STATE AND TIME GRANULARITY IN SYSTEM DESCRIPTION: AN EXAMPLE*

Fabio A. Schreiber
Dipartimento di Elettronica - Politecnico di Milano,
P. zza L. da Vinci 32, I-20133 Milano, Italy.
e-mail: schreibe@elet.polimi.it

Abstract: *It is shown how a complex system can be described at different levels of granularity, and how time-driven and event-driven descriptions can coexist.*

Keywords: *ATC radar, real-time systems, sequential machines, systems description and specification*

1 Introduction

Real world systems are complex "objects", which can be modeled at very different levels of detail, and indeed we can define a system recursively as: *a system is a composition of interacting component systems.* Therefore, we need a concept of system which allows us to treat it as atomic when we wish to stop decomposing, but also allows us to continue decomposing when we wish to do so. In order to fix the basic concepts in systems description, we are giving in the following some definitions, mostly taken from [4].

Three are the main types of levels we can consider:

- **behavior level**: this is also called *input-output description* or *black-box description.* In fact the system is viewed as a black box and measures done on it are recorded in a chronological order. This requires that a time-base be defined as a subset of $\mathbf{R}$ (continuous time) or of $\mathbf{Z}$ (discrete time). The behavior of the system is described as a set of *trajectories*, which are mappings from subintervals of the time-base to some sets of values representing possible observation results.

- **state structure level**: the systems is described in terms of mechanisms for its internal working. Such a description is sufficient to generate, by iteration over time, a set of trajectories. The tools for such a description are the *state set*, which represents the possible configurations at any time, the *state transition function*, which provides the rules for computing the future state from the current one, and possibly an *output function* to map the internal state set to an observable output set.

- **composite structure level**: the system is described as the connection of many black boxes; therefore it can also be called a *network description.* The black boxes are defined as the *components* with specified input variables and output variables, and a *coupling specification* must be given

*IEEE Real Time Systems Newsletter, Vol. 7, n. 4, 1991

which determines the interconnections of the components and the interfacing of the input with the output variables.

The set of internal states represents the memory of the system, and it is the heart of the modeling of its internal structure. The choice of the state set is not unique and even its dimensionality is not fixed. The single state can be considered as an *instantaneous snapshot of the universe* [2].

The different representations of a system can be related to each other both at the behavioral and at the internal structure level. At the **behavioral** level the basic relation is that of some kind of **equivalence**; at the **structural** level the basic notions are those of **homomorphism** and of **isomorphism**. Therefore it is possible to reduce a structural description of a system to a simpler one, which is homomorphic with the former description.

2 Ontology of states and events: an example

Given the preceding picture of the system modeling tools and alternatives, we are faced with the question of defining the right levels of description for a system, the primitive objects, their attributes, the values set of each attribute, the rules constituting the state transition function, the input stimuli which trigger state transitions.

We can define as a **state** the set of values taken by the attributes (state variables) of the objects which are the primitives at the level of description we are considering.

As an example we can consider the snaphsot of an ATC radar screen as the state of the sky above us, where the spatial coordinates of the light dots representing the airplanes (objects) are the state variables. An equivalent description could be that of considering the distance matrix between any two dots as the unique state variable, as we shall see in the following.

Most authors link the concept of system evolution, i.e. the transition from one state to another, to the concept of **event**. In many cases the considered event is just the flowing of time. Mc Dermott [2] states that every state has a time of occurrence and that states are arranged in totally ordered sets, called *chronicles*, which are complete possible histories of the universe, a concept analogous to what has been called trajectories before.

On the other hand, events can represent complex situations, which, in turn, entail a state transition. It seems therefore that a contradiction emerges from the two views: the first with an *absolutistic* flavor (events are marked with time), the second with a *relativistic* one (events constitute time) [3].

The contradiction can be resolved if we consider the two conceptions - i. e. an event triggers a state change or a state change is an event - at different levels of detail, or if we see states and events as elements of a duality relation with respect to different views of the system.

