightarrow{1_{i\in T} \times Context^{\pm k}} \textbf{Self-X}_{i\pm k\in T} \tag{51}$$

or

$$\{Self\text{-}X^{i\in T}_{l\in\mathbb{N}}\} \xrightarrow{1_{i\in T} \times Context^{\pm k}} \{Self\text{-}X^{i\pm k\in T}_{l\in\mathbb{N}}\} \tag{52}$$

These are all the basic ingredients we need to have the category **Cat**(**Self-X**). Let us see a general definition of category presented in Section 5 for reference.

7.3 Some Properties of Category **Cat(Self-X)**

By the construction of category **Cat**(**Self-X**), some emerging significant properties are presented in this subsection.

Property 1 All monoid morphisms of **Cat**(**Self-X**) are monoid isomorphisms.

Proof This result immediately stems from diagram (51). In fact, for every pair of monoid morphisms in *Arc*(**Cat**(**Self-X**)) between $\mathbf{Self\text{-}X}^{i\in T}$ and $\mathbf{Self\text{-}X}^{j\in T}$, we always have the following diagram:

$$\begin{tikzcd}
\mathbf{Self\text{-}X}^{i\in T} \arrow[loop below, "1_{i\in T}"] \arrow[rr, bend left, "1_{i\in T}\times Context^{j-i}"] & & \mathbf{Self\text{-}X}^{j\in T} \arrow[loop above, "1_{j\in T}"] \arrow[ll, bend left, "1_{j\in T}\times Context^{i-j}"]
\end{tikzcd} \tag{53}$$

These monoid morphisms satisfy an isomorphic relationship. Q.E.D.

Property 2 Isomorphisms between any pair of monoids in **Cat**(**Self-X**) are ever isomorphisms between the pair of ANCSs.

Proof This comes from the fact that each object of category **Cat**(**Self-X**) is just an ANCS. Q.E.D.

From the above-mentioned justification of **Cat**(**Self-X**), we are able to derive $\mathbf{Self\text{-}X}^{i\in T}$. Derivation of every $\mathbf{Self\text{-}X}^{i\in T}$ is simplified by the following facts:

Property 3 There exists always a self-* monoid **Self-X**, as simply as it can, in **Cat**(**Self-X**) constructed. Hence, it is available to start with.

Proof It emerges that

$$\begin{aligned}\mathbf{Self\text{-}X} &= \{Self\text{-}X_{i\in\mathbb{N}}(Sys\times Context^0, Sys\times Context^0)\} \\ &= \{Self\text{-}X_{i\in\mathbb{N}}(Sys, Sys)\}\end{aligned} \tag{54}$$

thus

$$1 \xrightarrow{\mathbf{Self\text{-}X}} Obj(\mathbf{Cat}(\mathbf{Self\text{-}X})) \tag{55}$$

Q.E.D.

Property 4 Given **Self-X**, we can compute $\mathbf{Self\text{-}X}^{i\in T}$.

Proof We evaluate self-* monoid $\mathbf{Self\text{-}X}^{i\in T}$ such that

$$1 \xrightarrow{\mathbf{Self\text{-}X}^{i\in T}} Obj(\mathbf{Cat}(\mathbf{Self\text{-}X})) \tag{56}$$

based on the facts that

$$\left(\begin{array}{c} 1 \xrightarrow{\quad\textbf{Self-X}\quad} Obj(\textbf{Cat}(\textbf{Self-X})) \\ \text{and} \\ \textbf{Self-X} \xrightarrow{\quad 1_0 \times Context^i \quad} \textbf{Self-X}^{i\in T} \end{array}\right) \tag{57}$$

Note that $\textbf{Self-X} \xrightarrow{1_0} \textbf{Self-X}$. Q.E.D.

Property 5 Given $\textbf{Self-X}^{i\in T}$, we can compute $\textbf{Self-X}^{j\in T}$ for every $j \neq i$.

Proof Self-* monoid $\textbf{Self-X}^{j\in T}$ is evaluated such that

$$1 \xrightarrow{\quad\textbf{Self-X}^{j\in T}\quad} Obj(\textbf{Cat}(\textbf{Self-X})) \tag{58}$$

based on the facts that

$$\left(\begin{array}{c} 1 \xrightarrow{\quad\textbf{Self-X}^{i\in T}\quad} Obj(\textbf{Cat}(\textbf{Self-X})) \\ \text{and} \\ \textbf{Self-X}^{i\in T} \xrightarrow{\quad 1_{i\in T} \times Context^{j-i} \quad} \textbf{Self-X}^{j\in T} \end{array}\right) \tag{59}$$

Q.E.D.

From the construction of **Cat**(**Self-X**), we see that every $\textbf{Self-X}^{i\in T}$ can be formed in the unifying way based on Properties 3–5. As a result, we gain a substantial procedure of construction at a high abstract level without any excessive inclination toward a specific implementation detail. This is quite helpful when we want to justify whether or not some certain properties of the construction are true. In fact, we can prove

Property 6 Every monoid $\textbf{Self-X}^{i\in T}$ can be constructed by any other monoid in **Cat**(**Self-X**)

Proof Applying Properties 3–5 to construct every monoid $\textbf{Self-X}^{i\in T}$ from another monoid in **Cat**(**Self-X**). Q.E.D.

