Some Remarks on Explanation of Data and Specification of Processes

Zdzisław Pawlak
Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Baltycka 5, 44 000 Gliwice, Poland
e-mail: zpw@ii.pw.edu.pl

1 Introduction

The aim of this note, which is an extenssion of the article [4], is to give few remarks on some properties of *discrete dynamic systems* (*DDS*). By a *DDS* we will mean a finite set $A = \{a_1, a_2 \dots a_n\}$ of elements called *discrete elements*. We assume that every discrete element can assume one of the its possible *internal states* from the set V_a. Besides, suppose that all the discrete elements change their states according to some fixed rules. We will briefly discuss the following two problems.

Suppose that the system is observed by an observer, who does not know the rules governing behavior of the system and he wants to derive them from the observed data. This process will be called an *explanation of data* (*system analysis*). Two approaches are here possible, the *closed word assumption* (*CWA*) and the *open world assumption* (*OWA*). In the case of *CWA* we assume that the observation contains all possible system behaviors, which characterize uniquely the intrinsic system mechanism. In other words the observation contains the whole knowledge about behavior of the system. In the case of OWA we assume that partial observation of the system only is available, i.e. the observer does not have the whole knowledge about the system behavior. We will deal here only with closed system.

The second problem, called *system specification* (*system synthesis*), consists in giving an algorithm which implements, in hardware or software, the system according to given specification of system behavior.

We will illustrate these ideas by an intuitive example of distributed traffic signals control. Let us consider a very simple intersection (T-intersection) shown in Fig. 1.

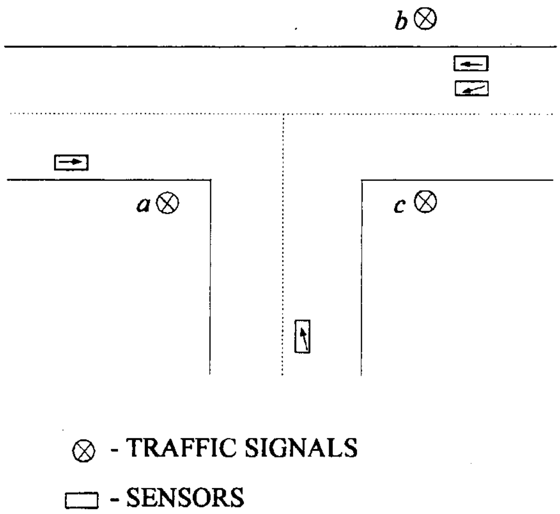

Fig. 1

We assume that a distributed control algorithm supervise the traffic on the basis of local conditions and that the conditions are determined by sensors placed in lanes and indicating the desired turn of a car approaching the intersection. For the sake of simplicity we omit many important factors, needed in a real-life control, e.g., traffic intensity, the busiest directions, length of the green period in each direction etc. Distributed control means that the control cycle is not fixed but it depends on the situation on the intersection

Suppose that the observer has recorded his observations as shown in Table 1.

State	a	b	c
1	1	1	0
2	0	2	0
3	0	0	2

Table 1

Columns of the table are labeled by a, b and c, denoting the traffic signals, rows are marked by numbers 1, 2 and 3 denoting the possible states of the observed system, whereas entries of the table 0, 1 and 2 denote colors of the traffic lights, red, green and green arrow, respectively.

2 Explanation

Using methods offered by set theory [1,5,7,13], not presented here, we get from the table the following control rules

$$
\begin{aligned}
b_0 \vee b_2 &\rightarrow a_0 \\
b_1 &\rightarrow a_1 \\
c_2 &\rightarrow b_0 \\
a_1 &\rightarrow b_1 \\
a_0 c_0 &\rightarrow b_2 \\
b_1 \vee b_2 &\rightarrow c_0 \\
b_0 &\rightarrow c_2
\end{aligned}
$$

which explain behaviour of the system (see also [4]).

Thus explanation of observed phenomena (processes, etc.) consists in deriving decision rules from the observed data.

3 Specification

Suppose now that Table 1 is not a result of observation but represents a set of requirements, which specifies the control algorithm. In this case we can also derive the same set of decision (control) rules from the table (specification) and used them as a basis for control algorithm design, which can be implemented in software or hardware. In the second case we can obtain from the decision rules a controller (switching circuits) presented in Fig. 2.

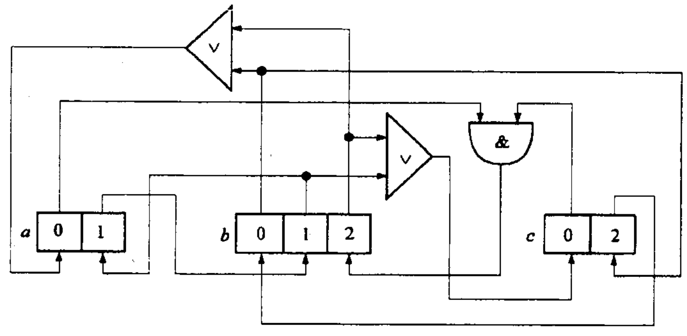

Fig. 2

We use here the standard notation for switching circuits. Particularly, the discrete elements, corresponding to a, b and c are bi-stable or three-stable devices (flip-flops) represented by squares. The complete controller is presented in Fig. 3.

