Formal assessment of some properties of Context-Aware Systems

Fabio A. Schreiber

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

and

Maria Elena Valcher

Department of Information Engineering, University of Padova Padova, Italy

Context-Aware systems are becoming useful components in autonomic and monitoring applications and the assessment of their properties is an important step towards reliable implementation, especially in safety-critical applications. In this paper, using an avalanche/landslide alert system as a running example, we propose a technique, based on Boolean Control Networks, to verify that the system dynamics has stable equilibrium states, corresponding to constant inputs, and hence it does not exhibit oscillatory behaviors, and to establish other useful properties in order to implement a precise and timely alarm system.

Keywords: Boolean Control Networks (BCN), Context-Aware Systems, Fault detection, Formal properties, Pervasive Systems, Reconstructibility, Stability assessment.

1. INTRODUCTION

In the first days of 2017 a hotel has been hit by an avalanche in Abruzzo, an Italian region, causing the death of 29 people. Preliminary technical findings stated that the incident was triggered by a series of earthquakes in central Italy, in conjunction with the raise of the atmospheric temperature that melted the snow. In such cases, when the alarm is spread, the security manager is confronted with two alternatives: i) evacuate the site, which often is a cumbersome and uncomfortable operation which spreads discontent among the guests if it results in a false alarm; ii) do nothing, in the hope that nothing dangerous will happen. In this paper, we address the problem of formalizing, by means of a context-Aware (C-A) system, a decision process to avoid this type of tragedies as well as false alarms. In addition, by making use of the algebraic approach to Boolean Control Networks, we are able to assess the existence of globally attractive equilibrium points of the overall decision system, corresponding to constant inputs, and to investigate some interesting structural properties, that formalize system features of great practical relevance. In detail, in Section 2 we introduce the background upon which our research is founded; in Section 3 we describe the system architecture and in Section 4 we briefly introduce the Boolean Control Networks algebraic description. Section 5 describes the BCN model for the hydrogeological example. In Section 6 it is shown that the BCN exhibits only globally attractive equilibrium points, and no limit cycles, corresponding to constant inputs; in Section 7 the observability and reconstructibility of the system are considered; Section 8 examines the possibility of identifying some kinds of faults in the inputs that could result in errors in the alarm system, and, in Section 9, some conclusive remarks and hints for future work are made.

2. RELATED WORKS

Context-aware computing was born out of the need to master the complexity and enhance the flexibility of modern computing and information systems. Among the most widely used definitions of Context, and of Context-aware Computing, those proposed by A. Dey [Dey 2001] state: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."* and *"A system is Context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task."*

Accordingly, sophisticated and general Context models have been proposed, to support Context-aware applications that use them to: (i) tailor the set of application-relevant data, (ii) increase the precision of information retrieval, (iii) discover services, (iv) build smart environments, and others [Bolchini et al. 2007].

In the domains of Databases and Programming Languages, the design of Context-aware and Self-adapting systems has been frequently based on the separation between Context and functional system [Bolchini et al. 2009; Schreiber and Panigati 2017]. Even if a holistic view of the Context-aware system in which the Context and the functional system variables sets are kept together is possible, a separation of the two sets has been advocated, by using a component-based approach, to master the growing complexity of modern software systems and enforcing the separation of concerns [Djoudi et al. 2016].

The introduction of Context-awareness and Self-adaptation in safety-critical applications arose the need of specifying and assessing their properties, mainly those related to the system dependability, by means of formal methods such as Bigraphs and model-checking [Djoudi et al. 2016; Cherfia et al. 2014]. Owing to the dynamic nature of self-adapting systems, *stability* has drawn great attention among the features affecting dependability. Nzekwa et Al. [Nzekwa et al. 2010] propose the composition of different mechanisms to obtain a flexible model for implementing stabilization in Context-aware systems.

In [Padovitz et al. 2004; 2005] Padovitz et Al. consider a state-space approach for describing the *situation* dimension and for determining the likelihood of transitions between *situation subspaces*, all other Context dimensions remaining constant. In their model, the state variables are constituted by the system's sensors outputs. In an analog system, many sets of sensors values, representing a system state, can belong to the same situation subspace as far as they satisfy the conditions in its defining expression; a transition starts whenever one or more values change in such a way as to respectively switch the expressions for two situation subspaces from TRUE to FALSE and vice-versa. The likelihood of the transition is evaluated by assuming notions analogous to those of velocity and acceleration in mechanical systems, and on the basis of the distance of the values of the actual situation from those of its boundary.

Stability is a traditional topic in control systems theory, and in [Diao et al. 2005] the authors explore *"... the extent to which control theory can provide an architectural and analytic foundation for building self-managing systems ..."*. However, control systems are typically described by means of differential equations and by Matrix Algebra, while Context-aware systems are digital and mostly based on Logics. Inspired by biological systems, Boolean Networks (BN) and Boolean Control Networks (BCN) have been introduced, their representative equations have been converted into an equivalent algebraic form [Cheng and Qi 2010a; 2010b], and solutions to problems such as controllability, observability, stability and reconstructibility have been proposed [Cheng and Qi 2009; Fornasini and Valcher 2013a; 2016].

We think that cooperation between the two disciplines can be fruitful, therefore, to fill the model gap, in this paper we model the Context as well as the functional system as Boolean Control Networks, as briefly explained in Section 5.

3. THE ARCHITECTURE OF THE MONITORING SYSTEM

Figure 1 shows the general architecture of a Context-aware system [Bolchini et al. 2007; Dey 2001] conceived for monitoring possible snow/ground slides. Signals, coming from physical sensors on the ground, are evaluated in the context of the seismic and meteorological information provided by Web Services RSSs - which can suggest immediate danger - in order to issue alarms. Combining the Context state with the actual physical data that are input to the functional system allows the design of a flexible and effective prevention information system which, as an example, can distinguish between the vibration caused by the detachment of a snow mass and that caused by a skier or a deer occasionally passing near a sensor.

In the monitoring system some states produce outputs that can affect the environment, e.g. by possibly activating an alarm siren. In case of an alarm, the time to evacuate a hotel can be in the order of hours, while the seismic and meteorological conditions can change faster. The ultimate goal of this study is to be sure that in dangerous situations an alarm signal is issued, but at the same time that frequent changes in the Context State do not induce an oscillatory behavior of the alarm system and the consequent movement of people out and back into the hotel. The designer of the C-A system must ensure that no action is started before the preceding one is terminated (e.g. reducing the evacuation time). In this paper we use a simple open-loop model; however, in more complex C-A self managing applications, the system output can affect the context itself.

