# From Data to Nets with Inhibitor Expressions: A Rough Set Approach

Krzysztof Pancerz[1], Zbigniew Suraj[1]

**Abstract**
In the paper we present a new method of concurrent model construction from data based on rough set approach. As a model for concurrency, nets with inhibitor expressions are chosen.

## 1. Introduction

Information systems [2] are used for representing knowledge. In the paper we apply them as a tool for specification of concurrent systems. We present a method for constructing a concurrent model from an arbitrary information system *S*, in the form of a net $N_S$ with inhibitor expressions. The net $N_S$ exhibits the following property: the reachability set of the net $N_S$ corresponds exactly to the set of all global states consistent with all rules valid in *S*. For this purpose we define in the paper the notion of a net with inhibitor expressions (*IE*-net). This definition is an extension of the notion of the net with inhibitor arcs (*IA*-net) [3]. Our method for constructing a net with inhibitor expressions consists of two stages. In the first stage, all dependencies represented by means rules between local processes (attributes) in the system are extracted from a given set of global states (objects). In the second stage, a net corresponding to these dependencies is built.
The construction method of a net proposed in the paper is easier to understand and simpler to implement than presented in [5]. Some new relationships of the knowledge represented by information systems and its Petri net models can also be considered. Constructed nets make possible evaluating the behavior of concurrent processes systems specified by information systems, and tracing computations in rules derived from information systems.
Application of concurrent model obtained from a given information system in the form of a place-transition net has recently been discussed in [5], [6], [7]. In this approach the net has been constructed on the base of the rules computed for the information system. Our approach is based on a new notion of so called inhibitor rules extracted from a given information system.
The contribution of this paper is a new approach to concurrent model construction from data.

## 2. Basic notions of information systems

In this section we recall some notions related to information systems.
An *information system* is a pair $S = (U, A)$, where *U* is a nonempty, finite set of objects, called the *universe*, *A* is a nonempty, finite set of *attributes*, i.e. $a : U \to V_a$ for $a \in A$, where $V_a$ is called the value set of *a*. The set $V = \bigcup_{a \in A} V_a$ is said to be the *domain* of *A*.
Elements of *U* can be interpreted as global states of system, however attributes can be interpreted as local processes in a given system. Hence with every local process $a \in A$ is associated a finite set $V_a$ of its internal states. A behavior of a system can be presented in a form of a table. Each row in the table includes record of local states of processes from *A*, and each record is labeled by an element from the set *U* of global states of the system. The columns in the table are labeled by attributes (processes).
Let $S = (U, A)$ be an information system. With any subset of attributes $B \subseteq A$ we associate a binary relation $ind(B)$, called an *indiscernibility relation*, which is defined by:

$$ind(B) = \{(u, u') \in U \times U : \forall_{a \in B} a(u) = a(u')\}.$$

[1] Chair of Computer Science Foundations, University of Information Technology and Management, Rzeszow, Poland

Any minimal subset $B \subseteq A$ such that $ind(B) = ind(A)$ is called a *reduct* in the information system *S* and denoted by *R*. The set of all reducts in *S* is denoted by $RED(S)$.
Let $S = (U, A)$ be an information system, where $A = \{a_1,...,a_m\}$, and *V* is the domain of *A*. Pairs $(a, v)$, where $a \in A$, $v \in V$ are called *descriptors* (over *A* and *V*). By $DESC(A,V)$ we denote the set of all descriptors over *A* and *V*. Instead of $(a, v)$ we write also $a = v$ or $a_v$.
A *rule* of information system [2] is any expression of the following form:

$$a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \Rightarrow a_p = v_p,$$

where $a_p, a_{i_j} \in A$, $v_p \in V_{a_p}, v_{i_j} \in V_{a_{i_j}}$ for $j = 1,...,r$, and $\wedge$ denotes conjunction (the classical propositional operator).
The rules describe the relationship between values of the attributes in the information system. The fact that the rule is true in *S* we denote in the following way:

$$a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} a_p = v_p.$$

The set of all rules computed for a given reduct $R \in RED(S)$ is denoted by $OPT(S, R)$. However the set of all rules in information system *S* is denoted by $OPT(S)$. Hence

