Sensor and Filter Models with Rough Petri Nets

J.F. Peters[1], A. Skowron[2], Z. Suraj[3], S. Ramanna[1]

[1] Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 5V6 Canada
[2]Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
[3]Institute of Mathematics, Pedagogical University, Rejtana 16A, 35-310 Rzeszów, Poland

Abstract. This paper considers models of sensors and filters with Petri nets defined in the context of rough sets. Sensors are fundamental computational units in the design of decision systems and neural computing systems. The intent of this work is to construct Petri nets to simulate conditional computation in neural computing systems, which are dependent on filtered input from sensors. In particular, there is interest in bringing to light the computational features of the beginning layer of what are known as rough membership function neural networks. In this paper, coloured Petri nets underlie the definition of a family of Petri nets based on rough set theory. Sensors are modeled relative to what are known as receptor processes in rough Petri nets. Filters are modeled as Lukasiewicz guards on some transitions in rough Petri nets. A Lukasiewicz guard is defined in the context of multivalued logic. The contribution of this paper is the modeling of sensors and filters in the context of receptor processes and Lukasiewicz guards, respectively.

Keywords: approximation, enabling, filter, guard, multivalued logic, neuron, Petri net, rough sets, sensor.

1 Introduction

Considerable work has already been carried out in modeling various forms of systems with Petri nets in Skowron and Suraj [1]-[5], Suraj [6], [31], Peters [7], Peters et al. [8]-[14], and in de Lope, Maravall, and Zato [32]. The aim of the earlier as well as the current research has been to provide a complete framework for approximate reasoning, especially in the context of rough set theory from Pawlak [15]-[20]. Rough set theory also provides an inductive approach to reasoning about data. This paper returns to the idea of rough Petri nets in Peters et al. [11]-[12]. Guarded transitions based on multivalued logic and rough sets are used to design Petri net models of sensor filters, which are fundamental in the design of rough neural computing systems. Dill receptor processes are used to define input places in sensor-driven systems. Lukasiewicz guards are introduced to provide conditional firing to a degree of one or more transitions in Petri net models of dynamical systems. The contribution of this paper is the modeling of receptor processes and Lukasiewicz guards in the context of sensors and filters, respectively.

This paper is structured as follows. A brief presentation of the basic concepts underlying the design of Petri net models of sensors and filters is given in Section 2. Included in Section 2 are a brief look at rough set theory, rough membership functions, rough Petri nets, receptor processes, and guarded transitions. The basic definitions and models of sensors and filters are given in Section 3.

2 Basic Concepts

2.1 Rough Sets

Rough set theory offers a systematic approach to set approximation [15]-[20], [24]. To begin, let S = (U, A) be an information system where U is a non-empty finite set of objects and A is a non-empty finite set of attributes where $a: U \rightarrow V_a$ for every a ∈ A. For each B ⊆ A, there is associated an equivalence relation $\mathrm{Ind}_A(B)$ such that

$$\mathrm{Ind}_A(B) = \{(x, x') \mid \forall a \in B.a(x) = a(x')\}$$

If (x, x') ∈ $\mathrm{Ind}_A(B)$, we say that objects x and x' are indiscernible from each other relative to attributes from B. The notation $[x]_B$ denotes equivalence classes of $\mathrm{Ind}_A(B)$. For X ⊆ U, the set X can be approximated only from information contained in B by constructing a B-lower and B-upper approximation denoted by $\underline{B}X$ and $\overline{B}X$

respectively, where $\underline{B}X = \{x \mid [x]_B \subseteq X\}$ and $\overline{B}X = \{x \mid [x]_B \cap X \neq \emptyset\}$. The objects of $\underline{B}X$ can be classified as members of X with certainty relative to attributes in B, while the objects of $\overline{B}X$ can only be classified as possible members of X relative to attributes in B. Let $BN_B(X) = \overline{B}X - \underline{B}X$. A set X is rough if $BN_B(X)$ is not empty. The notation $\alpha_B(X)$ denotes the accuracy of an approximation defined in (1).

$$\alpha_B(X) = \frac{|\underline{B}X|}{|\overline{B}X|} \quad (1)$$

where | X | denotes the cardinality of the non-empty set X, and $\alpha_B(X) \in [0, 1]$. The set X with respect to B is crisp precise, if $\alpha_B(X) = 1$. Otherwise, the set X is rough with respect to B, if $\alpha_B(X) < 1$.