Our aim is therefore:

—To describe a Context-Aware (C-A) system, as in Figure 1, by means of a logic State Space model; web services provide input messages to the Context and sensors provide input signals to the Monitoring System; the Context state is a further input to the latter.
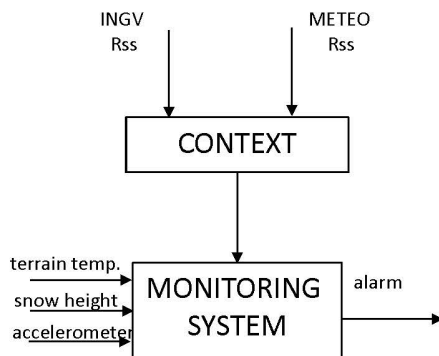

. Fig. 1 Open loop Context-Aware system

4. ALGEBRAIC REPRESENTATION OF BOOLEAN CONTROL NETWORKS

We consider Boolean vectors and matrices, taking values in $\mathcal{B} = \{0,1\}$, with the usual logical operations (And $\wedge$, Or $\vee$, Negation $\neg$). δ_k^i denotes the ith canonical vector of size k, namely the ith column of the k-dimensional identity matrix I_k. Δ_k is the set of all k-dimensional canonical vectors, and $\mathcal{L}_{k \times n} \subset \mathcal{B}^{k \times n}$ the set of all $k \times n$ logical matrices, namely $k \times n$ matrices whose

columns are canonical vectors of size k. Any matrix $L \in \mathcal{L}_{k \times n}$ can be represented as a row vector whose entries are canonical vectors in $\mathcal{L}_k$, namely $L = \begin{bmatrix} \delta_k^{i_1} & \delta_k^{i_2} & \dots & \delta_k^{i_n} \end{bmatrix}$, for suitable indices $i_1, i_2, \dots, i_n \in [1, k]$. $[A]_{\ell j}$ is the (ℓ, j)th entry of the matrix A. A nonsingular square matrix in $\mathcal{L}_{k \times k}$ is a *permutation matrix*.

There is a bijective correspondence between Boolean variables $X \in \mathcal{B}$ and vectors $\mathbf{x} \in \Delta_2$, defined by the relationship

$$\mathbf{x} = \begin{bmatrix} X \\ \overline{X} \end{bmatrix}.$$

We introduce the *(left) semi-tensor product* $\ltimes$ between matrices (and hence, in particular, vectors) as follows [Cheng et al. 2011]: given $L_1 \in \mathcal{L}_{r_1 \times c_1}$ and $L_2 \in \mathcal{L}_{r_2 \times c_2}$, we set

$$L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), \quad \text{where} \quad T := \text{l.c.m.}\{c_1, r_2\}.$$

The semi-tensor product represents an extension of the standard matrix product, by this meaning that if $c_1 = r_2$, then $L_1 \ltimes L_2 = L_1 L_2$. Note that if $\mathbf{x}_1 \in \Delta_{r_1}$ and $\mathbf{x}_2 \in \Delta_{r_2}$, then $\mathbf{x}_1 \ltimes \mathbf{x}_2 \in \Delta_{r_1 r_2}$. By resorting to the semi-tensor product, we can extend the previous correspondence to a bijective correspondence [Cheng et al. 2011] between $\mathcal{B}^n$ and $\mathcal{L}_{2^n}$. This is possible in the following way: given $X = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}^\top \in \mathcal{B}^n$ set

$$\mathbf{x} := \begin{bmatrix} X_1 \\ \overline{X}_1 \end{bmatrix} \ltimes \begin{bmatrix} X_2 \\ \overline{X}_2 \end{bmatrix} \ltimes \dots \ltimes \begin{bmatrix} X_n \\ \overline{X}_n \end{bmatrix}.$$

This corresponds to

$$\mathbf{x} = \begin{bmatrix} X_1 X_2 \dots X_{n-1} X_n \\ X_1 X_2 \dots X_{n-1} \overline{X}_n \\ X_1 X_2 \dots \overline{X}_{n-1} X_n \\ \vdots \\ \overline{X}_1 \overline{X}_2 \dots \overline{X}_{n-1} \overline{X}_n \end{bmatrix}.$$

A *Boolean Control Network* (BCN) is a logic state-space model taking the following expression:

$$\begin{aligned} X(t+1) &= f(X(t), U(t)), \\ Y(t) &= h(X(t), U(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

where $X(t)$, $U(t)$ and $Y(t)$ are the n-dimensional state variable, the m-dimensional input variable and the p-dimensional output variable at time t, taking values in $\mathcal{B}^n$, $\mathcal{B}^m$ and $\mathcal{B}^p$, respectively. f and h are logic functions, i.e. $f : \mathcal{B}^n \times \mathcal{B}^m \to \mathcal{B}^n$, while $h : \mathcal{B}^n \times \mathcal{B}^m \to \mathcal{B}^p$. By resorting to the semi-tensor product $\ltimes$, the BCN (1) can be described as [Cheng et al. 2011]

$$\begin{aligned} \mathbf{x}(t+1) &= L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ \mathbf{y}(t) &= H \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{2}$$

where $L \in \mathcal{L}_{2^n \times 2^{n+m}}$ and $H \in \mathcal{L}_{2^p \times 2^{n+m}}$. This is called the *algebraic expression* of the BCN. The matrix L can be partitioned into 2^m square blocks of size 2^n, namely as

$$L = \begin{bmatrix} L_1 & L_2 & \dots & L_{2^m} \end{bmatrix}.$$

For every $i \in \{1, 2, \dots, 2^m\}$, the matrix $L_i \in \mathcal{L}_{2^n \times 2^n}$ represents the logic matrix that relates $\mathbf{x}(t+1)$ to $\mathbf{x}(t)$, when $\mathbf{u}(t) = \delta_{2^n}^i$, namely

$$\mathbf{u}(t) = \delta_{2^m}^i \;\; \Rightarrow \;\; \mathbf{x}(t+1) = L_i \mathbf{x}(t).$$

It is worth remarking that the previous algebraic expression (2) can be adopted to represent any state-space model in which the state and input variables take values in finite sets, and hence the sizes of the state and input vectors need not be powers of 2. In that case oftentimes BCNs are called multi-valued Control Networks [Cheng et al. 2011]. With an abuse of terminology, in this paper we will always refer to them as BCNs. Moreover, we will replace $2^n, 2^m$ and 2^p with the generic symbols N, M and P.

