Control and Cybernetics

vol. **40** (2011) No. 2

Toward interactive rough–granular computing*

by

Andrzej Skowron and Piotr Wasilewski

Institute of Mathematics, The University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
e-mail: skowron@mimuw.edu.pl; piotr@mimuw.edu.pl

Abstract: Computations in Rough-Granular Computing (RGC) are performed on (information) granules. The rough set approach is used in RGC for inducing granules approximating other granules about which imperfect knowledge is given only. For modeling of complex systems, it is important to extend the RGC approach to Interactive Rough-Granular Computing (IRGC) based on interactions of granules. In this paper, we discuss some fundamental issues for interaction of granules such as general scheme of interactions and the role of dynamic attributes and dynamic information systems in modeling interactive computations.

Keywords: rough sets, granular computing, interactive computations, perception based computing, perception attributes, sensory attributes, action attributes, vague concepts, approximation of complex vague concepts, ontology approximation.

1. Introduction

We discuss some basic aspects of interactive computations on granules of information. In this section, we present a short introduction to rough-granular computing (RGC) and to interactive computing. Next, in the following sections we investigate some issues important for foundations of interactive rough-granular computing (IRGC).

Computations in granular computing are performed on granules (see, e.g., Pedrycz et al., 2008). Granules are constructed using granular calculi (see, e.g., Skowron and Stepaniuk 2008).

Granules are clumps of objects that are arranged together due to their similarity, indistinguishability or functionality. From a mathematical point of view, granules can be represented (exactly or at least to a degree) by sets, often from quite high levels of the powerset hierarchy. Granules can have an elementary structure (such as elementary neighborhoods of objects) or a complex structure

*Submitted: September 2010; Accepted: June 2011.

(such as cognitive agents, autonomous agents, coalitions of agents in multiagent system (MAS) (Sycara, 1998; Shoham and Leyton-Brown, 2009), sets of decision rules, clusters, complex patterns or classifiers in data mining). Granules sent by one agent may not be exactly “understandable” by other agents receiving these granules. In such cases, interacting agents would attempt to approximate the received granules. In RGC rough set tools are used for approximations of granules, such as vague concepts or classifications. Such rough set-based approximations are induced on the basis of partial information about objects and approximated concepts (or classifications). Granules approximating other granules are constructed using computations aiming at solving of some optimization tasks. The optimization criteria in searching for relevant granules are based on versions of the Rissanen minimum length principle (Pawlak and Skowron, 2007; Skowron and Stepaniuk, 2011). In searching for (sub)optimal solutions, it is necessary to construct compound granules using some specific kinds of operations such as generalization, specification or fusion. Granules are parameterized. By tuning these parameters we optimize the granules relative to their description size and the quality of data description, i.e., two basic components on which the optimization measures are defined. From mathematical point of view, granules are sets of different types, e.g., represented using the powerset hierarchy (Skowron and Stepaniuk, 2011). It is worthwhile to mention that in RGC it is also necessary to discover relevant types of induced granules in searching for approximations of higher order granules.

From this general description of tasks in RGC, it follows that together with the specification of elementary granules and operations on them it is necessary to define measures of granule quality (e.g., measures of their inclusion such as the rough inclusion measures, covering or closeness measures, Skowron and Stepaniuk, 2008) and tools for measuring the sizes of granules. Optimization strategies of already constructed (parameterized) granules are also very important.

Developing methods for the approximation of compound concepts expressing the result of perception is one of the main challenges of Perception Based Computing (PBC) (Zadeh, 1999, 2001). Concepts representing perceived information are expressed in natural language. The rough-granular approach proved successful in approximation of such concepts from sensory data and domain knowledge. This additional knowledge, represented by an ontology of concepts, is used to make it feasible to search for features (condition attributes) relevant to the approximation of concepts on different levels of the concept hierarchy defined by a given ontology. We later report on several experiments with the proposed methodology for the approximation of compound concepts from sensory data and domain knowledge. The approach is illustrated by examples relative to interactions of agents, ontology approximation, adaptive hierarchical learning of compound concepts and skills, behavioral pattern identification, planning, conflict analysis and negotiations, and perception-based reasoning. The presented results seem to justify the following claim of Lotfi A. Zadeh (2006):

In coming years, granular computing is likely to play an increasingly important role in scientific theories — especially in human-centric theories in which human judgment, perception and emotions are of pivotal importance.