This is certainly a property we expect of any construction procedure.

Property 7 **Cat**(**Self-X**) is a complete graph

Proof In fact, this is a consequence stemming from Property 6. Q.E.D.

This is indeed a property of our abstract construction mechanism.

8 Series of Self-* Facets

A number of different notations are in use for denoting series of self-* facets.

$$sf = (f_0, f_1, f_2, \ldots) \tag{60}$$

is a common notation which specifies a series of self-* facets *sf* which is indexed by the natural numbers in $T(= \mathbb{N} \cup \{0\})$. We are also accustomed to

$$sf = (f_{t \in T}) \tag{61}$$

Informally, series of self-* facets can be understood as a rope on which we hang up a sequence of self-* facets for display. Hence it follows that

Definition 6 (Series of self-* facets) *For morphisms* $1 \xrightarrow{t} T$ *and* $1 \xrightarrow{f_t} \textbf{Self-X}^{n \in T}$*, there exists a unique morphism* $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$ *such that the equation* $t; sf = f_t$ *holds. This is described by the following commutative diagram*

$$\begin{tikzcd}
1 \arrow[r, "t"] \arrow[dr, "f_t"'] & T \arrow[d, "sf"] \\
 & \textbf{Self-X}^{n \in T}
\end{tikzcd} \tag{62}$$

Morphism $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$ *defines a series of self-* facets.*

Explanation on semantics: Note that morphism $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$ is read as

$$\forall t[t \in T \Longrightarrow \exists!\ f_t[f_t \in \textbf{Self-X}^{n \in T}\ \&\ sf(t) = f_t]]$$

In other words, $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$ generates series of self-* facets as an infinite sequence of $sf(0) = f_0$, $sf(1) = f_1$, ..., $sf(t) = f_t$, ... which is written as $(sf(0), sf(1), \ldots, sf(t), \ldots)$ or $(f_0, f_1, \ldots, f_t, \ldots)$

Definition 7 (Set of series of self-* facets) *Given* $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$ *then the set of series of self-* facets, denoted by* $\textbf{Self-X}_{\omega}^{n \in T}$*, is defined by*

$$\textbf{Self-X}_{\omega}^{n \in T} = \{sf \mid T \xrightarrow{sf} \textbf{Self-X}^{n \in T}\} \tag{63}$$

We obtain

Corollary 1 *If* $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$ *then* $1 \xrightarrow{sf} \textbf{Self-X}_{\omega}^{n \in T}$

Proof This result stems immediately from Definitions 6 and 7. Q.E.D.

Explanation on semantics: This corollary means that for each morphism $T \xrightarrow{sf} \textbf{Self-X}^{n \in T}$, there is a morphism $1 \xrightarrow{sf} \textbf{Self-X}_{\omega}^{n \in T}$ generating member in $\textbf{Self-X}_{\omega}^{n \in T}$.

That is, morphism $T \xrightarrow{sf} \mathbf{Self\text{-}X}^{n\in T}$ generates series of self-* facets and $1 \xrightarrow{sf} \mathbf{Self\text{-}X}_{\omega}^{n\in T}$ constructs the set of series of self-* facets.

For series of self-* facets, we can define a mechanism to generate them. This mechanism consists of an object T equipping with structural morphisms $1 \xrightarrow{0} T \xrightarrow{succ} T$ with the property that for $\mathbf{Self\text{-}X}^{n\in T}$, any $1 \xrightarrow{f_0} \mathbf{Self\text{-}X}^{n\in T}$ and $\mathbf{Self\text{-}X}^{n\in T} \xrightarrow{next} \mathbf{Self\text{-}X}^{n\in T}$ then there exists a unique morphism $T \xrightarrow{sf} \mathbf{Self\text{-}X}^{n\in T}$ such that the following diagram commutes

$$\begin{tikzcd}
\mathbf{1} \arrow[r, "0"] \arrow[dr, "f_0"'] & T \arrow[r, "succ"] \arrow[d, "sf"] & T \arrow[d, "sf"] \\
 & \mathbf{Self\text{-}X}^{n\in T} \arrow[r, "next"'] & \mathbf{Self\text{-}X}^{n\in T}
\end{tikzcd} \tag{64}$$

Definition 8 (Construction of series of self-* facets) *We define a construction morphism of series of self-* facets, denoted by ‡, such that*

$$\mathbf{Self\text{-}X}^{n\in T} \times [T \xrightarrow{sf} \mathbf{Self\text{-}X}^{n\in T}] \xrightarrow{\ddagger} [T \xrightarrow{sf} \mathbf{Self\text{-}X}^{n\in T}] \tag{65}$$