5. THE BCN MODEL OF THE HYDROGEOLOGICAL EXAMPLE

5.1 The Context model

5.1.1 *Context Input Variables.* The values of the Context Input Variables are supplied by RSS messages coming from National Web Services, such as Meteorological forecasts and the National Geophysics Institute. Even if the message frequency can be variable, for ease of modeling, we suppose that the system samples them with the same frequency. Moreover, we suppose that a real danger situation can be expected only when a defined number - in our example at least four - of consecutive earthquake announcements are sent together with a snow forecast.
We assume:

INGV$\{earthquake, \neg earthquake\} := $ U$_1$
METEO$\{snow, \neg snow\} :=$ U$_2$

Therefore, by expressing the context input variables in terms of canonical vectors, we get:

Context Input Vector:

$$\mathbf{u}(\text{t}) = \begin{vmatrix} U_1 & U_2 \\ U_1 & \neg U_2 \\ \neg U_1 & U_2 \\ \neg U_1 & \neg U_2 \end{vmatrix} \in \Delta_4 := \{\delta_4^1, \delta_4^2, \delta_4^3, \delta_4^4\}$$

5.1.2 *Context States.* As previously mentioned, we assume that simultaneous snow and earthquake alerts can be regarded as reliable only if not isolated, namely if a sufficiently high number of consecutive (simultaneous) alerts are sent (and received). For this reason we introduce as Context State a counter:

COUNTER$\{0, 1, 2, 3, > 3\} =:$ C

In the representation by means of canonical vectors, the counter is denoted by $\mathbf{c}$ and takes values in $\Delta_5 := \{\delta_5^1, \delta_5^2, \delta_5^3, \delta_5^4, \delta_5^5\}$, depending on how many consecutive simultaneous alerts for snow and earthquake have been received. Specifically, for $i = 1, 2, 3, 4$, we have that $\mathbf{c}(t) = \delta_5^i$ if the counter is $i - 1$ at time t, while $\mathbf{c}(t) = \delta_5^5$ if the counter is at least 4 at time t.
If the counter at time t has a value in $\{\delta_5^1, \delta_5^2, \delta_5^3, \delta_5^4\}$ and the context input is $\mathbf{u}(t) = \delta_4^1$ (another simultaneous snow and earthquake alert comes in), then the counter value at $t + 1$ is increased by 1. If $\mathbf{c}(t) = \delta_5^5$ and $\mathbf{u}(t) = \delta_4^1$, then $\mathbf{c}(t+1) = \delta_5^5$, while in every other case the counter is reset[1] to $\mathbf{c}(t+1) = \delta_5^1$.
Therefore, the counter updates according to the following model (BCN):
$\mathbf{c}(t+1) = \mathbf{C} \ltimes \mathbf{u}(t) \ltimes \mathbf{c}(t)$, where

$C = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 \end{bmatrix} \in \mathcal{L}_{5 \times 20}$, and

$C_1 = C \ltimes \delta_4^1 = [\delta_5^2 \ \delta_5^3 \ \delta_5^4 \ \delta_5^5 \ \delta_5^5]$
$C_2 = C \ltimes \delta_4^2 = [\delta_5^1 \ \delta_5^1 \ \delta_5^1 \ \delta_5^1 \ \delta_5^1]$
$C_3 = C \ltimes \delta_4^3 = C_2$
$C_4 = C \ltimes \delta_4^4 = C_2$

Obviously, *the number of consecutive alert situations is a design variable which allows to set more stringent - if increased - or more relaxed - if lowered - requirements on the alarm system.*

---

[1] This is one possible solution, but it may be regarded as somewhat dangerous: if the counter gets erroneously reset, then the alert ends up being significantly delayed. An alternative solution could be that of simply decreasing by one the counter if $\mathbf{u}(t) \neq \delta_4^1$ (or if $\mathbf{u}(t) = \delta_4^i, i = 2, 3$). This solution would be more robust to possible disturbances occasionally affecting the context inputs.

5.1.3 *Context Output.* Introduce the Context model output

CONTEXT-ALERT$\{danger, quiet\} := U_c$

We assume that the CONTEXT-ALERT variable U_c is *danger* (the corresponding canonical vector $\mathbf{u}_c$ takes the value δ_2^1) if and only if there have been at least four simultaneous snow and earthquake alerts.
So, the variable $\mathbf{u}_c$ updates following the algebraic rule:
$\mathbf{u}_c(t) = \mathbf{H_c} \ltimes \mathbf{c}(t)$
where
$\mathbf{H_c} = \begin{bmatrix} \delta_2^2 & \delta_2^2 & \delta_2^2 & \delta_2^2 & \delta_2^1 \end{bmatrix} \in \mathcal{L}_{2 \times 5}$

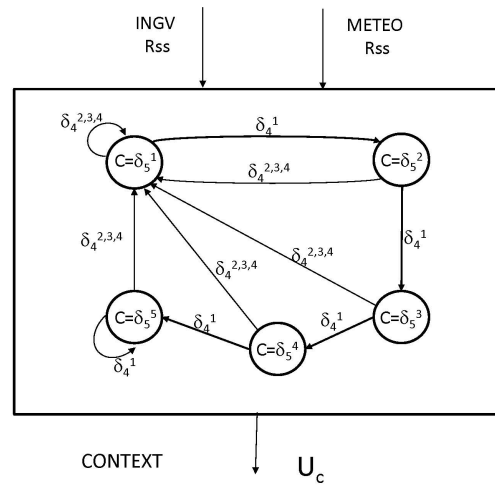Figure 2 shows the state diagram for the Context automaton.

.   Fig. 2 Context State diagram for the hydrogeological example

5.2   The Functional System model

5.2.1 *Functional System Input Variables.* We assume that, in addition to the CONTEXT-ALERT variable, the Functional System model receives other three input signals from local sensors, so that at the end the input variables determining the system dynamics are the following ones:

terrain temperature $\{high, low\} := V_1$
snow height $\{high, low\} := V_2$
accelerometer $\{high, low\} := V_3$
context-alert $\{danger, quiet\} := V_4 = U_c$

Input Vector:

$$\mathbf{v}(t)= \begin{vmatrix} V_1 & V_2 & V_3 & V_4 \\ V_1 & V_2 & V_3 & \neg V_4 \\ V_1 & V_2 & \neg V_3 & V_4 \\ V_1 & V_2 & \neg V_3 & \neg V_4 \\ V_1 & \neg V_2 & V_3 & V_4 \\ V_1 & \neg V_2 & V_3 & \neg V_4 \\ V_1 & \neg V_2 & \neg V_3 & V_4 \\ V_1 & \neg V_2 & \neg V_3 & \neg V_4 \\ \neg V_1 & V_2 & V_3 & V_4 \\ \neg V_1 & V_2 & V_3 & \neg V_4 \\ \neg V_1 & V_2 & \neg V_3 & V_4 \\ \neg V_1 & V_2 & \neg V_3 & \neg V_4 \\ \neg V_1 & \neg V_2 & V_3 & V_4 \\ \neg V_1 & \neg V_2 & V_3 & \neg V_4 \\ \neg V_1 & \neg V_2 & \neg V_3 & V_4 \\ \neg V_1 & \neg V_2 & \neg V_3 & \neg V_4 \end{vmatrix} \in \Delta_{16}$$

