Formal Verification of Context Aware Systems *

Fabio A. Schreiber[1][0000-0002-5237-0455] and Maria Elena Valcher[2][0000-0002-7595-3688]

[1] DEIB - Politecnico di Milano {fabio.schreiber@polimi.it}
[2] DEI - University of Padova {meme@dei.unipd.it}

Discussion Paper

Abstract. This discussion paper introduces a modeling technique, based on Boolean Control Networks, for assessing useful properties of Context Aware systems. Indeed, 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.

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

1 Introduction and Related Works

Context-aware (C-A) computing was born out of the need to master, by using a component based approach, the growing complexity of modern software systems and enforcing the separation of concerns [9]. Among the most widely used definitions of Context, and of C-A Computing, those proposed by A. Dey [7] 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 C-A applications that: (i) tailor the set of application-relevant data, (ii) increase the precision of information retrieval, (iii) discover services, (iv) build smart environments, and others [2].

In the domains of Databases and Programming Languages, the design of C-A and Self-adapting systems has been frequently based on the separation between Context and functional systems [1, 15]. The introduction of Context-awareness and Self-adaptation in safety-critical applications led to the need for specifying and assessing their properties, mainly those related to the system dependability, by means of formal methods such as Bigraphs and model-checking [6, 9].

In [13, 14] 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. The likelihood of the transition is evaluated by assuming notions analogous to those of velocity and acceleration in

* Fabio A. Schreiber was partially supported by INAIL, RECKON Project..

mechanical systems.
Stability is a traditional topic in control systems theory, and in [8] 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 C-A 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, and solutions to problems such as controllability, observability, stability and reconstructibility have been proposed [3, 10, 12].

In Section 2 the case study is outlined. In Section 3, we show how, by making use of the algebraic approach to Boolean Control Networks, it is possible to formalize, by means of a C-A system, a decision process to avoid hydrogeological disasters, as the one that happened in Abruzzo, where a hotel was hit by an avalanche [16], as well as false alarms. We can then assess the existence of globally attractive equilibrium points of the overall system, corresponding to constant inputs, and investigate some interesting structural properties such as observability and reconstructibility that formalize system features of great practical relevance. Moreover, the possibility of identifying some kinds of faults in the inputs, that could result in errors in the alarm system, is examined. Sections 4 and 5 respectively describe the results of the analysis and possible future developments.

To conclude, we think that cooperation between C-A computing and Control Systems theory can be fruitful to fill in the model gap.

2 The architecture of the monitoring system

Figure 1 shows the architecture of a C-A system [2, 7] 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. The *integration* of data from 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 the one 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 resulting 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. 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 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.
- To use BCNs and System Theory tools to asses properties of a C-A system such as: the existence of globally stable equilibrium points and the absence of oscillatory behaviors (limit cycles) under constant inputs; the reconstructibility of the system and the detection of some faults affecting the C-A system inputs.

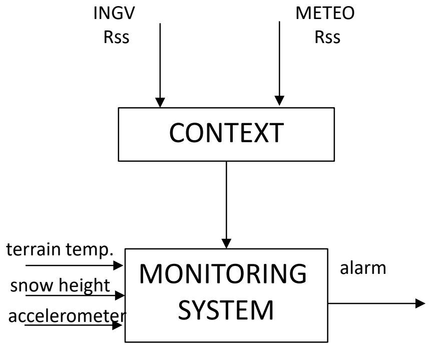

Fig. 1. Open loop Context Aware system

Although the following analysis refers to the case study, the proposed approach is quite general and can be applied whenever properties should be proved in C-A dynamic systems.

3 The BCN Model of the Hydrogeological Example

In [16] a short introduction to the algebraic representation of Boolean Control Networks can be found; in the following we introduce the BCN model of our example case.

3.1 The Context model

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

where δ_n^k is the canonical vector of size n whose kth entry is unitary.