At a lower level, a relativistic view is taken, where some simple events, like e. g. the coincidence of two marks, constitute time and are taken as primitives, we build some simple systems which count the events, and we call them **clocks**. At a higher level we consider clocks as components in a system and their outputs are (slightly more complex) events which cause state changes, which, in turn, can be taken as (complex) events at yet another level of description. In this way it is possible to distinguish between the local states - as the internal states of each object - and the global states of the whole system. It should be noticed that, by adopting the recursive definition of system, these descriptions can be

AC	CC	E	*rules*
FALSE	FALSE	N	¬AC
TRUE	FALSE	A	AC ∧¬CC
TRUE	TRUE	C	CC
FALSE	TRUE	–	–

Figure 1: States table

stacked up and the global states of a lower level become local states of an object at a higher level.

Going back to the radar example, we can think at the screen snapshot as a macro object, composed of micro objects constituted by the light spots. The state change of the micro world depends on the flow of time, since at every sweep of the radar antenna the airplane positions on the screen change. However, from the ATC controller point of view, we can distinguish at the macro level three different states of the screen: the first is the “normal sky”, the second is an “alert situation”, the third is a “collision scene”. Obviously, under the state changes we can find the flow of time, but a more fruitful representation at the macro level is to consider the state change to be induced by an event, e. g. two airplanes whose distance is below a minimal threshold.

From this example, we can notice that an event can be the result of an evaluation operation on some state variables. Is this fact in contrast with the previous definitions of event?

3 Transforming states into events

Let us look closer at the example. Let us call $S_t \equiv \{x_{ti} y_{ti} z_{ti}\}$ the state of the system at time t, where the state variables $x_i,\ y_i,\ z_i$ are the spatial coordinates of each airplane a_i on the screen. This is the lowest description level of our system, and it can be modelled as a Moore synchronous sequential machine [1], in which the output is the state S_t itself, and a state transition is triggered by a periodical signal every T (the sweep period); let us call it the *MSM machine*.

Let us define now the function $d(a_i, a_j)$

$$d_t(ij) = \sqrt{(x_{tj} - x_{ti})^2 + (y_{tj} - y_{ti})^2 + (z_{tj} - z_{ti})^2}$$

1

the spatial distance between any two airplanes $a_i,\ a_j$, and a particular value of d, $d = DTH > 0$, called the *danger threshold*.

Since we are mainly interested in aircrafts mutual positions, we can assume all the states having the same distance matrix to be equivalent; therefore, we can transform the *MSM* machine into an *internally equivalent* machine MSM', whose states $D_t \equiv \{d_t(ij)\}$ are defined by the isomorphism (but for rotations or translations) induced by d.

Then, the following predicates can be set:

- **AC (Alert Condition)**: $(\exists t' :\ d_{t'}(ij) \leq DTH) \wedge (\forall t, t' :\ t > t' \rightarrow d_{t'}(i,j) \geq d_t(i,j))$
- **CC (Collision Condition)**: $\exists t :\ d_t(i,j) = 0$

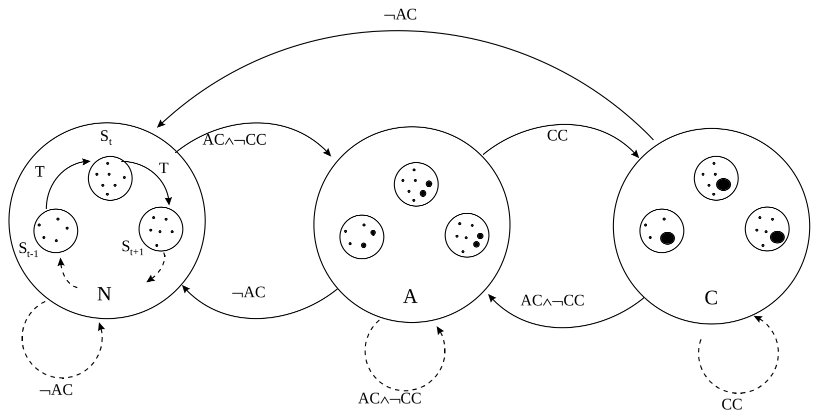

Figure 2: States diagram

Now we group the states of the MSM' machine into three equivalence classes following the rules shown in figure 1, where the equivalence classes are called: N for *Normal*, A for *Alert*, and C for *Collision*.