5.2.2 *Functional System State.* The CONTEXT-ALERT input is already the result of repeated and consecutive notifications of alert situations, so we may regard it as a variable that is hardly affected by false alarms. Also, we assume that a disturbance that can instantaneously modify the terrain temperature or the snow height, unless connected with an earthquake, is statistically not very realistic. On the other hand, the accelerometer may be a source of false alarms since it can detect a "high" signal for reasons that are not related to earthquakes: for instance, animals running close to the accelerometer. As a result, we regard as reliable only repeated alerts coming from the accelerometer. So, as in the case of simultaneous snow and earthquake warnings, we require, for instance, that the accelerometer has been "high" for two consecutive time instants (before t) in order to regard the information given by the accelerometer as a real warning.
We introduce the state variable:

ACC-COUNTER$\{0, 1, > 1\}$

The canonical vector representing the accelerometer counter is denoted by $\mathbf{a}$ and takes values in $\Delta_3 = \{\delta_3^1, \delta_3^2, \delta_3^3\}$. Specifically, $\mathbf{a}(t) = \delta_3^1$ if the counter is 0 at time t; $\mathbf{a}(t) = \delta_3^2$ if the counter is 1 at time t; and $\mathbf{a}(t) = \delta_3^3$ if the counter is at least 2 at time t.
If the counter at time t has a value in $\{\delta_3^1, \delta_3^2\}$ and the accelerometer vector is $\mathbf{v}_3(t) = \delta_2^1$, then the counter value at $t+1$ is increased by 1. If $\mathbf{a}(t) = \delta_3^3$ and $\mathbf{v}_3(t) = \delta_2^1$, then $\mathbf{a}(t+1) = \delta_3^3$, while when $\mathbf{v}_3(t) = \delta_2^2$ the counter is moved back to $\mathbf{a}(t+1) = \delta_3^1$.
Therefore, the accelerometer counter updates according to the following BCN:

$\mathbf{a}(t+1) = \mathbf{A} \ltimes \mathbf{v_3}(t) \ltimes \mathbf{a}(t)$, where

$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \in \mathcal{L}_{3 \times 6}$, and

$A_1 = A \ltimes \delta_2^1 = [\delta_3^2 \ \delta_3^3 \ \delta_3^3]$
$A_2 = A \ltimes \delta_2^2 = [\delta_3^1 \ \delta_3^1 \ \delta_3^1]$

5.2.3 *Functional System Output.* We assume that the Functional System output can take three values:

ALARM $\{temp-high, snow-high, acc-counter > 1, acc-high, ctx-danger\}$
ATTENTION $\{temp-low, snow-high, acc-counter-*, acc-*, ctx-* \ OR \ temp-high, snow-$

$low, acc - counter - *, acc - *, ctx - * \ OR \ temp - high, snow - high, acc - counter - low, acc -$
$*, ctx - * \ OR \ temp - high, snow - high, acc - counter - *, acc - low, ctx - * \ OR \ temp - high, snow -$
$high, acc - counter - *, acc - *, ctx - quiet\}$
NORMAL $\{temp - low, snow - low, acc - counter - *, acc - *, ctx - *\}$

Note that the alarm is sent out only when "$acc - counter > 1$" *and* "acc-high". This means
that at the time t^* the alarm signal is issued if the accelerometer has detected some movement
for at least three consecutive time instants t^*, $t^* - 1$ and $t^* - 2$. Of course, as for the context-
alert variable, *the choice of how long we want to wait before issuing the alarm signal is a design
parameter that balances conflicting requirements: security on the one hand and the need to avoid
false alarms on the other.*

The functional system output is denoted by $\mathbf{m}$ and takes values in Δ_3. Based on the previous
description of the three possible output values, it follows that the output vector is generated
based on the state $\mathbf{a}(t)$ and the input $\mathbf{v}(t)$ according to the following model :

$\mathbf{m}(t) = \mathbf{M} \ltimes \mathbf{v}(t) \ltimes \mathbf{a}(t)$, where

$\mathbf{M} = \begin{bmatrix} M_1 & M_2 & \dots & M_{16} \end{bmatrix} \in \mathcal{L}_{6 \times 16}$, and

$M_1 = M \ltimes \delta_{16}^1 = [\delta_3^2 \ \delta_3^2 \ \delta_3^1]$
$M_2 = M \ltimes \delta_{16}^2 = [\delta_3^2 \ \delta_3^2 \ \delta_3^2]$
$M_i = M \ltimes \delta_{16}^i = M_2, \quad \text{for} \quad i = 3, \dots, 12$
$M_{13} = M \ltimes \delta_{16}^{13} = [\delta_3^3 \ \delta_3^3 \ \delta_3^3]$
$M_i = M \ltimes \delta_{16}^i = M_{13}, \quad \text{for} \quad i = 14, 15, 16$

So, overall, the system model is a Boolean Control Network obtained by connecting the BCN
describing the context and the BCN describing the functional model, and hence it is described
by the following equations:

$$\begin{aligned}
\mathbf{c}(t+1) &= \mathbf{C} \ltimes \mathbf{u}(t) \ltimes \mathbf{c(t)} & (3) \\
\mathbf{a}(t+1) &= \mathbf{A} \ltimes \mathbf{v_3(t)} \ltimes \mathbf{a(t)} & (4) \\
\mathbf{v}_4(t) &= \mathbf{H_c} \ltimes \mathbf{c}(t) & (5) \\
\mathbf{m}(t) &= \mathbf{M} \ltimes \mathbf{v}(t) \ltimes \mathbf{a}(t). & (6)
\end{aligned}$$

Note that the previous system could be represented as a standard BCN having $\mathbf{u}(t) := \mathbf{u}(t) \ltimes$
$\mathbf{v}_1(t) \ltimes \mathbf{v}_2(t) \ltimes \mathbf{v}_3(t)$ as input, $\mathbf{x}(t) := \mathbf{c}(t) \ltimes \mathbf{a}(t)$ as state vector, and $\mathbf{y}(t) = \mathbf{m}(t)$ as output. Such
a representation, however, would be of larger dimension and would not contribute to a better
understanding of the system properties. On the contrary, it would make the overall analysis more
complicated. So, we investigate the model properties by making use of the previous description
(3)-(6). This provides further evidence of the convenience of using Context-Aware systems to
model the system dynamics. Note that the current cascade structure, having two counter variables
as state variables of the two connected BCNs, can be easily adapted to model a large class of
Context-Aware systems that describe a decision process, in particular, an alert system. So, even
if we focus on this specific model, it is immediate to understand how the results and properties
derived in the following extend to all the alert systems that can be modeled along these same
lines.