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 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 $\ltimes$ denotes the semi-tensor product, $C = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 \end{bmatrix} \in \mathscr{L}_{5\times 20}$, the set of logic matrices of size 5 × 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 stricter - if increased - or loser - if lowered - requirements on the alarm system.*

Context Output Introduce the Context model output
CONTEXT-ALERT$\{danger, quiet\}$:= U$_c$
Since 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, the variable $\mathbf{u}_c$ is updated following the algebraic rule:
$\mathbf{u}_c(t) = \mathbf{H_c} \ltimes \mathbf{c}(t)$ where
$\mathbf{H_c}$ = $\begin{bmatrix} \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^1 \end{bmatrix} \in \mathscr{L}_{2\times 5}$

Figure 2 shows the state diagram for the Context automaton.

This is one possible solution, but it may be regarded as somewhat dangerous: if the counter is erroneously reset, then the alert ends up being significantly delayed. An alternative solution could be that of simply decreasing the counter by one 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.

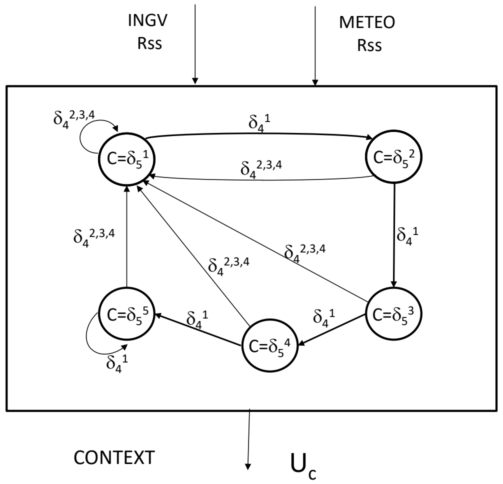

Fig. 2. Context State diagram for the hydrogeological example

3.2 The Functional System model

Functional System Input Variables We assume that, in addition to the CONTEXT-ALERT variable, the Functional System model receives three more input signals from local sensors, so that, at the end, the Input Vector $\mathbf{v}(t) \in \Delta_{16}$ is composed by the following variables determining the system dynamics:

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

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 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. Therefore, 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 $A = \left[A_1\ A_2\right] \in \mathscr{L}_{3\times 6}$, and
$A_1 = A \ltimes \delta_2^1 = [\delta_3^2\ \delta_3^3\ \delta_3^3]$
$A_2 = A \ltimes \delta_2^2 = [\delta_3^1\ \delta_3^1\ \delta_3^1]$

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 $\mathbf{M} = \left[M_1\ M_2\ \dots\ M_{16}\right] \in \mathscr{L}_{6\times 16}$, 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 BCN obtained by connecting the BCN describing the context and the BCN describing the functional model, and hence it is described by the following equations:

$$
\begin{aligned}
\mathbf{c}(t+1) &= \mathbf{C} \ltimes \mathbf{u}(t) \ltimes \mathbf{c}(\mathbf{t}) && (1)\\
\mathbf{a}(t+1) &= \mathbf{A} \ltimes \mathbf{v_3}(\mathbf{t}) \ltimes \mathbf{a}(\mathbf{t}) && (2)\\
\mathbf{v}_4(t) &= \mathbf{H_c} \ltimes \mathbf{c}(t) && (3)\\
\mathbf{m}(t) &= \mathbf{M} \ltimes \mathbf{v}(t) \ltimes \mathbf{a}(t). && (4)
\end{aligned}
$$

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 (1) - (4). This provides further evidence of the convenience of using C-A 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 C-A 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 modelled as the cascade of a Context and a Functional Systems, both of the them affected by external signals and measurements, and whose target is to generate alert/alarm notifications based on the occurrence of specific (possibly repeated) combinations of data.

4 The main results

In [16] we present a detailed description of the kind of reasoning that can be made using the model developed in the previous sections. Here we only mention the main results, referring to the original paper for a full treatment.