Explanation on semantics: This definition means that $\ddagger(A \times B \xrightarrow{f\times g} C \times D) = A\ddagger B \xrightarrow{f\ddagger g} C\ddagger D$. It follows that any series of self-* facets $T \xrightarrow{sf} \mathbf{Self\text{-}X}^{n\in T}$ can be represented in a format including two parts of *head* and *tail* to be connected by "‡" such that

$$\begin{tikzcd}
T \arrow[r, "sf"] & \mathbf{Self\text{-}X}^{n\in T} \arrow[r, phantom, "\overset{equiv}{\equiv}"] & 1 \arrow[r, "0"] \arrow[rr, bend left, "f_0"] & T \arrow[r, "sf"] & \mathbf{Self\text{-}X}^{n\in T} \arrow[r, phantom, "\ddagger"] & 1 \arrow[r, "t>0"] \arrow[rr, bend left, "f_{t>0}"] & T \arrow[r, "sf"] & \mathbf{Self\text{-}X}^{n\in T}
\end{tikzcd} \tag{66}$$

where $\begin{tikzcd} 1 \arrow[r, "0"] \arrow[rr, bend left, "f_0"] & T \arrow[r, "sf"] & \mathbf{Self\text{-}X}^{n\in T} \end{tikzcd} = sf(0)$ and $\begin{tikzcd} 1 \arrow[r, "t>0"] \arrow[rr, bend left, "f_{t>0}"] & T \arrow[r, "sf"] & \mathbf{Self\text{-}X}^{n\in T} \end{tikzcd} = (sf(1), sf(2), \ldots)$ to be called head and tail, respectively.

Definition 9 (Head of series of self-* facets) *We define a head construction morphism, denoted by $1 \xRightarrow{0} (\_\,)$, such that*

$$1 \xRightarrow{0} (\_\,) : [T \xrightarrow{sf} \mathbf{Self\text{-}X}^{n\in T}] \longrightarrow \mathbf{Self\text{-}X}^{n\in T} \tag{67}$$

Explanation on semantics: This definition states that $\forall(a\ddagger s)[(a\ddagger s) \in [T \xrightarrow{sf} \textbf{Self-X}^{n\in T}] \Longrightarrow \exists!\, f_0[f_0 \in \textbf{Self-X}^{n\in T} \;\&\; 1 \xRightarrow{0} (a\ddagger s) = a = f_0]]$

It follows that $1 \xRightarrow{0} (T \xrightarrow{sf} \textbf{Self-X}^{n\in T}) \overset{equiv}{\equiv} 1 \xrightarrow{0} T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$.

Definition 10 (Tail of series of self-* facets) *We define a tail construction morphism, denoted by* $(\_\,)'$*, such that*

$$(\_\,)' : [T \xrightarrow{sf} \textbf{Self-X}^{n\in T}] \longrightarrow [T \xrightarrow{sf} \textbf{Self-X}^{n\in T}] \tag{68}$$

Explanation on semantics: This definition means that $\forall(a\ddagger s)[(a\ddagger s) \in [T \xrightarrow{sf} \textbf{Self-X}^{n\in T}] \Longrightarrow \exists!(f_1, f_2, \ldots)[(f_1, f_2, \ldots) \in [T \xrightarrow{sf} \textbf{Self-X}^{n\in T}] \;\&\; (a\ddagger s)' = s = (f_1, f_2, \ldots)]]$

As a convention, $(\_\,)^{\langle n\rangle}$ denotes applying recursively the $(\_\,)'$ n times. Thus, specifically, $(\_\,)^{\langle 2\rangle}, (\_\,)^{\langle 1\rangle}$, and $(\_\,)^{\langle 0\rangle}$ stand for $((\_\,)')', (\_\,)'$, and $(\_\,)$, respectively.

It follows that the first member of series of self-* facets $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ is given by

$$1 \xRightarrow{0} ((T \xrightarrow{sf} \textbf{Self-X}^{n\in T})') \overset{equiv}{\equiv} 1 \xrightarrow{1} T \xrightarrow{sf} \textbf{Self-X}^{n\in T} \tag{69}$$

and, in general, for every $k \in T$ the k-th member of series of self-* facets $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ is provided by

$$1 \xRightarrow{0} ((T \xrightarrow{sf} \textbf{Self-X}^{n\in T})^{\langle k\rangle}) \overset{equiv}{\equiv} 1 \xrightarrow{k} T \xrightarrow{sf} \textbf{Self-X}^{n\in T} \tag{70}$$

Series of self-* facets to be an infinite sequence of all $f_{t\in T}$ is viewed and treated as single mathematical entity, so the derivative of series of self-* facets $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ is given by $(T \xrightarrow{sf} \textbf{Self-X}^{n\in T})'$

Now using this notation for derivative of series of self-* facets, we can specify series of self-* facets $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ as in

Definition 11 *A series of self-* facets* $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ *can be specified by*

- *Initial value:* $1 \xrightarrow{0} T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ *and*
- *Differential equation:* $((T \xrightarrow{sf} \textbf{Self-X}^{n\in T})^{\langle n\rangle})' = (T \xrightarrow{sf} \textbf{Self-X}^{n\in T})^{\langle n+1\rangle}$