2.2 Rough Membership Functions

A rough membership function (*rm function*) makes it possible to measure the degree that any specified object with given attribute values belongs to a given set X [20], [24]. A rm function μ_X^B is defined relative to a set of attributes $B \subseteq A$ in information system S = (U, A) and a given set of objects X. Let $B \subseteq A$, and let X be a set of observations of interest. The degree of overlap between X and $[x]_B$ containing x can be quantified with the rough membership function in (2).

$$\mu_X^B : U \rightarrow [0,1] \text{ defined by } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \quad (2)$$

2.3 Example

A high voltage direct current (dc) transmission system connected between alternating current (ac) source and ac power distribution system has two converters. In the case where the flow of power is from the ac side to the dc side as in Fig. 1, then a converter acts as a rectifier in changing ac to dc. The inverter in Fig. 1 converts dc power to ac power at desired output voltage and frequency. The Dorsey Station in the Manitoba Hydro system, for example, acts an inverter in converting dc to ac, which is distributed throughout North America.

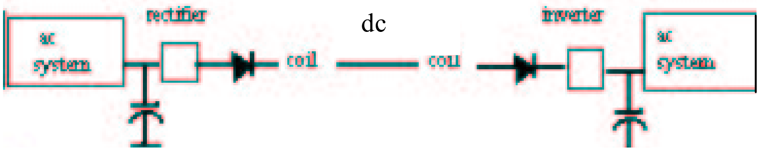

Fig. 1 dc Link Between ac Systems

Power system faults are recorded in files F. Let A be the set of attributes such as phase current, current setting, and maximum phase current. Assume that $\overline{A}F = \{file3, file4, file7, file8\}$. Further, assume that $[f]_A$ = {equivalence class consisting of files representing a known power system fault} = {file3, file9, file10}. Then consider the degree of overlap between $\overline{A}F$ and $[f]_A$ (see Fig. 2).

$$\mu_{\overline{A}F}^A(f) = \frac{|\overline{A}F \cap [f]_A|}{|[f]_A|} = \frac{1}{4}$$

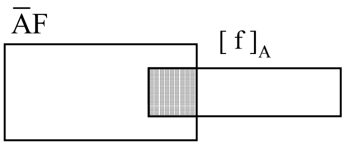

Fig. 2 Overlapping regions

2.4 Rough Petri Nets

In what follows, it is assumed that the reader is familiar with classical Petri nets in Petri [25] and coloured Petri nets in Jensen[26]. Rough Petri nets are derived from coloured and hierarchical Petri nets as well as from rough set theory. A rough Petri net provides a basis for modeling, simulating and analyzing approximate reasoning and decision systems.

Def. 1 *Rough Petri Net*. A rough Petri net is a structure (Σ, P, T, A, N, C, G, E, I, W, $\Re$, ξ) where

- Σ is a finite set of non-empty data types called color sets.
- P is a finite set of places.
- T is a finite set of transitions.
- A is a finite set of arcs such that $P \cap T = P \cap A = T \cap A = \emptyset$.
- N is a 1-1 node function where N: $A \rightarrow (P \times T) \cup (T \times P)$.
- C is a color function where C: $P \rightarrow \Sigma$.
- G is a guard function where G: $T \rightarrow [0, 1]$.
- E is an arc expression function where E: $A \rightarrow$ Set_of_Expressions where E(a) is an expression of type C(p(a)) and p(a) is the place component of N(a).
- I is an initialization function where I: $P \rightarrow$ Set_of_Closed_Expressions where I(p) is an expression of type C(p).
- W is a set of strengths-of-connections where ξ: $A \rightarrow W$.
- $\Re = \{\rho_\sigma \mid \rho_\sigma$ is a method$\}$
- ρ_σ is a method which either constructs or analyzes a rough set structure.

Let S be an information system S. Examples of rough set structures constructed by ρ_σ from decision system tables are upper approximations and the set OPT(S) of all rules derived from reducts of a decisions system table for S. Borrowing from coloured Petri nets, a rough Petri net provides data typing (colour sets) and sets of values of a specified type for each place. The expression E(p, t) specifies the input associated with the arc from input place p to transition t, and the expression E(t, p') specifies a transformation (activity) performed by transition t on its inputs {E(p, t)} to produce an output for place p'.