Let us start by investigating the equilibrium points corresponding to constant inputs.

6. EQUILIBRIA CORRESPONDING TO CONSTANT INPUTS

Definitions and methods to find equilibrium points in a system modeled as a BN have been described in detail in [Cheng et al. 2011; Cheng et al. 2011; Fornasini and Valcher 2013b] to which we refer for further deepening. In this paper we make use of these concepts and characterizations to address equilibrium points of BCNs corresponding to constant inputs.

DEFINITION 1. *Given a BCN*

$$\mathbf{x}(t+1) = L \ltimes \tilde{\mathbf{u}}(t) \ltimes \mathbf{x}(t), \tag{7}$$

$$\mathbf{y}(t) = H \ltimes \tilde{\mathbf{u}}(t) \ltimes \mathbf{x}(t), \tag{8}$$

with $\mathbf{x}(t) \in \Delta_N, \tilde{\mathbf{u}}(t) \in \Delta_M$ and $\mathbf{y}(t) \in \Delta_P$, we say that $\mathbf{x}_e \in \Delta_N$ is an equilibrium point *of the BCN corresponding to the constant input $\bar{\mathbf{u}}$, if*

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_e \\ \tilde{\mathbf{u}}(t) = \bar{\mathbf{u}}, \forall\, t \in \mathbb{Z}_+ \end{cases} \quad \Rightarrow \quad \mathbf{x}(t) = \mathbf{x}_e, \forall\, t \in \mathbb{Z}_+.$$

$\mathbf{x}_e \in \Delta_N$ *is a* globally attractive equilibrium point *of the BCN corresponding to the constant input $\bar{\mathbf{u}}$, if for every $\mathbf{x}(0) \in \Delta_N$ when applying $\tilde{\mathbf{u}}(t) = \bar{\mathbf{u}}, \forall\, t \in \mathbb{Z}_+$, we obtain that there exists $\tau = \tau(\mathbf{x}(0)) \geq 0$ such that $\mathbf{x}(t) = \mathbf{x}_e, \forall\, t \in \mathbb{Z}_+, t \geq \tau$.*

Clearly, if $\mathbf{x}_e \in \Delta_N$ is an equilibrium point of the BCN (7)-(8) corresponding to the constant input $\bar{\mathbf{u}}$, then the corresponding output takes the constant value $\mathbf{y}_e := H \ltimes \bar{\mathbf{u}} \ltimes \mathbf{x}_e$.

In order to identify the equilibrium points of a BCN corresponding to some constant input $\bar{\mathbf{u}} = \delta_M^k$, it is sufficient to evaluate the equilibria of the Boolean network [Cheng et al. 2011]

$$\mathbf{x}(t+1) = L_k \mathbf{x}(t). \tag{9}$$

Such equilibria are the states $\delta_N^i \in \Delta_N$ such that $\delta_N^i = L_k \delta_N^i$, and hence the states $\delta_N^i \in \Delta_N$ such that $[L_k]_{ii} = 1$. Furthermore, an equilibrium point is globally attractive (assuming that the input remains constant) if and only if all columns of L_k^N coincide with δ_N^i.

In order to identify the equilibrium points of our specific BCN (3)-(6), we easily observe that it is sufficient to first identify the equilibria of the context and then those of the functional model. The analysis of (3) and the expressions of the matrices $\mathbf{C}_k, k \in \{1, 2, 3, 4\}$, immediately reveal that

—for $\bar{\mathbf{u}} = \delta_4^1$ there is a unique equilibrium point $\mathbf{c}_e = \delta_5^5$, and it is globally attractive;

—for $\bar{\mathbf{u}} = \delta_4^i, i \neq 1$, there is a unique equilibrium point $\mathbf{c}_e = \delta_5^1$, and it is globally attractive, in turn.

The constant output value corresponding to the two cases are $\bar{\mathbf{v}}_4 = \delta_2^1$ for $\bar{\mathbf{u}} = \delta_4^1$, and $\bar{\mathbf{v}}_4 = \delta_2^2$ for $\bar{\mathbf{u}} = \delta_4^i, i \neq 1$.

Let us now consider the functional model (4). We have the following two cases:

—for $\bar{\mathbf{v}}_3 = \delta_2^1$ there is a unique equilibrium point $\mathbf{a}_e = \delta_3^3$, and it is globally attractive;

—for $\bar{\mathbf{v}}_3 = \delta_2^2$ there is a unique equilibrium point $\mathbf{a}_e = \delta_3^1$, and it is globally attractive, in turn.

So, to summarize, we have the following situation:

Constant input	Equilibria
$\bar{\mathbf{u}} = \delta_4^1, \bar{\mathbf{v}}_3 = \delta_2^1$	$\mathbf{c}_e = \delta_5^5, \mathbf{a}_e = \delta_3^3$
$\bar{\mathbf{u}} = \delta_4^1, \bar{\mathbf{v}}_3 = \delta_2^2$	$\mathbf{c}_e = \delta_5^5, \mathbf{a}_e = \delta_3^1$
$\bar{\mathbf{u}} = \delta_4^i, i \neq 1, \bar{\mathbf{v}}_3 = \delta_2^1$	$\mathbf{c}_e = \delta_5^1, \mathbf{a}_e = \delta_3^3$
$\bar{\mathbf{u}} = \delta_4^i, i \neq 1, \bar{\mathbf{v}}_3 = \delta_2^2$	$\mathbf{c}_e = \delta_5^1, \mathbf{a}_e = \delta_3^1$

If we now introduce the remaining inputs, and we recall that the input $\bar{\mathbf{v}}_4$ is the output of the context model, we obtain the following results that describe for each constant input the equilibria and the corresponding constant outputs.