- We use definitions and methods as in [4, 5, 11] to find equilibrium points of BCNs corresponding to constant inputs. The analysis shows that *no limit cycles can appear in the system, and hence no contradicting alarm messages can be delivered by it*.
- 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*. The analysis shows that *the hydrogeological system described by* (1)-(2)-(3)-(4) *is reconstructible and the definition of reconstructibility holds for* $T = 4$.
- Detection of *stuck-in faults*. We proved that *Given the hydrogeological system described by* (1)-(2)-(3)-(4)*, 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 from the fault occurrence.* 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!*

5 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.

The main advantages of the proposed approach are on the one hand to provide a solid system theoretic framework where intuitive and practical features or goals for Context-aware systems can be properly formalised, on the other hand to offer rigorous algebraic tools to test these properties and solve those problems. The proposed solutions are easily converted into computer algorithms. We are working 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

1. Bolchini, C., Curino, C., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F.A., Tanca, L.: And what can context do for data? Commun. ACM **52**(11), 136–140 (2009). https://doi.org/10.1145/1592761.1592793, https://doi.org/10.1145/1592761.1592793
2. Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F.A., Tanca, L.: A data-oriented survey of context models. ACM SIGMOD Record **36, n.4**, 19–26 (2007)
3. Cheng, D., Qi, H.: Controllability and observability of Boolean Control Networks. Automatica **45**(7), 1659 – 1667 (2009)
4. Cheng, D., Qi, H., Li, Z.: Analysis and control of Boolean networks. Springer-Verlag, London (2011)
5. Cheng, D., Qi, H., Li, Z., Liu, J.B.: Stability and stabilization of Boolean networks. Int. J. Robust Nonlin. Contr. **21**, 134–156 (2011)
6. Cherfia, T.A., Belala, F., Barkaoui, K.: Towards formal modeling and verification of context-aware systems. In: VECoS (2014)
7. Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**(1), 4–7 (January 2001)
8. Diao, Y., Hellerstein, J.L., Parekh, S., Griffith, R., Kaiser, G., Phung, D.: Self-managing systems: A control theory foundation. In: 12th IEEE - ECBS'05. pp. 441–448 (April 2005)
9. Djoudi, B., Bouanaka, C., Zeghib, N.: A formal framework for context-aware systems specification and verification. J. Syst. Softw. **122**(C), 445–462 (Dec 2016). https://doi.org/10.1016/j.jss.2015.11.035, https://doi.org/10.1016/j.jss.2015.11.035
10. Fornasini, E., Valcher, M.E.: Observability, reconstructibility and state observers of Boolean control networks. IEEE Tran. Aut. Contr. **58 (6)**, 1390 – 1401 (2013)
11. Fornasini, E., Valcher, M.E.: On the periodic trajectories of Boolean Control Networks. Automatica **49**, 1506–1509 (2013)
12. Fornasini, E., Valcher, M.E.: Recent developments in Boolean Control Networks. Journal of Control and Decision **3**(1), 1–18 (2016)
13. Padovitz, A., Zaslavsky, A.B., Loke, S.W., Burg, B.: Stability in context-aware pervasive systems: A state-space modeling approach. In: INSTICC. pp. 129–138 (2004)
14. Padovitz, A., Zaslavsky, A.B., Loke, S.W., Burg, B.: Maintaining continuous dependability in sensor-based context-aware pervasive computing systems. Proceedings of the 38th Hawaii International Conference on System Sciences (2005)
15. Schreiber, F.A., Panigati, E.: Context-aware self adapting systems: a ground for the cooperation of data, software, and services. IJNGC **8**(1) (2017), http://perpetualinnovation.net/ojs/index.php/ijngc/article/view/364
16. Schreiber, F.A., Valcher, M.E.: Formal assessment of some properties of context-aware systems. IJNGC **10**(3), 163–177 (2019), http://arxiv.org/abs/2005.00373