2.5 Receptor Processes

Let X be a set of inputs, B a set of attributes, and let $[u]_B$ be an equivalence class defined with respect to the set of attributes B. Let ρ be a procedure which constructs $\overline{B}X$ and let $\mu^{B}_{\overline{B}X}(x)$ compute the degree of overlap between $\overline{B}X$ and $[u]_B$ (see Fig. 3).

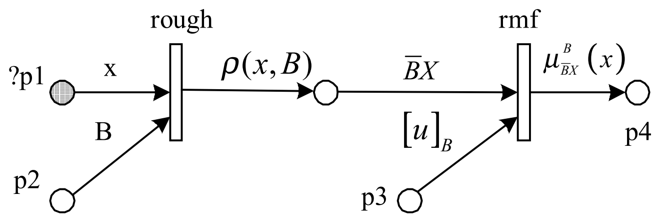

Fig. 3 Sample Rough Petri Net

The input place labeled ?p1 in Fig. 3 represents a Dill receptive process [27], which is always input ready. The transition labeled "rough" in Fig. 3 is enabled by the input of signal x and set of attributes B. When this transition fires, ρ(x, B) constructs $\overline{B}X$. The availability of $\overline{B}X$ and equivalence class $[u]_B$ enables the transition labeled "rmf" in Fig. 3. Whenever the rmf transition fires, $\mu^{B}_{\overline{B}X}(x)$ computes the degree of overlap between $[u]_B$ and $\overline{B}X$. Further, it should be noted that the Petri net in Fig. 3 is a model of what is known as an approximation neuron [14]. An approximation neuron is a computational unit in the input layer of a neural network with a design derived from rough set theory. Of particular interest in the Petri net in Fig. 3 is the receptor process denoted by ?p1.

Def. 2 *Receptive Process*. A receptor process is a process which provides an interface between a system and its environment by recording the value of each stimulus whenever stimuli are detected.

In effect, a receptor process is said to be always input-ready. That is, in the case where an environment is a source of continuous stimulation, a receptor process continuously enables any transition connected to it. The advantage in constructing such a Petri net model of sensor-dependent systems is that it facilitates simulation of the responses of a system to stimuli. In addition, a number of tests such as reachability of each of the transitions in the model can be performed.

2.6 Guarded transitions

A guard G(t) is an enabling condition associated with transition t. In a rough Petri net, various families of guards can be defined which induce a level-of-enabling of transitions [12]. Consideration of level-of-enabling stems from guards named after Jan Lukasiewicz [28], who inaugurated the study of multivalued logic. Let U denote a universe of objects, and let $X \subseteq U$. Let $\lambda: U \rightarrow [0, 1]$.

Def. 3 *Lukasiewicz Guard*. A *Lukasiewicz guard* on transition t with input x is a higher order propositional function P(λ(x)) labeling the transition t with input x and evaluation λ(x). The guard $P(\lambda(x)) = \lambda(x) \in (0,1]$, where $0 < \lambda(x) \leq 1$ enables t.

Let a Lukasiewicz guard be defined on a transition t as in Fig. 4.

Sample model for a Lukasiewicz guard:

$$\lambda(x) = e^{\left[\frac{-(x-\overline{x})^2}{s^2}\right]}$$

$$P(\lambda(x)) = \begin{cases} \lambda(x), & \text{if } \lambda(x) \in (0.75, 1] \\ 0, & \text{otherwise} \end{cases}$$

Sample enabling:

$\overline{x} = 45$

$s = 20$

$\lambda(44) = 0.9975$

P(λ(44)) = 0.9975, which enables transition t

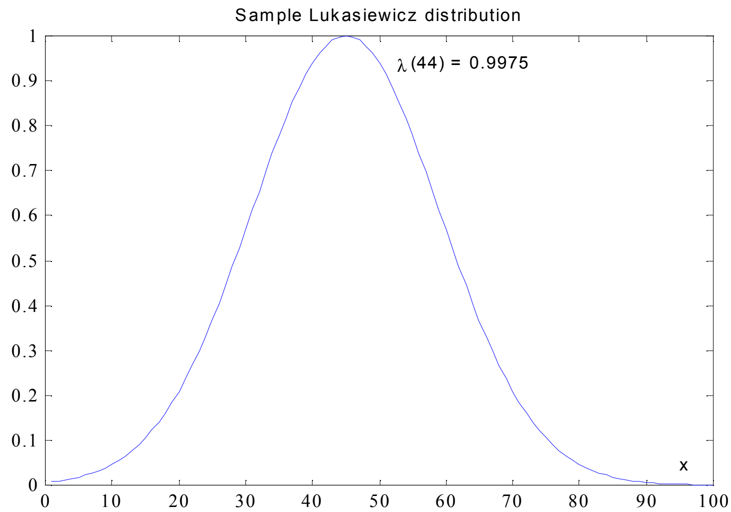

Fig. 4 Sample Lukasiewicz Guard

3 Sensors and Filters

A sensor converts some form of stimulus into a measurable output. By contrast with transducers, the output of some sensors might not be some form of energy. For example, the output of a digital thermometer is numerical. Hence, a digital thermometer would be classified as a sensor but not a transducer. Notice, also, that some sensors are also transducers. Most sensors employ one or more transduction mechanisms to produce a usable output signal.

Def. 4 *Sensor*. A sensor is a device which responds to each stimulus by converting its measured input to a some form of usable output.

A sensor responds to each stimulus which the sensor is designed to detect. In effect, a sensor is always input-ready.

Prop. 1 A receptor process can be modeled as a sensor.

Hence, the receptor process in Fig. 3 can be modeled as some form of sensor, which supplies input to a rough transition. Let π, X, V denote an internal recording function of a receptor process ?p, set of stimuli, and set of values in the π range, respectively. Further, π: X $\rightarrow$ V. A Petri net model of the receptor process ?p is given in Fig. 5.

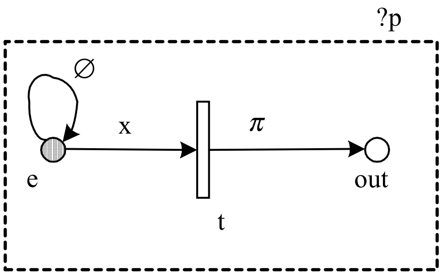

Fig. 5 Receptor Process

Place e in Fig. 5 represents the environment of the receptor process ?p. Let ∅ denote the empty set (no stimuli from the environment). The "looping" arc notation on place e is borrowed from traditional finite state machine theory. In the absence of stimuli from the environment, there is a continuous looping back to place (state) e. Each new stimuli x from the environment enables transition t in Fig. 5. In the case where stimuli from the environment continuously occur, transition t is continuously enabled. The two arcs leading from place e satisfy the "always input ready" condition of receptor processes. Whenever transition t fires, π maps x to a value, which is recorded in the place labeled "out" in Fig. 5.

3.1 Example.

A light sensor L responds to light (i.e., a detected number of photons per second) by a voltage ouput (e.g., 0.87 volts when a photoresistor is struck by 2.3 x 10^{13} photons per second) [29]. Light sensors can be used to enable mobile robots to stop its activity in the dark or to move toward a beacon. Let R, R_L, V, V_{PE} be the ohms of a resistor in the photoresistor circuit for L, resistance of L, voltage input of the circuit containing L, and voltage of a L, respectively, where

$$V_{PE} = \frac{R_L}{R + R_L} V$$

To obtain a Petri net model of sensor L, replace x in Fig. 5 with R,R_L,V and π with V_{PE}, respectively (see Fig. 6).

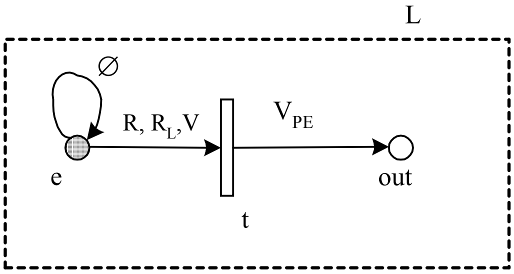

Fig. 6 Petri Net Model of a Light Sensor

3.2 Filters

Filters provide a means of weeding out sensor signals which are not wanted and in modifying or manipulating sensor signals to facilitate their usage in a system. This is case in noisy communication system, for example, where electric filters are used to eliminate signal contamination.

Def. 5 *Filter.* A filter is a mechanism which selects readings of a sensor relative to one or more selection criteria.

In the case of an electric filter, its selection can include the modification, reshaping or manipulation of the frequency spectrum of an electric signal according to some prescribed requirements [30]. Filters are useful, for example, in eliminating signal contamination and in separating relevant from irrelevant sensor inputs. It is in this latter sense of filter-usefulness that filters are considered in the context of rough Petri nets.