Explanation on semantics: The initial value of $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ is defined as its first element $1 \xrightarrow{0} T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$, and the derivative of series of self-* facets, denoted by $(T \xrightarrow{sf} \textbf{Self-X}^{n\in T})'$, is defined by $((T \xrightarrow{sf} \textbf{Self-X}^{n\in T})^{\langle n\rangle})' = (T \xrightarrow{sf} \textbf{Self-X}^{n\in T})^{\langle n+1\rangle}$, for any integer n in T. In other words, the initial value and derivative equal the head and tail of $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$, respectively. The behavior of a series of self-* facets $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ consists of two aspects: it allows for the observation of its initial value $1 \xrightarrow{0} T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$; and it can make an evolution to the new series of self-* facets $(T \xrightarrow{sf} \textbf{Self-X}^{n\in T})'$, consisting of the original series of self-* facets from which the first element has been removed. The initial value of $(T \xrightarrow{sf} \textbf{Self-X}^{n\in T})'$, which is $1 \xRightarrow{0} ((T \xrightarrow{sf} \textbf{Self-X}^{n\in T})') = 1 \xrightarrow{1} T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ can in its turn be observed, but note that we have to move from $T \xrightarrow{sf} \textbf{Self-X}^{n\in T}$ to $(T \xrightarrow{sf} \textbf{Self-X}^{n\in T})'$ first in order to do so. Now a behavioral differential equation defines a series of self-* facets by specifying its initial value together with a description of its derivative, which tells us how to continue.

Note: Every member $f_{t\in T}$ in $\textbf{Self-X}^{n\in T}$ can be considered as a series of self-* facets in the following manner. For every $f_{t\in T}$ in $\textbf{Self-X}^{n\in T}$, a unique series of self-* facets is defined by morphism f:

$$\begin{tikzcd} 1 \arrow[r, "f_t"] \arrow[rr, bend left, "{(f_t,\circ,\circ,\ldots)}"] & \textbf{Self-X}^{n\in T} \arrow[r, "f"] & \textbf{Self-X}_{\omega}^{n\in T} \end{tikzcd} \tag{71}$$

such that the equation $f_t; f = (f_i, \circ, \circ, \ldots)$ holds, where $\circ$ denotes empty member (or null member) in $\textbf{Self-X}^{n\in T}$. Thus $(f_t, \circ, \circ, \ldots)$ is in $\textbf{Self-X}_{\omega}^{n\in T}$.

Definition 12 (Equivalence) *For any* $T \xrightarrow{sf1} \textbf{Self-X}^{n\in T}$ *and* $T \xrightarrow{sf2} \textbf{Self-X}^{n\in T}$, *sf1 = sf2 iff* $1 \xrightarrow{t} T \xrightarrow{sf1} \textbf{Self-X}^{n\in T} = 1 \xrightarrow{t} T \xrightarrow{sf2} \textbf{Self-X}^{n\in T}$ *with every* t *in* T.

Definition 13 (Bisimulation) *Bisimulation on* $\textbf{Self-X}_{\omega}^{n\in T}$ *is a relation, denoted by* $\sim$, *between series of self-* facets* $T \xrightarrow{sf1} \textbf{Self-X}^{n\in T}$ *and* $T \xrightarrow{sf2} \textbf{Self-X}^{n\in T}$ *such that if* $sf1 \sim sf2$ *then* $1 \xRightarrow{0} (sf1) = 1 \xRightarrow{0} (sf2)$ *and* $(sf1)' \sim (sf2)'$.

Two series of self-* facets are bisimular if, regarding their behaviors, each of the series "simulates" the other and vice versa. In other words, each of the series cannot be distinguished from the other by the observation. Let us do the following exercises related to the bisimulation between series of self-* facets.

Exercise 10 Let sf, $sf1$ and $sf2$ be in $\mathbf{Self\text{-}X}_{\omega}^{n\in T}$. Show that if $sf \sim sf1$ and $sf1 \sim sf2$ then $(sf \sim sf1) \circ (sf1 \sim sf2) = sf \sim sf2$, where the symbol $\circ$ denotes a relational composition. For more descriptive notation, we can write this in the form

$$\frac{sf \sim sf1, sf1 \sim sf2}{(sf \sim sf1) \circ (sf1 \sim sf2) = sf \sim sf2} \tag{72}$$

and conversely, if $sf \sim sf2$ then there exists $sf1$ such that $sf \sim sf1$ and $sf1 \sim sf2$. This can be written as

$$\frac{sf \sim sf2}{\exists sf1 : sf \sim sf1 \quad \text{and} \quad sf1 \sim sf2} \tag{73}$$

Solution Proving (72) originates as the result of the truth that the relational composition between two bisimulations $L_1 \subseteq sf \times sf1$ and $L_2 \subseteq sf1 \times sf2$ is a bisimulation obtained by $L_1 \circ L_2 = \{\langle x, y\rangle \mid x\ L_1\ z \text{ and } z\ L_2\ y \text{ for some } z \in sf1\}$, where $x \in sf$, $z \in sf1$ and $y \in sf2$.