$$OPT(S) = \bigcup \{OPT(S, R) : R \in RED(S)\}.$$

A method for generating the rules in this form from a given information system has been described (e.g. in [5], [6]).
An *inhibitor rule* of information system is any expression of the following form:

$$a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} \neg(a_p = v_p),$$

where: $a_p, a_{i_j} \in A$ and $v_p \in V_{a_p}, v_{i_j} \in V_{a_{i_j}}$ for $j = 1,...,r$, and $\neg$ denotes negation (the classical propositional operator).
The inhibitor rules describe values of a given attribute which cannot exist when other attributes obtain definite values.
For each rule of the form:

$$a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} a_p = v_p$$

we can compute the set of inhibitor rules of the following form:

$$a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} \neg(a_p = v_{p_1})$$

$$......................................................$$

$$a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} \neg(a_p = v_{p_k})$$

where $v_{p_j} \in V_{a_p} - \{v_p\}$ for $j = 1,...,k$.
The set of all inhibitor rules in the information system *S* corresponding to the set $OPT(S)$ is denoted by $INH(S)$.

## 3. Nets with inhibitor expressions

We assume that the reader is familiarized with the basic notions of Petri nets [4]. In this section we introduce the notion of a net with inhibitor expressions.
Let $S = (U, A)$ be an information system, and $DESC(A,V)$ be the set of all descriptors over *A* and *V*. A *net with inhibitor expressions* associated with $DESC(A,V)$ is a tuple $IEN = (P, T, F, W, M_0, E, \varepsilon)$, where *P, T, F, W,* $M_0$ are defined the same as for a net with inhibitor arcs (*IA*-net) [3], i.e. *P* is a set of *places*, *T* is a set of *transitions*, *F* is a *flow relation*, *W* is a *weight function*, $M_0$ is an *initial marking* and

- *E* is a set of *Boolean expressions* over $DESC(A,V)$,

- $\varepsilon : T \rightarrow E$ is an *expression function*, a mapping from a set of transitions *T* into a set of Boolean expressions *E*.

With every transition $t \in T$ we associate a Boolean expression $e \in E$ over $DESC(A,V)$. This expression we call an *inhibitor expression*. An inhibitor expression for a given transition is associated with all places coming to this transition by inhibitor arcs. A Boolean variable corresponding to place $p$ is true when $M(p) \geq W((p,t))$ and false otherwise, where *M* is a marking and *W* is a weight function for an inhibitor arc. A transition *t* can be *fired* if and only if the value of an inhibitor expression is false and firing conditions of *t* corresponding to place-transition net (*PT*-net) [4] are satisfied. If none places are connected to a given transition $t$ by inhibitor arcs then an inhibitor expression has a form $e = 0$ (where 0 denotes the Boolean value *false*) and it is not represented in a net.

## 4. Constructing a net with inhibitor expressions for an information system

Having already computed inhibitor rules for a given information system we can create its concurrent model in the form of a net with inhibitor expressions. Below we describe a method for creating such concurrent model. We only use nets in which all markings are binary for any place and so are weights of all arcs.

In the net which is a concurrent model of a given information system there are two kinds of places:
(1) places labeled by names of attributes of an information system, which can be interpreted as temporary states of processes,
(2) places labeled by all descriptors from $DESC(A,V)$ set, which can be interpreted as local states of every process.

A method for constructing a concurrent model consists of two steps. First, we construct the net representing all processes in a given information system. This net is constructed with using elements of *PT*-net. Next, the net obtained in the first step is extended by adding inhibitor arcs and inhibitor expressions. The inhibitor expressions are determined from inhibitor rules computed for a given information system associated with the constructed net. The initial marking $M_0$ of a constructed net is defined as follows:

$$M_0(p) = \begin{cases} 1 \text{ for all places described by the condition (1)}, \\ 0 \text{ otherwise.} \end{cases}$$

Inhibitor expression has a disjunctive normal form. Every implicant of expression corresponds to one inhibitor rule associated with a given transition. For example, if we have inhibitor rules of the following form:

$p_0 \Rightarrow \neg q_1,\ p_1 \Rightarrow \neg q_1,\ p_2 \wedge p_3 \Rightarrow \neg q_1,$

then the corresponding net has the form illustrated in Figure 1,

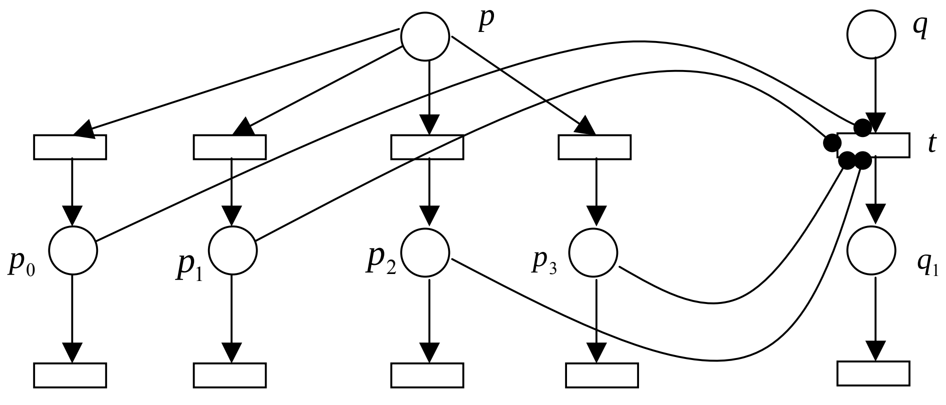


Figure 1.

and inhibitor expression for the transition *t* shown in this figure is as follows:

$e = p_0 \vee p_1 \vee (p_2 \wedge p_3)$.

**Example**. Let us consider an information system $S = (U, A)$ such that $U = \{u_1, u_2, u_3, u_4\}$, $A = \{a, b\}$ and the values of the attributes are defined as in the following Table 1.

Table 1

| $U \backslash A$ | $a$ | $b$ |
|---|---|---|
| $u_1$ | 0 | 1 |
| $u_2$ | 1 | 0 |
| $u_3$ | 0 | 2 |
| $u_4$ | 2 | 0 |

By applying methods for generating the reducts and the rules described in [5], [6] for the system *S* we obtain one reduct $R_1 = \{a, b\}$ and the set of all rules corresponding to all nontrivial dependencies between attributes of the following form:

$OPT(S) = \{a_1 \underset{S}{\Rightarrow} b_0, a_2 \underset{S}{\Rightarrow} b_0, b_1 \underset{S}{\Rightarrow} a_0, b_2 \underset{S}{\Rightarrow} a_0\}$.

On the base of this set we get the following set $INH(S)$ of inhibitor rules for this system:

$INH(S) = \{a_1 \underset{S}{\Rightarrow} \neg b_1, a_1 \underset{S}{\Rightarrow} \neg b_2, a_2 \underset{S}{\Rightarrow} \neg b_1, a_2 \underset{S}{\Rightarrow} \neg b_2, b_1 \underset{S}{\Rightarrow} \neg a_1, b_1 \underset{S}{\Rightarrow} \neg a_2, b_2 \underset{S}{\Rightarrow} \neg a_1, b_2 \underset{S}{\Rightarrow} \neg a_2\}$.

The concurrent model of *S* in the form of a net with inhibitor expressions is shown in Figure 2. The inhibitor expressions for all transitions of the constructed net are defined as in Table 2.