Notice that a Lukasiewicz guard $\lambda(x)$ can be used as the basis for a model of a filter on sensor input of an approximation neuron, since there is interest in preventing input signals with approximately zero strength from enabling an input transition. To complete the modeling of an input sensor filter, a restricted Lukasiewicz guard can be introduced (see Fig. 4).

Def. 6 *Restricted Lukasiewicz Guard*. A *restricted Lukasiewicz guard* on transition t with input x is a function $P(\lambda(x)) = \lambda(x) \in [a,b] \subset [0,1]$, where $a \leq \lambda(x) \leq b$ enables t. In effect,

$$P(\lambda(x)) = \begin{cases} \lambda(x), & \text{if } x \in [a,b] \subset (0,1] \\ 0, & \text{otherwise} \end{cases}$$

If we assume that the input to a Lukasiewicz guard comes from a sensor, then such a guard acts a filter.

Prop. 2 A restricted Lukasiewicz guard with sensor input is a filter.

Notice that a Lukasiewicz guard can be defined over [a, b] = [a, 1] where a > 0. In effect, $\lambda(x) \in (0,1]$.

Prop. 3 A Lukasiewicz guard with sensor input is a filter.

3.3 Example

Consider a large room where a uniform temperature must be achieved with a combination of sensors, sensor filters and heating elements (see Fig. 7). Ambient temperatures are a problem in controlling room temperature. It may be warm in one part of the room, and cold in another part of the room. For simplicity, a single sensor plus filter and single heating element are represented in Fig. 7. Let T, X denote a set of sensors and reference set of sampled temperatures, respectively. Let $\tau(x)$ denote a sensor which records the current temperature x. Also assume that τ is connected to a filter ρ, which maps the recorded temperature to a value into one of the intervals shown in Fig. 8. A sample decision table reflecting reactions of the system to sensed temperatures is shown in Fig. 8. The decision d has a value chosen by an expert for any global temperature state (represented by measurements of temperature in all points of a sufficiently dense set of points in a room). For example, $\rho(\tau(x_1))$ returns a value in a vector of temperature measurements in [0.00, 0.10] for different points in a room. The corresponding approximation regions are shown in Fig. 9.

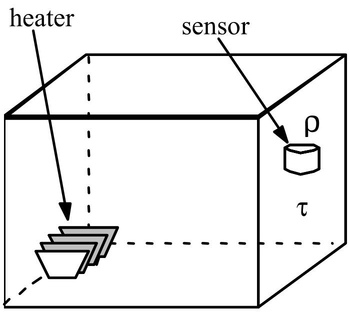

Fig. 7 Heat controlled room

	$\rho(\tau(x))$	d
x_1	[0.00,0.10]	heatOn
x_2	[0.05,1.50]	heatOff
x_3	[4.00,6.00]	heatOn
x_4	[4.00,6.00]	heatOff
x_5	[8.00,9.00]	heatOff
x_6	[0.00,0.10]	heatOn

Fig. 8 Decision Table

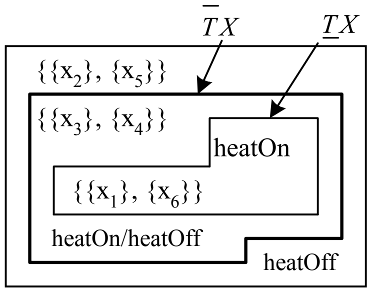

Fig. 9 Approximation Regions

A Petri net model of the sensor-filter combination used to construct the decision table in Fig. 8 can now be given. First, a receptor process model for a receptor place denoted ?tempSensor representing a temperature sensor is given in Fig. 10. The temperature sensor is then connected to a model of the filter in form of a Lukasiewicz guard (see Fig. 11). To complete the model of the system used to produce approximation regions specified by $\overline{TX}$, a rough transition ρ is introduced in Fig. 12.