Proving (73) comes from the fact that there are always $sf1 = sf$ or $sf1 = sf2$ as simply as they can. Hence, (73) is always true in general. □

Exercise 11 Let $sf_i, \forall i \in \mathbb{N}$, be in $\mathbf{Self\text{-}X}_{\omega}^{n\in T}$ and $\bigcup_{i\in\mathbb{N}}$ be union of a family of sets. Show that

$$\frac{sf \sim sf_i \quad \text{with } i \in \mathbb{N}}{\bigcup_{i\in\mathbb{N}}(sf \sim sf_i) = sf \sim \bigcup_{i\in\mathbb{N}} sf_i} \tag{74}$$

and conversely,

$$\frac{sf \sim \bigcup_{i\in\mathbb{N}} sf_i}{\exists i \in \mathbb{N} : sf \sim sf_i} \tag{75}$$

Solution Proving (74) stems straightforwardly from the fact that sf bisimulates sf_i (i.e., $sf \sim sf_i$) then, sf bisimulates each series in $\bigcup_{i\in\mathbb{N}} sf_i$.

Conversely, proving (75) develops as the result of the fact that for each $\langle x, y\rangle \in \bigcup_{i\in\mathbb{N}}(sf \times sf_i)$, there exists $i \in \mathbb{N}$ such that $\langle x, y\rangle \in sf \times sf_i$. In other words, it is formally denoted by $\bigcup_{i\in\mathbb{N}}(sf \times sf_i) = \{\langle x, y\rangle \mid \exists i \in \mathbb{N} : x \in sf \quad \text{and} \quad y \in sf_i\}$, where $x \in sf$ and $y \in sf_i$. □

The union of all bisimulations between sf and sf_i (i.e., $\bigcup_{i\in\mathbb{N}}(sf \sim sf_i)$) is the greatest bisimulation. The greatest bisimulation is called the *bisimulation equivalence* or *bisimilarity* [23, 42] (again denoted by the notation $\sim$).

Exercise 12 Check that the bisimilarity $\sim$ on $\bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)$ is an equivalence relation.

Solution In fact, a bisimilarity $\sim$ on $\bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)$ is a binary relation $\sim$ on $\bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)$, which is reflexive, symmetric and transitive. In other words, the following properties hold for $\sim$

- Reflexivity:

$$\frac{\forall(a \sim b) \in \bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)}{(a \sim b) \sim (a \sim b)} \tag{76}$$

- Symmetry: $\forall(a \sim b), (c \sim d) \in \bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)$,

$$\frac{(a \sim b) \sim (c \sim d)}{(c \sim d) \sim (a \sim b)} \tag{77}$$

- Transitivity: $\forall(a \sim b), (c \sim d), (e \sim f) \in \bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)$,

$$\frac{((a \sim b) \sim (c \sim d)) \bigwedge ((c \sim d) \sim (e \sim f))}{(a \sim b) \sim (e \sim f)} \tag{78}$$

to be an equivalence relation on $\bigcup\limits_{i\in\mathbb{N}}(sf \sim sf_i)$. □

For some constraint α, if $sf1 \sim sf2$ then two series $sf1$ and $sf2$ have the following relation.

$$\frac{sf1 \models \alpha}{sf2 \models \alpha} \tag{79}$$

That is, if series $sf1$ satisfies constraint α then this constraint is still preserved on series $sf2$. Thus it is read as $sf1 \sim sf2$ in the constraint of α (and denoted by $sf1 \sim_\alpha sf2$).

For validating whether $sf1 = sf2$, a powerful method is so-called *proof principle of coinduction* [43] that states as follows:

Theorem 1 (Coinduction) *For any* $T \xrightarrow{sf1} \textbf{Self-X}^{n\in T}$ *and* $T \xrightarrow{sf2} \textbf{Self-X}^{n\in T}$, *if* $sf1 \sim sf2$ *then* $sf1 = sf2$.

Proof In fact, for two series of self-* facets $sf1$ and $sf2$ and a bisimulation $sf1 \sim sf2$. We see that by inductive bisimulation for $k \in T$, then $sf1^{\langle k\rangle} \sim sf2^{\langle k\rangle}$. Therefore, by Definition 13, $1 \xRightarrow{0} (sf1^{\langle k\rangle}) = 1 \xRightarrow{0} (sf2^{\langle k\rangle})$. By the equivalence in (70), then $1 \xrightarrow{k} sf1 = 1 \xrightarrow{k} sf2$ with every $k \in T$. It follows that, by Definition 12, we obtain $sf1 = sf2$. Q.E.D.

Hence in order to prove the equivalence between two series of self-* facets $sf1$ and $sf2$, it is sufficient to establish the existence of a bisimulation relation $sf1 \sim sf2$. In other words, using coinduction we can justify the equivalence between two series of self-* facets $sf1$ and $sf2$ in $\textbf{Self-X}_{\omega}^{n \in T}$.