Constant input	Equilibria	Constant output
$\bar{\mathbf{v}}_1 = \delta_2^1, \bar{\mathbf{v}}_2 = \delta_2^1,$ $\bar{\mathbf{v}}_3 = \delta_2^1, \bar{\mathbf{u}} = \delta_4^1$	$\mathbf{c}_e = \delta_5^5, \mathbf{a}_e = \delta_3^3$	$\mathbf{m}_e = \delta_3^1$
$\bar{\mathbf{v}}_1 = \delta_2^2, \bar{\mathbf{v}}_2 = \delta_2^2$ $\bar{\mathbf{v}}_3, \bar{\mathbf{u}}$ arbitrary	$(\mathbf{c}_e, \mathbf{a}_e) \in$ $\{(\delta_5^1, \delta_3^1), (\delta_5^1, \delta_3^3), (\delta_5^5, \delta_3^1), (\delta_5^5, \delta_3^3)\}$	$\mathbf{m}_e = \delta_3^3$
all other choices	$(\mathbf{c}_e, \mathbf{a}_e) \in$ $\{(\delta_5^1, \delta_3^1), (\delta_5^1, \delta_3^3), (\delta_5^5, \delta_3^1), (\delta_5^5, \delta_3^3)\}$	$\mathbf{m}_e = \delta_3^2$

This analysis shows that *no limit cycles can appear, and hence no contradicting alarm messages can be delivered by the system.*

7. OBSERVABILITY AND RECONSTRUCTIBILITY

The definitions of observability and reconstructibility are given by slightly adjusting those given in [Fornasini and Valcher 2013a], since in this paper we assume that the input at time t directly affects the update of the output at time t (we consider proper BCNs as opposed to the strictly proper BCNs, typically investigated in the literature). These properties have been the subject of an extensive research, in particular we refer the interested reader to [Laschov et al. 2013; Zhang and Zhang 2016].

DEFINITION 2. *Given a BCN (7)-(8), with* $\mathbf{x}(t) \in \Delta_N, \tilde{\mathbf{u}}(t) \in \Delta_M$ *and* $\mathbf{y}(t) \in \Delta_P$, *we say that the BCN is*

—observable *if there exists* $T \in \mathbb{Z}_+$ *such that the knowledge of the input and output vectors in the discrete interval* $\{0, 1, \dots, T\}$ *allows to uniquely determine the initial state* $\mathbf{x}(0)$;
—reconstructible *if there exists* $T \in \mathbb{Z}_+$ *such that the knowledge of the input and output vectors in the discrete interval* $\{0, 1, \dots, T\}$ *allows to uniquely determine the final state* $\mathbf{x}(T)$.

The hydrogeological model proposed in this paper is not observable. Indeed, it is easily seen that the first BCN:

$$\begin{aligned} \mathbf{c}(t+1) &= \mathbf{C} \ltimes \mathbf{u}(t) \ltimes \mathbf{c}(\mathbf{t}) \\ \mathbf{v}_4(t) &= \mathbf{H_c} \ltimes \mathbf{c}(t) \end{aligned} \qquad (10)$$

is not observable, since initial states as $\mathbf{c}(0) = \delta_5^1$ and $\mathbf{c}(0) = \delta_5^2$ corresponding to the constant input $\tilde{\mathbf{u}}(t) = \delta_4^4, t \in \mathbb{Z}_+$, generate the same output sequence $\mathbf{v}_4(t) = \delta_2^2, t \in \mathbb{Z}_+$. This clearly prevents the whole interconnected BCN representing the hydrogeological system to be observable. It is worth remarking, however, that the system is observable in a weak sense (see [Laschov et al. 2013; Zhang and Zhang 2016]), since for every pair of initial states there exists a specific choice of the input sequence that would generate two distinct output trajectories from which the initial states could be recognised.

Lack of observability is not a major issue. In particular, observability does not seem to be a fundamental system property for the hydrogeological model, since identifying the initial state of the system during some observation interval does not bring any practical advantage. On the other hand, reconstructibility is a more relevant property to investigate: by identifying the current system state, say $\mathbf{x}(T)$, from the observation of the input and the output in some time interval $[0, T]$, *one may anticipate whether an alert signal will lead to an alarm signal at the next time instant or not and hence be ready to run away or to provide support.*

PROPOSITION 1. *The hydrogeological system described by (3)-(4)-(5)-(6) is reconstructible and the definition of reconstructibility holds for $T = 4$.*

PROOF. We exploit the fact that the overall system is represented as a cascade of two BCNs and first investigate the possibility of identifying the context state from the available information. We note that every time the input $\mathbf{u}(t)$ is equal to δ_4^1 the counter variable $\mathbf{c}$ increases when moving from t to $t+1$, unless it has already reached the maximum value in which case it remains constant to the value $\mathbf{c}(t+1) = \mathbf{c}(t) = \delta_5^5$. Conversely, if $\mathbf{u}(t) \neq \delta_4^1$ then, independently of $\mathbf{c}(t)$, the counter state $\mathbf{c}(t + 1)$ takes the value δ_5^1. This implies that if we know the input sequence $\mathbf{u}(t), t \in \mathbb{Z}_+$, (even if we ignore the output) then at latest at $T_1 = 4$ (a situation that occurs only if $\mathbf{c}(0) = \delta_5^1$ and $\mathbf{u}(t) = \delta_4^1$ for $t = 0, 1, 2, 3$) we are able to identify exactly $\mathbf{c}(T_1)$. Therefore the state of the context is always reconstructible (indeed, based on the input sequence alone). If we identify $\mathbf{c}(T_1)$ then, knowing $\mathbf{u}(t)$ for $t \geq T_1$, we are able to determine $\mathbf{c}(t)$ and $\mathbf{v}_4(t)$ for $t \geq T_1$.
At the same time, we now know $\mathbf{v}(t), t \geq T_1$, which is the input of the Functional System model:

$$
\begin{aligned}
\mathbf{a}(t+1) &= \mathbf{A} \ltimes \mathbf{v_3(t)} \ltimes \mathbf{a(t)} \\
\mathbf{m}(t) &= \mathbf{M} \ltimes \mathbf{v}(t) \ltimes \mathbf{a}(t).
\end{aligned}
\tag{11}
$$

So, if we prove the reconstructibility of this second system, we have proved the reconstructibility of the overall hydrogeological system. Reconstructibility of (11) is easily proved along the same lines as for the first BCN. Indeed, every time the input $\mathbf{v}_3(t)$ is equal to δ_2^1 the counter variable $\mathbf{a}$ increases when moving from t to $t + 1$, unless it has already reached the maximum value in which case it remains constant to the value $\mathbf{a}(t + 1) = \mathbf{a}(t) = \delta_3^3$. Conversely, if $\mathbf{v}_3(t) \neq \delta_2^1$ then, independently of $\mathbf{a}(t)$, $\mathbf{a}(t + 1)$ takes the value δ_3^1. This means that if we know the input sequence $\mathbf{v}_3(t), t \in \mathbb{Z}_+$, then at latest at $T_2 = 2$ we are able to identify exactly $\mathbf{a}(T_2)$ (and hence $\mathbf{a}(t), t \geq T_2$).
Putting together the two parts of the reasoning, we can claim that from $T = \max\{T_1, T_2\} = 4$ onward, we are able to identify both $\mathbf{c}(t)$ and $\mathbf{a}(t)$. This proves that the system is reconstructible and the definition holds for $T = 4$. □

8. FAULT DETECTION

A general theory of fault detection in the context of BCNs is still at an early stage, nonetheless there have been some interesting contributions addressing this important problem [Fornasini and Valcher 2015a; 2015b; Sutavani et al. 2019; Zhang et al. 2018]. The fault detection problem investigated in [Fornasini and Valcher 2015a; 2015b; Zhang et al. 2018] refers to the case when the matrices L and H involved in the state and output equations (7) and (8) are replaced by two different (and potentially arbitrary) logical matrices L_F and H_F, as a consequence of a fault. In the context of the hydrological system (and of any alarm system for which the alarm is launched only when some variable take critical values on a sufficiently long time interval), the state variables represent counter variables, and the state-update equations are extremely elementary. So, the case of a fault that arbitrarily affects the matrices that regulate the counter variables update does not seem a very realistic one. Similarly, the case when the matrix that generates the alarm/alert output signal is replaced by a different logic matrix seems too general and not representative of the real faults that may affect the system.