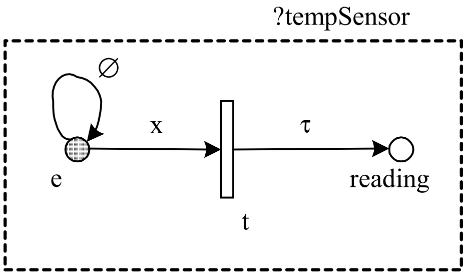

Fig. 10 Receptor Model of Sensor

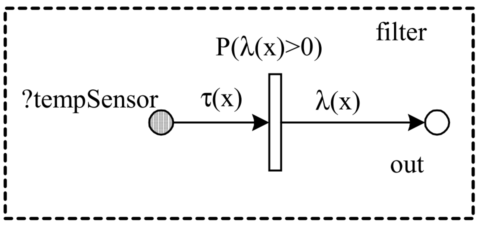

Fig. 11 Model of Temperature Filter

The model in Fig. 12 can be taken a step further to create a model of rough neuron.

Let $\overline{TX}$, $[u]_T$ denote an input place which supplies an upper approximation (see Fig. 12), and input place which supplies a reference equivalence class, respectively. Further, assume that $\overline{TX}$, $[u]_T$ supply input to a transition which computes a rough membership function value as in Fig. 3.

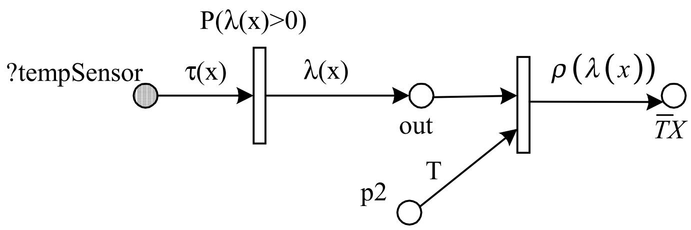

Fig. 12 Model of Partial Approximation System

Such a Petri net provides a simple model of what is known as a rough neuron, which can be used in classifying room temperature readings.

4 Concluding Remarks

This paper introduces Petri net models of sensors and filters in the context of rough sets. Input places in models of sensor-driven systems such as controllers are conceptualized as Dill receptor processes, which are always input-ready. By definition, a receptor process can be modeled as a sensor. In a rough Petri net, conditional activation (firing) of a transition and associated action is made possible by introducing Lukasiewicz guards on transitions. Such guards are defined in the context of multivalued logic. Lukasiewicz guards provide a basis for modeling filters.

Acknowledgement.

Recent discussions concerning sensors and Petri net modeling with Dario Maravall and Luis Baumela, Department of Artificial Intelligence, Technical University of Madrid, Spain have been particularly helpful and are gratefully acknowledged. Support for this project from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Manitoba Hydro is gratefully acknowledged.

References

1. A. Skowron, Z. Suraj, A rough set approach to real-time state identification for decision making, Institute of Computer Science Research Report 18/93, Warsaw University of Technology.
2. A. Skowron, Z. Suraj, Rough sets and concurrency, Bull. Polish Academy of Science and Technology, vol. 41, no. 3, 1993, 237-254.
3. A. Skowron, Z. Suraj, Synthesis of concurrent systems specified by information systems, ICS Research Report 39/94, Institute of Computer Science, Warsaw Institute of Technology, 1994.
4. A. Skowron, Z. Suraj,A parallel algorithm for real-time decision making: a rough set approach, Journal. of Intelligent Information Systems, Vol. 7,1996, 5-28.
5. A. Skowron, Z. Suraj, Discovery of concurrent data models from experimental data tables: A rough set approach. In: U. Fayyad, R. Uthurusamy (Eds.), Proc. of the First International Conference on Knowledge Discovery and Data Mining. MA: The MIT Press, 1995, 288-293.
6. Z. Suraj, Discovery of concurrent data models from experimental tables: A rough set approach, Fundamenta Informaticae, vol. 28, nos. 3-4, 1996, 353-376.
7. J.F. Peters, Time and clock information systems: Concepts and roughly fuzzy Petri net models. In: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery. Heidelberg, Physica-Verlag, Vol. 2, 1997, 385-418.
8. J.F. Peters, X. Xu, V. Degtyaryov, Sensors, Report CIL08.06.00.02, Computational Intelligence Laboratory, University of Manitoba, 8 June 2000.
9. J.F. Peters, W. Pedrycz, Software Engineering: An Engineering Approach, NY, John Wiley & Sons, 2000.
10. J.F. Peters, S. Skowron, Z. Suraj, W. Pedrycz, S. Ramanna, Approximate real-time decision-making: Concepts and roughly fuzzy Petri net model, International Journal of Intelligent Systems, vol. 14, no. 4, 1999, 4-37.
11. J.F. Peters, A. Skowron, Z. Suraj, S. Ramanna, A. Paryzek, Modeling real-time decision-making systems with rough fuzzy Petri nets. In: Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT98), Vol. 1, Aachen, Germany, pp. 985-989.
12. J.F. Peters, A. Skowron, Z. Suraj, S. Ramanna, Guarded transitions in rough Petri nets. In: Proc. of 7th European Congress on Intelligent Systems & Soft Computing (EUFIT'99), 13-16 Sept. 1999, Aachen, Germany.
13. J.F. Peters, K. Ziaei, S. Ramanna, S., Approximate Time Rough Control: Concepts and Application to Satellite Attitude Control, In: Proc. Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC'98), Warsaw, Poland, Lecture Notes in Artificial Intelligence 1424, Berlin, Springer-Verlag, 1998, 491-498.
14. J.F. Peters, A. Skowron, Z. Suraj, L. Han, S. Ramanna, Design of rough neurons: Rough set foundation and Petri net model, International Symposium on Methodologies for Intelligent Systems (ISMIS'2000) [to appear].
15. Z. Pawlak, Rough sets, Int. J. of Computer and Information Sciences, Vol. 11, 1982, 341-356.
16. Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Boston, MA, Kluwer Academic Publishers, 1991.