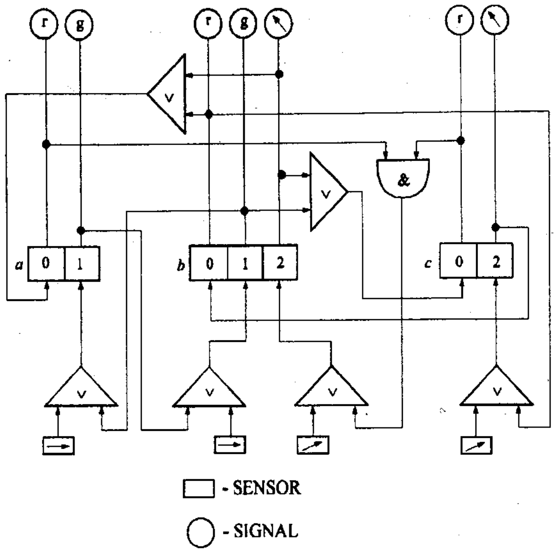

Fig. 3

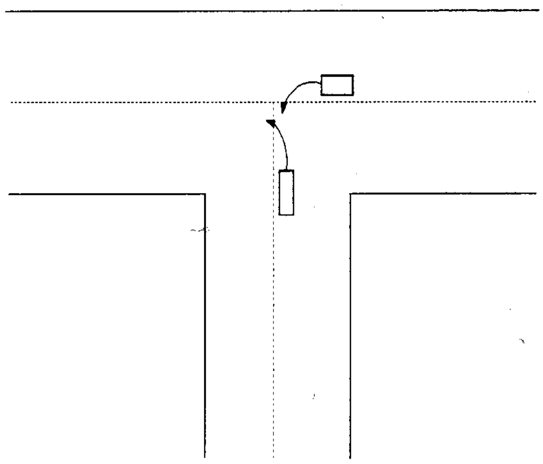

Fig. 4

i.e., both want to turn left. This causes conflict, since $c_2 \longrightarrow b_0$ and $b_2 \longrightarrow c_0$ (states c_2 and b_2 do not occur simultaneously in the specification table) Hence special means to resolve this kind of conflicts must be incorporated in the method, but we will not discuss this issue here.

4 Conclusions

Formally, both explanation (of data) and specification (of processes) are seemingly similar problems. They consists in deriving decision rules from data, representing observation or specification of certain processes. The obtained decision algorithm (set of decision rules) can be used to explain the system behavior or can form a basis for controller design.

It seems the proposed idea can be used as basis for a certain class of control system design. Besides, after some extension it could be also used for software specification [2].

In general, the proposed approach can be also seen as method for concurrent system specification, similar to that offered by Petri Nets [3,6,8,9,10, 11,12].

References

[1] Bazan, J., Skowron, A., and Synak, P., (1995), "Discovery of decision rules from experimental data", in: T.Y. Lin and A.M. Wildberger (eds.), *The Third International Workshop on Rough Sets and Soft Computing Proceedings* (*RSSC'94*), San Jose State University, San Jose, California, USA, November 10–12, 276–279.

[2] Hurley, R.B. (1983). "Decision Tables in Software Engineering" Van Nostrad Reinhold Company, New York.

[3] Pawlak, Z., (1992), "Concurrent versus sequential the rough set perspective", *Bulletin of the EATCS*, Vol. 48, 178-190.

[4] Pawlak, Z., (1996). "Data versus logic a rough set view", in: S. Tsumoto, S. Kobayashi, T. Yokomori, H. Tanaka and A. Nakamura (eds). *The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, Proceedings*, November 6-8, 1996, The University of Tokyo, 1–8.

[5] Pawlak, Z. and Skowron, A. (1993). "A rough set approach for decision rules generation". it Institute of Computer Science Report 23/93, Warsaw University of Technology, Poland. Also in: *Proc. of the IJCAI'93 Workshop: The Management of Uncertainty in AI*, France.

[6] Skowron, A., (1995), "Extracting laws from decision tables", *Computational Intelligence*, 11/2, 371–388.

[7] Skowron, A. and Suraj, Z. (1993). "Rough sets and concurrency". *Bulletin of the PAS*, 41-3, 237–254.

[8] Skowron, A. and Suraj, Z. (1994). "Synthesis of concurrent system specified by information systems". *Institute of Computer Science Report 39/94*, Warsaw University of Technology.

[9] Skowron, A. and Suraj, Z. (1995). "Discovery of concurrent data models form experimental tables". A Rough Set Approach in: Usama M. Fayyad and Ramasamy Uthurusamy (eds.), *Proc. of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, August 19-21, Montreal, 1995, Canada, The AAAI Press, Menlo Park, CA, 288–293.

[10] Skowron, A. and Suraj, Z. (1996). "Parallel algorithm for real-time decision making: A Rough Set Approach. *Journal of Intelligent Information Systems*, 7, Kluwer Academic Publishers, Dordrech, 15–28.

[11] Suraj, Z. (1995). "PN-tools: Environment for the design and analysis of Petri nets". *Control and Cybernetics*, 24-2, Systems Research Institute of Polish Academy of Science, 199–222.

[12] Suraj, Z. (1996). "An application of rough set methods to cooperative informations systems reengineering", in: S. Tsumoto, S. Kobayashi, T. Yokomori, H. Tanaka and A. Nakamura (eds). *The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, Proceedings*, November 6-8, 1996, The University of Tokyo, 364–371.

[13] Ziarko, W., (1991), "The discovery, analysis and representation of data dependencies in databases", in: G. Piatetsky-Shapiro, and W. J. Frawley (eds.),*Knowledge Discovery in Databases*, AAAI Press/MIT Press, 177–195.