An exception is represented by the case when the change of the matrices L or H formalizes a very classical type of fault that has been investigated for logic circuits: the so called *stuck-in fault*. In the context of the hydrogeological system, this corresponds to the case when one (or more) state variable is stuck at some constant value, independently of the soliciting input. In other words, we are considering the case when one of the counters for some reasons does not update (the case when both counters get stuck at the same time is extremely unrealistic).

Assume that there exists some time instant $\tau \in \mathbb{Z}_+$ such that either $\mathbf{c}(t) = \mathbf{c}(\tau)$ for every $t \geq \tau$ or $\mathbf{a}(t) = \mathbf{a}(\tau)$ for every $t \geq \tau$. The problem that we want to investigate is the following

one: Assuming that we know the hydrogeological system model (3)-(4)-(5)-(6), and that we have access both to the inputs $\mathbf{u}(t), \mathbf{v}_1(t), \mathbf{v}_2(t), \mathbf{v}_3(t)$ and to the output $\mathbf{y}(t) = \mathbf{m}(t)$, are we able to detect such a fault? If so, are we able to identify which of the two counters is blocked?

The observability and reconstructibility analysis carried on in the previous section allows to provide a quite complete answer to both questions. It is in fact easy to see that if the stuck-in fault affects the context state variable at $t = \tau$ and blocks the context state at the value $\mathbf{c}(t) = \delta_5^i, i \neq 5$, for every $t \geq \tau$, then every input signal $\mathbf{u}(t), t \in \mathbb{Z}_+$, that starting from $t = \tau$ does not take the value δ_2^1 for more than 3 consecutive time instants, will generate the constant context output signal $\mathbf{v}_4(t) = \delta_2^2$ (context-alert= quiet), exactly as it would if the context state would be correctly working starting from that same value $\mathbf{c}(t) = \delta_5^i$ at $t = \tau$ but correctly evolving in time. This is due to the fact that the system is not observable and hence different state trajectories are compatible with the same input-output trajectories; in particular, there exist constant state trajectories that cannot be distinguished from time-varying state trajectories. A quite similar analysis could be carried on for the Functional System state variable $\mathbf{a}(t)$, since the case when $\mathbf{a}$ is stuck at δ_3^1 or at δ_3^2 cannot be detected from the output signal $\mathbf{m}(t)$, when $\mathbf{v}_3$ is identically equal to δ_2^2. This allows to say that, in general, a stuck-in fault may not be detected and hence, a fortiori, identified.

This is surely not a good system feature. However, it must be remarked that the situations we have depicted are those when an alarm signal would have not been generated even if the system state would have not been stuck at a constant value. Indeed if the input signals $\mathbf{u}(t), \mathbf{v}_1(t), \mathbf{v}_2(t), \mathbf{v}_3(t)$ simultaneously take the value δ_2^1 on a sufficiently large time window, then stuck-in faults that would erroneously lead to a non-alarm signal could be easily identified and hence corrected.

Indeed, the simple knowledge of $\mathbf{u}(t), t \in \mathbb{Z}_+$, allows to identify $\mathbf{c}(t)$ for $t \in \mathbb{Z}_+, t \geq T_1 = 4$. Similarly, the knowledge of $\mathbf{v}_3(t), t \in \mathbb{Z}_+$, allows to identify $\mathbf{a}(t)$ for $t \in \mathbb{Z}_+, t \geq T_2 = 2$. This means that we can obtain an estimate $\hat{\mathbf{c}}(t)$ of $\mathbf{c}(t)$ and an estimate $\hat{\mathbf{a}}(t)$ of $\mathbf{a}(t)$, and these estimates are both exact from $T = T_1 = 4$ onward, provided that the system is not affected by faults. By making use of these estimates and the system model, we can derive the estimate of the context output

$$\hat{\mathbf{v}}_4(t) = \mathbf{H_c} \ltimes \hat{\mathbf{c}}(t) \tag{12}$$

and of the overall system output

$$\hat{\mathbf{m}}(t) = \mathbf{M} \ltimes \hat{\mathbf{v}}(t) \ltimes \hat{\mathbf{a}}(t), \tag{13}$$

where $\hat{\mathbf{v}}(t) := \mathbf{v}_1(t) \ltimes \mathbf{v}_2(t) \ltimes \mathbf{v}_3(t) \ltimes \hat{\mathbf{v}}_4(t)$.
So, if the inputs are all equal to δ_2^1 for at least 4 consecutive time instants, we know that an alarm signal should be generated and if this is not the case then a fault has necessarily occurred.

To conclude, we have proved what follows for the hydrogeological model:

PROPOSITION 2. *Given the hydrogeological system described by (3)-(4)-(5)-(6), a stuck-in fault for one of the state variables, $\mathbf{c}(t)$ or $\mathbf{a}(t)$, cannot be identified corresponding to all the input sequences, but if one of the counters gets stuck at a value that is not maximum, thus preventing the possible generation of an alarm, then the previous state estimator always allows to detect and identify the stuck-in fault at latest after $T = 4$ times instants after the fault has occurred.*

Note, finally, that a false alarm cannot possibly be issued, because this would require not only that one of the counters is stuck to the maximum value but also that the other is at the maximum value in turn and the inputs are all high, but *this is the case when the alarm message should be issued!*

9. CONCLUSIONS AND FUTURE WORK

In this paper we model a simple Context-aware system as a Boolean Control Network in order to use the powerful tools typical of system theory, which apply to linear analog systems, also to digital systems, whose properties are usually expressed by logical rules. The ultimate goal is to pave the way to formally assess reliability and safety properties of self adapting safety critical systems.

The existence of globally attractive equilibrium points under constant input and the reconstructibility of the system have been proved, as well as the possibility of identifying some faults which could adversely affect the system output.