Tracing back to the physical world, this clustering amounts to *interpreting* the meaning of the rough data by means of conditions AC and CC. At each sweep of the antenna the data are evaluated and one of the rules becomes *TRUE*; the state of the MSM' machine possibly changes, moving among the clusters.

Therefore, at the *behavior level*, the system can be described by a sequential machine *BSM* whose state set is $E \equiv \{N, A, C\}$ and whose state transitions are triggered by a set of *events*, constituted by changes in conditions AC and CC. These are not external inputs to the system, but they come from the evaluation of the boolean function $g(AC(DTH, D), CC(D)) = \neg AC \vee (AC \wedge \neg CC) \vee CC$, which also determines the equivalence classes $\{N, A, C\}$.

Figure 2 gives a global picture of the radar system, where all the description levels are shown.

So far we showed how it is possible to transform a low level description, with a potentially infinite number of states, into a high level description, with only three states. Let us examine now the time structure of the MSM and of the BSM machines; this is important mainly for real-time system description, where synchronous and asynchronous inputs rise rather different problems.

We saw above that MSM is a synchronous machine, driven by a periodic input (i.e. its state change at every sweep), whose output depends only on the internal state: $O_t = f(S_t)$.

On the contrary, BSM is **driven by** *non temporally deterministic* **events**, constituted by predicates AC or CC switching from TRUE to FALSE or viceversa. However we can notice that switching is not totally asynchronous since it can only happen at some multiple n of the sweep period T.

Therefore, also BSM can be described as a synchronous sequential machine, but of the Mealy type, that is whose output is function both of the input I and of the state E: $O_t = f(I_t, E_t)$. In fact, in a Mealy machine the output is produced at the same time as the input (but for internal delays), while in the equivalent Moore machine an input at time t produces an output at time $t+1$ [1]. This feature seems more important for real-time systems than the risk that a noise input to the Mealy machine (which could be filtered ahead of the decision unit) could produce a spurious output.

4 Conclusions

In this paper we showed by a simple, but realistic example, which can be extended to other application domains such as nuclear power plants or the game of chess, that the description tools for complex systems must be adapted to the desired level of granularity, and that the different levels can be formally related to each other.

In an ATC radar system, a change in the state value at the lower level, i.e. airplanes position, results in an event which, in turn, possibly triggers a state change at the upper level, i.e. the meaning of the radar snapshot. This shows that not only the duality between events and state change can be fruitful for modeling complex systems, but also that the concept of state evaluation is consistent with such a model. Figure 3 synthetically shows the main steps to the description transformation.

5 References

[1] G.B. Gerace – La logica dei sistemi di elaborazione – Editori Riuniti, Roma 1987

[2] D. McDermott – A Temporal Logic for Reasoning about Processes and Plans – Cognitive Science, Vol. 6, 1982, p. 101-155

[3] F.A. Schreiber – A Breviary to Time Concepts for Information Systems – Rivista di Informatica, Vol. XX, n. 1, 1990, pp. 59-68

[4] J.A. Spriet, G.C. Vansteenkiste – Computer Aided Modeling and Simulation – Academic Press, London 1982.

$$BSM$$
$$I \equiv \{AC, CC\};\;\; E \equiv \{N, A, C\};$$
$$O_t = f(I_t, E_t)$$

behavior equivalence

$$\Downarrow$$

$$map\ into\ equiv.\ class$$
$$\{D, DTH, AC, CC\} \xrightarrow{g} \{N, A, C, \}$$

$$\Downarrow$$

$$MSM'$$
$$I \equiv \{T_i\};\;\; D \equiv \{d_{ij}\};$$
$$O_t = f(D_t)$$

state isomorphism

$$\Downarrow$$

$$transf.\ state\ variab.$$
$$S \xrightarrow{d} D$$

$$\Downarrow$$

$$MSM$$
$$I \equiv \{T_i\};\;\; S \equiv \{x_i, y_i, z_i\};$$
$$O_t = f(S_t)$$

Figure 3: Transformation between the descriptions