Exercise 13 (Generating series of self-* facets) For every sf in $\textbf{Self-X}_{\omega}^{n \in T}$, show that

$$sf = 1 \overset{0}{\Longrightarrow} (sf) \ddagger (sf)' \tag{80}$$

Solution This stems from the coinductive proof principle in Theorem 1. In fact, it is easy to check the following bisimulation $sf \sim 1 \overset{0}{\Longrightarrow} (sf) \ddagger (sf)'$. It follows that $sf = 1 \overset{0}{\Longrightarrow} (sf) \ddagger (sf)'$ □

In (80), operation ‡ as a kind of series integration, the exercise states that series derivation and series integration are inverse operations. It gives a way to obtain sf from $(sf)'$ and the initial value $1 \overset{0}{\Longrightarrow} (sf)$. As a result, the exercise allows us to reach solution of differential equations in an algebraic manner.

9 Discussions and Comparisons

The aim of this chapter has been both to give an in-depth analysis as well as to present the new material on the notion of self-* computing. Below we briefly discuss the Wang's approach in [53] and compare it with our development.

- In the paper entitled "Toward Theoretical Foundations of Autonomic Computing" [53], a particular way of considering AC has been approached as a novel computing system at the highest level of machine intelligence, whose goal- and inference-driven computational behaviors have been expressed on top of imperative computing (IC) techniques with event-,time-, and interrupt-driven computational behaviors. By that approach, Wang has developed the overarching foundations and engineering paradigms of AC including the notions of behaviorism, cognitive informatics, denotational mathematics, and intelligent science. In particular, the paper has presented the theorems of the necessary and sufficient conditions of IC and AC, and the generic intelligence model of natural and machine intelligence for further dealing with advanced AC techniques and their engineering applications.

  However, by our approach, each computational behavior defined by Y. Wang in [53] becomes really an algebraic object of category. In addition, imperative computing systems, adaptive computing systems, and autonomic computing systems are just instances of a functor on such the category.

- In the considered context, we apply the category theory, which deals in an abstract way with algebraic objects and relationships between them for specifying interaction behaviors in ANCSs. For modeling, analyzing, and verifying the interaction behaviors, category theory is much better-approaching than other ones such as *process algebras* (or *process calculi*), FSM (Finite State Machine), or UML (Unified Modeling Language). In fact, the categorical approach becomes more powerful since process algebras and FSM are just of algebraic objects of category and UML is really a semi-formal approach. Categories were first described by Samuel Eilenberg and Saunders Mac Lane in 1945 [29], but have since grown substantially to become a branch of modern mathematics. Category theory spreads its influence over the development of both mathematics and theoretical computer science. The categorical structures themselves are still the subject of active research, including work to increase their range of practical applicability.

10 Conclusions

In this chapter, we have rigorously approached to the notion of self-* in ANCSs from which formal aspects of the self-* emerge.

We have started with investigating self-* in DANCSs and NANCSs, where we have modeled configuration of the system at every adaptation step as a member in the set $Sys \times Context^{i \in T}$, then self-* as a morphism from a configuration to another and self-* facet as a member of self-*. Moreover, $\mathbf{Self\text{-}X}^{i \in T}$ has been constructed as a self-* monoid to shape series $T \xrightarrow{sf} \mathbf{Self\text{-}X}^{i \in T}$ of self-* facets. By the self-* monoids, we have formed **Cat**(**Self-X**) to be a category of the self-* monoids for discovering the significant properties of the self-*.

Acknowledgments Thank you to the anonymous reviewers for their helpful comments and valuable suggestions which have contributed to the final preparation of the chapter. As always, I am deeply indebted to Professor Jonathan P. Bowen, Head of the Centre for Applied Formal Methods (CAFM) at London South Bank University (LSBU) in United Kingdom, for a constant source of inspiration and encouragement for the work which culminated in the publication of this chapter.

References

1. S. Abdelwahed and N. Kandasamy. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter A Control-Based Approach to Autonomic Performance Management in Computing Systems, pages 149–168. CRC Press, 1st edition, 2006.
2. J. Adamek, H. Herrlich, and G. Strecker. *Abstract and Concrete Categories*. John Wiley and Sons, 1990.
3. R. Adams et al. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Scalable Management – Technologies for Management of Large-Scale, Distributed Systems, pages 305–328. CRC Press, 1st edition, 2006.
4. R.R. Amoud et al. *Advanced Autonomic Networking and Communication*, chapter An Autonomic MPLS DiffServ-TE Domain, pages 149–168. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.