Further work is to be made to apply these techniques to more complex feedback systems, where the output of the functional systems can affect in turn the state of the Context, and to enhance fault tolerance by considering possible correlations among the sensors and other system input/output devices.

REFERENCES

Bolchini, C., Curino, C., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F. A., and Tanca, L. 2009. And what can context do for data? *Commun. ACM 52,* 11, 136–140.

Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F. A., and Tanca, L. 2007. A data-oriented survey of context models. *ACM SIGMOD Record 36, n.4*, 19–26.

Cheng, D. and Qi, H. 2009. Controllability and observability of Boolean Control Networks. *Automatica 45,* 7, 1659 − 1667.

Cheng, D. and Qi, H. 2010a. A linear representation of dynamics of Boolean Networks. *IEEE Transactions on Automatic Control 55,* 10 (Oct), 2251–2258.

Cheng, D. and Qi, H. 2010b. State space analysis of Boolean Networks. *IEEE Transactions on Neural Networks 21,* 4 (April), 584–594.

Cheng, D., Qi, H., and Li, Z. 2011. *Analysis and control of Boolean networks.* Springer-Verlag, London.

Cheng, D., Qi, H., Li, Z., and Liu, J. B. 2011. Stability and stabilization of Boolean networks. *Int. J. Robust Nonlin. Contr. 21*, 134–156.

Cherfia, T. A., Belala, F., and Barkaoui, K. 2014. Towards formal modeling and verification of context-aware systems. In *VECoS.*

Dey, A. K. 2001. Understanding and using context. *Personal Ubiquitous Computing 5,* 1 (January), 4–7.

Diao, Y., Hellerstein, J. L., Parekh, S., Griffith, R., Kaiser, G., and Phung, D. 2005. Self-managing systems: A control theory foundation. In *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*. 441–448.

Djoudi, B., Bouanaka, C., and Zeghib, N. 2016. A formal framework for context-aware systems specification and verification. *J. Syst. Softw. 122,* C (Dec.), 445–462.

Fornasini, E. and Valcher, M. E. 2013a. Observability, reconstructibility and state observers of Boolean control networks. *IEEE Tran. Aut. Contr. 58 (6)*, 1390 − 1401.

Fornasini, E. and Valcher, M. E. 2013b. On the periodic trajectories of Boolean Control Networks. *Automatica 49*, 1506–1509.

Fornasini, E. and Valcher, M. E. 2015a. Fault detection analysis of Boolean control networks. *IEEE Trans. Automatic Control*, DOI: 10.1109/TAC.2015.2396646.

Fornasini, E. and Valcher, M. E. 2015b. Fault detection problems for Boolean networks and Boolean control networks. In *Proc. of the 34th Chinese Control Conference*. Hangzhou, China, 1–8.

Fornasini, E. and Valcher, M. E. 2016. Recent developments in Boolean Control Networks. *Journal of Control and Decision 3,* 1, 1–18.

Laschov, D., Margaliot, M., and Even, G. 2013. Observability of Boolean networks: A graph-theoretic approach. *Automatica 49 (8)*, 2351–2362.

Nzekwa, R., Rouvoy, R., and Seinturier, L. 2010. A flexible context stabilization approach for self-adaptive application. In *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 7–12.

Padovitz, A., Zaslavsky, A. B., Loke, S. W., and Burg, B. 2004. Stability in context-aware pervasive systems: A state-space modeling approach. INSTICC, Ed. 129–138.

Padovitz, A., Zaslavsky, A. B., Loke, S. W., and Burg, B. 2005. Maintaining continuous dependability in sensor-based context-aware pervasive computing systems. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 290a–290a.

Schreiber, F. A. and Panigati, E. 2017. Context-aware self adapting systems: a ground for the cooperation of data, software, and services. *IJNGC 8,* 1.

Sutavani, S., Sonam, K., Wagh, S., and Singh, N. 2019. Disturbance decoupling and instantaneous fault detection in Boolean Control Networks.

Zhang, K. and Zhang, L. 2016. Observability of Boolean Control Networks: A unified approach based on Finite Automata. *EEE Trans. Automatic Control 61 (9)*, 2733–2738.

Zhang, Z., Leifeld, T., and Zhang, P. 2018. Active fault detection of Boolean Control Networks. In *Proc. of the American Control Conference (ACC) 2018*. Milwaukee, WI, 5001–5006.

Prof. Ing. Fabio A. Schreiber is Professor Emeritus (retired since November 2015) of Pervasive Data Management and of Database Systems at Politecnico di Milano. From 1981 to 1986 he was full professor of Computer Science at the Mathematical Department of the Universit di Parma. He received the Dr. Ing. degree in Electronic Engineering from Politecnico di Milano in 1969. His main research interests are in the fields of Distributed Informatics and Information Systems. His current research topics include: Database Systems for context-aware applications, Very Small and Mobile Database design methodologies and applications; Pervasive Information Systems. Other research interests include: dependability evaluation of computing and information systems and distributed information systems design. On these and other topics he authored more than hundred papers published in international journals and conferences. He organized the "3rd International Seminar on Distributed Data Sharing Systems" (Parma, 1984), is member of the editorial boards of Data and Knowledge Engineering, and, from 1975 to 1993, he has been Editor-in Chief of Rivista di Informatica - the journal of the Italian Association for Informatics (AICA). He participated to several research projects funded by Italian research institutions and by the ESPRIT and HORIZON 2020 European actions. He also partecipated to several projects on Information Systems and Telematics for Public Administrations and Local Authorities; for this activity he was bestowed the title of "Commendatore" of the Order "Al Merito della Repubblica Italiana". He is member of AICA, Senior member of ACM, and Life Senior member of IEEE.

Prof. Maria Elena Valcher received the Master degree (1991) and the PhD degree (1995) from the University of Padova, Italy. Since January 2005 she is full professor at the University of Padova. She is author/co-author of 85 papers appeared in international journals, 100 conference papers, 2 text-books and several book chapters. Her research interests include cooperative control and consensus, positive switched systems and Boolean control networks. She was in the Editorial Board of the IEEE Transactions on Automatic Control (1999-2002), Systems and Control Letters (2004-2010) and she is currently in the Editorial Boards of Automatica (2006-today), Multidimensional Systems and Signal Processing (2004-today), SIAM J. on Control and Optimization (2012-today), European Journal of Control (2103-today) and IEEE Access (2014-today). She was Vice President Member Activities of the CSS (2006-2007); Vice President Conference Activities of the CSS (2008-2010); CSS President (2015). She received the 2011 IEEE CSS Distinguished Member Award and she is an IEEE Fellow since 2012. Since January 2017 she is the founding Editor in Chief of the IEEE Control Systems Letters.