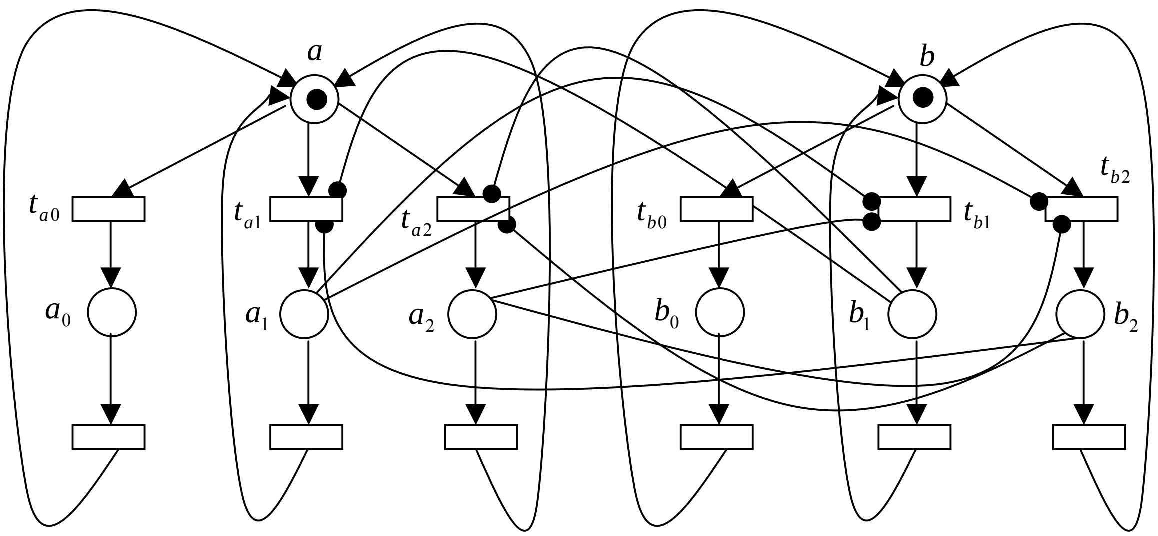


Figure 2.

Table 2

| *Transition* | *Inhibitor expression* |
|---|---|
| $t_{a0}$ | $e_{a0} = 0$ |
| $t_{a1}$ | $e_{a1} = b_1 \vee b_2$ |
| $t_{a2}$ | $e_{a2} = b_1 \vee b_2$ |
| $t_{b0}$ | $e_{b0} = 0$ |
| $t_{b1}$ | $e_{b1} = a_1 \vee a_2$ |
| $t_{b2}$ | $e_{b2} = a_1 \vee a_2$ |

Every received global state in this model requires verification. We check for every local state the following condition: if $e_d = 1$ then $M(d) = 0$, where $M$ is a marking of the net reachable from the initial marking $M_0$ of the net and $d \in DESC(A,V)$. If this condition is not satisfied for any local state then a given global state is not permissible.

## 5. Conclusions

The paper concerns a new approach to concurrency, based on rough set theory. Nets with inhibitor expressions built on the base of inhibitor rules extracted from a given information system have been chosen as a model for concurrency. Applying nets with inhibitor expressions for representing inhibitor rules of a given information system allows for:

- finding out all global states of the modeled system, consistent with all dependencies of a given system;
- observing concurrent and sequential subsystem of the system;
- graphic representation of dependencies between the processes within an information system and their dynamic interactions.

In the next paper we shall study applications of the presented herewith formalism for synthesis of concurrent systems.
Our results seem to have some significance for investigating the relationships between concurrency and rough set theory. The new approach will stimulate the theoretical research on both problems and their new practical applications, among others in knowledge discovery area, real-time decision making, control design for discrete event systems.
The method proposed in the paper has been implemented according to a new programming technology in the ROSECON system [1] running on IBM PC computers under Windows operating system. The ROSECON system is being developed in the Chair of Computer Science Foundations at the University of Information Technology and Management in Rzeszów.

## Acknowledgements

Supported by Polish National Committee for Scientific Research (KBN) grant No. 8T11C02519.

## References

[1] Pancerz, K., Suraj, Z.: ROSECON – system automatycznego odkrywania modeli współbieżnych z tablic danych. In: *Proceedings of the IX Environmental Conference on Mathematics and Informatics*, Korytnica, June 12-16, 2002, Poland (in Polish).

[2] Pawlak, Z.: *Rough Sets – Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, Dordrecht 1991.

[3] Peterson, J.L.: *Petri net theory and the modeling of systems.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.

[4] Reisig, W.: *Petri Nets. An Introduction.* EATCS Monographs on Theoretical Computer Sciences, Vol. 4, Springer-Verlag, Berlin 1985.

[5] Skowron, A., Suraj, Z.: Rough Sets and Concurency. *Bulletin of PAS*, Vol.41, No. 3, 1993, pp. 237 – 254.

[6] Suraj, Z.: Discovery of Concurrent Data Models from Experimental Tables: A Rough Set Approach. *Fundamenta Informaticae*, Vol. 28, No. 3, 4, December 1996, pp. 353 – 376.

[7] Suraj, Z.: Rough Set Methods for the Synthesis and Analysis of Concurrent Processes. In: L. Polkowski, S. Tsumoto, T.Y. Lin (eds.) *Rough Set Methods and Applications*, Springer, Berlin 2000, pp. 379-488.