5. R. Anthony, A. Butler, and M. Ibrahim. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Exploiting Emergence in Autonomic Systems, pages 121–148. CRC Press, 1st edition, 2006.
6. A. Asperti and G. Longo. *Categories, Types and Structures*. M.I.T. Press, 1991.
7. G. M. Bergman. *An Invitation to General Algebra and Universal Constructions*. Henry Helson, 15 The Crescent, Berkeley CA 94708, USA, 1998.
8. V. Bhat, M. Parashar, and N. Kandasamy. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Autonomic Data Streaming for High-Performance Scientific Applications, pages 413–434. CRC Press, 1st edition, 2006.
9. D.W. Bustard and R. Sterritt. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter A Requirements Engineering Perspective on Autonomic Systems Development, pages 19–34. CRC Press, 1st edition, 2006.
10. W. Butera. Text Display and Graphics Control on a Paintable Computer. In G.D.M. Serugendo, J.P.M. Flatin, and M. Jelasity, editors, *Proceedings of 1st International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07)*, pages 45–54. IEEE Computer Society Press. Boston, MA, USA, 9–11 July 2007.
11. M. Calisti, R. Ghizzioli, and D. Greenwood. *Advanced Autonomic Networking and Communication*, chapter Autonomic Service Access Management for Next Generation Converged Networks, pages 101–126. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
12. M. Calisti, S.V.D. Meer, and J. Strassner, editors. *Advanced Autonomic Networking and Communication*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 2008. 190 pages.
13. A. Chakravarti, G. Baumgartner, and M. Lauria. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Self-Organizing Scheduling on the Organic Grid, pages 389–412. CRC Press, 1st edition, 2006.
14. J. Chen et al. *Advanced Autonomic Networking and Communication*, chapter Game Theoretic Framework for Autonomic Spectrum Management in Heterogeneous Wireless Networks, pages 169–190. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
15. D.M. Chess, J.E. Hanson, J.O. Kephart, I. Whalley, and S.R. White. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Dynamic Collaboration in Autonomic Computing, pages 253–274. CRC Press, 1st edition, 2006.
16. L. Durham, M. Milenkovic, P. Cayton, and M. Yousif. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Platform Support for Autonomic Computing: A Research Vehicle, pages 329–350. CRC Press, 1st edition, 2006.
17. C. Fahy et al. *Advanced Autonomic Networking and Communication*, chapter Modelling Behaviour and Distribution for the Management of Next Generation Networks, pages 43–62. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
18. A. Ganek. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Overview of Autonomic Computing: Origins, Evolution, Direction, pages 3–18. CRC Press, 1st edition, 2006.
19. D. Greenwood and R. Ghizzioli. *Advanced Autonomic Networking and Communication*, chapter Autonomic Communication with RASCAL Hybrid Connectivity Management, pages 63–80. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
20. R. Griffith, G. Valetto, and G. Kaiser. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Effecting Runtime Reconfiguration in Managed Execution Environments, pages 369–388. CRC Press, 1st edition, 2006.
21. T. Heinis, C. Pautasso, and G. Alonso. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter A Self-Configuring Service Composition Engine, pages 237–252. CRC Press, 1st edition, 2006.

22. IBM. Autonomic Computing Manifesto. Retrieved from http://www.research. ibm.com/ autonomic/, 2001.
23. B. Jacobs and J. Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *Bulletin of EATCS*, 62:222–259, 1997.
24. G. Jiang et al. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Trace Analysis for Fault Detection in Application Servers, pages 471–492. CRC Press, 1st edition, 2006.
25. X. Jin and J. Liu. From Individual Based Modeling to Autonomy Oriented Computation. In M. Nickles, M. Rovatsos, and G. Weiss, editors, *Agents and Computational Autonomy: Potential, Risks, and Solutions*, volume 2969 of *Lecture Notes in Computer Science*, pages 151 – 169. Springer Berlin, April 2004.
26. B. Khargharia and S. Hariri. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Autonomic Power and Performance Management of Internet Data, pages 435–470. CRC Press, 1st edition, 2006.
27. W. Kinsner. Towards Cognitive Machines: Multiscale Measures and Analysis. *The International Journal on Cognitive Informatics and Natural Intelligence (IJCINI)*, 1(1):28–38, 2007.
28. S. Ko, I. Gupta, and Y. Jo. Novel Mathematics-Inspired Algorithms for Self-Adaptive Peer-to-Peer Computing. In G.D.M. Serugendo, J.P.M. Flatin, and M. Jelasity, editors, *Proceedings of 1st International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07)*, pages 3–12. IEEE Computer Society Press. Boston, Massachusetts, USA, 9–11 July 2007.
29. F.W. Lawvere and S.H. Schanuel. *Conceptual Mathematics: A First Introduction to Categories*. Cambridge University Press, 1 st edition, 1997.
30. M. Levine. Categorical Algebra. In G. Benkart, T.S. Ratiu, H.A. Masur, and M. Renardy, editors, *Mixed Motives*, volume 57 of *Mathematical Surveys and Monographs*, chapter I, II, II of Part II, pages 373–499. American Mathematical Society, USA, 1998.
31. H. Liu and M. Parashar. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter A Programming System for Autonomic Self-Managing Applications, pages 211–236. CRC Press, 1st edition, 2006.
32. J.A.L López, J.M.G. Munoz, and J.M. Padial. *Advanced Autonomic Networking and Communication*, chapter A Telco Approach to Autonomic Infrastructure Management, pages 27–42. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
33. S.V.D. Meer et al. *Advanced Autonomic Networking and Communication*, chapter Technology Neutral Principles and Concepts for Autonomic Networking, pages 1–25. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
34. D.A. Menascé and M.N. Bennani. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Dynamic Server Allocation for Autonomic Service Centers in the Presence of Failures, pages 353–368. CRC Press, 1st edition, 2006.
35. G. Nguengang et al. *Advanced Autonomic Networking and Communication*, chapter Autonomic Resource Regulation in IP Military Networks: A Situatedness Based Knowledge Plane, pages 81–100. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
36. O. Pacheco. Autonomy in an Organizational Context. In M. Nickles, M. Rovatsos, and G. Weiss, editors, *Agents and Computational Autonomy: Potential, Risks, and Solutions*, volume 2969 of *Lecture Notes in Computer Science*, pages 195–208. Springer Berlin, April 2004.
37. M. Parashar. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Autonomic Grid Computing: Concepts, Requirements, and Infrastructure, pages 49–70. CRC Press, 1st edition, 2006.
38. M. Parashar and S. Hariri, editors. *Autonomic Computing: Concepts, Infrastructure and Applications*, 568 pages. CRC Press, 1st edition, December 2006.
39. G. Qu and S. Hariri. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Anomaly-Based Self Protection against Network Attacks, pages 493–522. CRC Press, 1st edition, 2006.