The concept approximation problem is the basic problem investigated in machine learning, pattern recognition and data mining (Hastie et al., 2008). It is necessary to induce approximations of concepts (models of concepts) consistent (or almost consistent) with some constraints. In the most typical case, the constraints are defined by the training sample. For more complex concepts, we consider constraints defined by a domain ontology consisting of vague concepts and relations between them. Information about the classified objects and concepts is incomplete. In the most general case, the adaptive approximation of concepts is performed under interaction with the dynamically changing environment. In all these cases, searching for sub-optimal models relative to the minimum length principle (MLP) is performed. Notice that in the adaptive concept approximation, one of the components of the model should be the adaptation strategy. Components, involved in the construction of concept approximations, which are tuned in searching for sub-optimal models relative to MLP are called information granules. In rough granular computing (RGC), information granule calculi are used in the construction of classifier components and classifiers themselves (see, e.g., Skowron and Stepaniuk, 2011) satisfying given constraints. An important mechanism in RGC is related to generalization schemes making it possible to construct complex patterns from simpler patterns. Generalization schemes are tuned using, e.g., some evolutionary strategies.

Rough set theory, due to Zdzisław Pawlak (1982, 1991); Pawlak and Skowron (2007), is a mathematical approach to imperfect knowledge. The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians and mathematicians. Recently, it has also become a crucial issue for computer scientists, particularly in the area of artificial intelligence. There are many approaches to the problem of understanding and manipulating imperfect knowledge. The most successful one is, no doubt, the fuzzy set theory proposed by Lotfi A. Zadeh (1965). Rough set theory is another attempt to solve this problem. It is based on the assumption that objects and concepts are perceived through partial information about them. Due to this, some objects can be indiscernible. From this fact it follows that some sets cannot be exactly described by available information about objects; they are rough and not crisp. Any rough set is characterized by its (lower and upper) approximations. The difference between the upper and lower approximation of a given set is called its boundary. Rough set theory expresses vagueness by employing a boundary region of a set. If the boundary region of a set is empty then the set is crisp, otherwise the set is rough (inexact). A nonempty boundary region of a set means that our knowledge about the set is insufficient to define the set precisely. One can recognize that rough set theory is, in a sense, a formalization of the idea presented by Gotlob Frege (1903). There are many methods based on the combination of the rough set approach and the fuzzy set approach, as well as on other soft computing methods. The hybridization often helps to obtain granules with high quality, much higher than any single approach. Such methods are also used in RGC.

One of the consequences of perceiving objects using only available information about them is that for some objects one cannot decide whether they belong to a given set or not. However, one can estimate the degree, to which objects belong to sets. This is another crucial observation in building the foundations for approximate reasoning. In dealing with imperfect knowledge one can only characterize satisfiability of relations between objects to a degree, and not precisely. Among relations between objects, the rough inclusion relation, which describes to what degree objects are parts of other objects, plays a special role. A rough mereological approach (see, e.g., Polkowski and Skowron, 1996) is an extension of the Leśniewski mereology (Leśniewski, 1929) and is based on the relation *to be a part to a degree*.

The rough set approach offers also tools for approximate reasoning in multiagent systems (MAS). A typical example is one agent's approximation of concepts of an another agent. The approximation of a concept is based on a decision system representing information about objects perceived by both agents.

The developed strategies for inducing data models are often unsatisfactory for the approximation of compound concepts that occur in the perception process. Researchers from different areas have recognized the necessity to work on new methods of concept approximation (see, e.g., Breiman, 2001; Vapnik, 1998). The main reason for this is that these compound concepts are, in a sense, too far from measurements, making the search for relevant features in a very large space unfeasible. There are several research directions aiming to overcome this difficulty. One of them is based on interdisciplinary research, with knowledge pertaining to perception in psychology and neuroscience used to help dealing with compound concepts (see, e.g., Miikkulainen et al., 2005). There is a great effort in neuroscience towards understanding the hierarchical structures of neural networks in living organisms (Miikkulainen et al., 2005). Also mathematicians recognize the problem of learning as the main problem of the current century (Poggio and Smale, 2003). These problems are closely related to complex system modeling, as well. In such systems again the problem of concept approximation and its role in reasoning about perceptions is one of challenges nowadays. One should take into account that modeling complex phenomena entails the use of local models (captured by local agents, if one used the multi-agent terminology, Sycara, 1998) to be fused afterwards. The process involves negotiations between agents (Sycara, 1998) to resolve contradictions and conflicts in local modeling. This kind of modeling is increasingly important in dealing with complex real-life phenomena, which we cannot model using traditional analytical approaches. The latter approaches lead to exact models. However, the necessary assumptions behind them result in solutions that are too far from reality to be accepted. New methods or even a new science should therefore be developed for such modeling (Gell-Mann, 1994).