17. Z. Pawlak, Reasoning about data--A rough set persepective. Lecture Notes in Artificial Intelligence 1424, L. Polkowski and A. Skowron (Eds.). Berlin, Springer-Verlag, 1998, 25-34.
18. Z. Pawlak, Decision rules, Baye's rule and rough sets. In: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Artificial Intelligence No. 1711. Berlin: Springer-Verlag, 1999, 1-9.
19. Z. Pawlak, Drawing conclusions from data--The rough set way, Int. Journal of Intelligent Systems [to appear].
20. Z. Pawlak, A. Skowron, Rough membership functions. In: R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, NY, John Wiley & Sons, 1994, 251-271.
21. L. Han, J.F. Peters, S. Ramanna, R. Zhai, Classifying faults in high voltage power systems: A rough-fuzzy neural computational approach. In: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Notes in Artificial Intelligence 1711. Berlin: Springer, 1999, 47-54.
22. L. Polkowski, A. Skowron, Rough neural computing, In: Proc. Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC'00), Banff, Alberta, Canada [to appear].
23. J.F. Peters, A. Skowron, L. Han, S. Ramanna, Towards rough neural computing based on rough membership functions: Theory and Application, Rough Sets and Current Trends in Computing (RSCTC'2000) [submitted].
24. J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, Rough sets: A tutorial. In: S.K. Pal, A. Skowron (Eds.), Rough Fuzzy Hybridization: A New Trend in Decision-Making. Singapore: Springer-Verlag, 1999, 3-98.
25. Petri, C.A., 1962, "Kommunikation mit Automaten", Schriften des IIM Nr. 3, Institut für Instrumentelle Mathematik, Bonn, West Germany.
26. Jensen, K., 1992, "Coloured Petri Nets--Basic Concepts, Analysis Methods and Practical Use 1", Berlin, Springer-Verlag.
27. D.L. Dill, Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits. Cambridge, MA: MIT Press, 1989.
28. J. Lukasiewicz, O logice trojwartosciowej, Ruch Filozoficzny 5, 1920, 170-171. See "On three-valued logic", English translation in L. Borkowski (Ed.), Jan Lukasiewicz: Selected Works, Amsterdam, North-Holland, 1970, pp. 87-88.
29. J.L. Jones, A.M. Flynn, Mobile Robots: Inspiration to Implementation. Wellesley, MA: AK Peters, 1993.
30. A. Antoniou, General characteristics of filters. In: W.-K. Chen, The Circuits and Filters Handbook. Boca Raton, FL: CRC Press, 1995, 2129-2238.
31. Z. Suraj, Rough set methods for the synthesis and analysis of concurrent processes, Report Nr 893, Institute of Computer Science, Polish Academy of Sciences, Warsaw, December 1999.
32. J. de Lope, D. Maravall, J.G. Zato, Topological modeling with fuzzy Petri nets for autonomous mobile robots. In: A.P. del Pobil, J. Mira, M. Ali (Eds.), Proc. 11th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA-98-AIE, Lecture Notes in Artificial Intelligence 1416, Springer-Verlag, Berlin, 1998, 290-299.