40. M.A. Razzaque, S. Dobson, and P. Nixon. *Advanced Autonomic Networking and Communication*, chapter Cross-layer Optimisations for Autonomic Networks, pages 127–148. Whitestein Series in Software Agent Technologies and Autonomic Computing. Springer-Verlag, 1st edition, 2008.
41. R.V. Renesse and K.P. Birman. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Autonomic Computing: A System-Wide Perspective, pages 35–48. CRC Press, 1st edition, 2006.
42. J.J.M.M. Rutten. Universal Coalgebra: A Theory of Systems. *Theoretical Computer Science*, 249(1):3–80, 17 October 2000.
43. J.J.M.M. Rutten. Elements of Stream Calculus (An Extensive Exercise in Coinduction). *Electronic Notes in Theoretical Computer Science*, 45, 2001. Elsevier Science Publishers Ltd.
44. S.M. Sadjadi and P.K. McKinley. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Transparent Autonomization in Composite Systems, pages 169–188. CRC Press, 1st edition, 2006.
45. K. Schwan et al. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter AutoFlow: Autonomic Information Flows for Critical Information Systems, pages 275–304. CRC Press, 1st edition, 2006.
46. P. Steenkiste and A.C. Huang. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Recipe-Based Service Configuration and Adaptation, pages 189–208. CRC Press, 1st edition, 2006.
47. J.W. Sweitzer and C. Draper. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter Architecture Overview for Autonomic Computing, pages 71–98. CRC Press, 1st edition, 2006.
48. P.C. Vinh. *Formal Aspects of Dynamic Reconfigurability in Reconfigurable Computing Systems*. PhD thesis, London South Bank University, 103 Borough Road, London SE1 0AA, UK, 4 May 2006.
49. P.C. Vinh. Homomorphism between AOMRC and Hoare Model of Deterministic Reconfiguration Processes in Reconfigurable Computing Systems. *Scientific Annals of Computer Science*, (XVII):113–145, 2007.
50. P.C. Vinh and J.P. Bowen. A Formal Approach to Aspect-Oriented Modular Reconfigurable Computing. In *Proceedings of 1st IEEE & IFIP International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 369–378. IEEE Computer Society Press. Shanghai, China, 6–8 June 2007.
51. P.C. Vinh and J.P. Bowen. Formalization of Data Flow Computing and a Coinductive Approach to Verifying Flowware Synthesis. *LNCS Transactions on Computational Science*, 1(4750):1–36, June 2008.
52. Y. Wang. Exploring machine cognition mechanisms for autonomic computing. *The International Journal on Cognitive Informatics and Natural Intelligence (IJCINI)*, 1(2):i–v, 2007.
53. Y. Wang. Toward Theoretical Foundations of Autonomic Computing. *The International Journal of Cognitive Informatics and Natural Intelligence (IJCiNi)*, 1(3):1–16, July–September 2007.
54. M. Witkowski and K. Stathis. A Dialectic Architecture for Computational Autonomy. In M. Nickles, M. Rovatsos, and G. Weiss, editors, *Agents and Computational Autonomy: Potential, Risks, and Solutions*, volume 2969 of *Lecture Notes in Computer Science*, pages 261–273. Springer Berlin, April 2004.
55. T.D. Wolf and T. Holvoet. *Autonomic Computing: Concepts, Infrastructure and Applications*, chapter A Taxonomy for Self-* Properties in Decentralized Autonomic Computing, pages 101–120. CRC Press, 1st edition, 2006.
56. B. Yang and J. Liu. An Autonomy Oriented Computing (AOC) Approach to Distributed Network Community Mining. In G.D.M. Serugendo, J.P.M. Flatin, and M. Jelasity, editors, *Proceedings of 1st International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07)*, pages 151–160. IEEE Computer Society Press, Boston, MA, USA, 9–11 July 2007.