One of successful research directions, based on RGC is related to approximation of ontology representing in a hierarchical form domain knowledge. Discovery of relevant attributes on each level of hierarchy is supported by domain knowledge provided, e.g., by concept ontology together with illustration of concepts by means of samples of objects taken from the concepts and their complements. The developed methods of ontology approximation were successfully applied in different domains, such as risk prediction and medical therapy support from medical data and domain knowledge (see, e.g., Bazan, 2008; Bazan and Skowron, 2005b; Bazan et al., 2006c,b,a), prediction of the situation on the road (e.g., Bazan, 2008; Nguyen et al., 2006; Bazan et al., 2005, 2006c; Bazan and Skowron 2005a), or sunspot classification (see, e.g., Nguyen et al., 2005). Such application of domain knowledge, often taken from human experts, serves as another example of interaction of a system (classifier) with its environment. Additionally, for the support of relevant attributes, discovery on a given level and on other levels of hierarchy can be found using different ontologies. These ontologies can be described by different sets of formulas and possibly by different logics. Thus, description of such interaction as well as its support give a good reason for applying fibring logics methods (Gabbay, 1977; Gabbay and Pirri, 1997). Note that in a hierarchical modeling of relevant complex patterns also top-down interactions of higher levels of hierarchy with lower levels should be considered, e.g., if patterns constructed on higher levels are not relevant for the target task the top-down interaction should inform lower levels about the necessity of searching for new patterns. The question of how concept ontologies can be discovered from sensory data remains one of the greatest challenges for many interdisciplinary projects on learning of concepts.

The idea of interactive computing stems from many fields in computer science such as concurrent processes, non-terminating reactive processes (e.g. operating systems), distributed systems, distributed nets and objective programming. It is still in development stage and its foundations are not clarified yet. There are at least two main schools of thought, one pioneered by Peter Wegner and another by Yuri Gurevich (see Goldin et al., 2006). Both schools use the notion of *algorithm* but with a different approach. Wegner's school uses it in the classical Turing's sense, excluding interactive systems from the scope of the notion and introducing persistent Turing machines (PTMs) for formal description of interactive systems. Gurevich's school expands meaning of the notion of algorithm, covering interactive systems and classical algorithms. However, Gurevich claims that the difference is based solely on terminology. For formal descriptions of algorithms, Gurevich introduced abstract state machines (ASMs). ASMs are more powerful than PTMs as they are capable of simulating PTMs, while the opposite is not true. In addition to strings or matrices, ASMs also compute non-constructive inputs as relational structures (finite graphs). PTMs can only compute constructive inputs as strings (or matrices written as strings). There is still no consensus between theoreticians on the statement that interactive systems are more powerful than classical algorithms and cannot be simulated by Turing machines. However, the idea of interactive computing still seems to be appealing from a practical point of view: interaction with or harnessing the external environment is inevitable to capture (and steer) behavior of systems acting in the real world. For unpredictive and uncontrolled environments it is impossible to specify the exact set of input states. In data mining or machine learning, the most common case is when we start searching for patterns or constructing concepts on the basis of sample of objects since the whole universe of objects (data) is not known or it would be impractical to begin with the basis of the whole object universe.

Interactive systems have huge learning potential and are highly adaptive. Interactive agents adapt dynamically and harness their environment in achieving goals. Interacting algorithms can not only learn from experience (which also the classical non-interacting learning algorithms do), they can change themselves during the learning process in response to experience. This property opens up room for a new technology called Wisdom technology (Wistech, Jankowski and Skowron, 2009a,b) and, moreover, for the case of intelligent agents this technology becomes inevitable. Intelligent agents make decisions during dynamic interactions within their environment. To meet this challenge they need to use complex vague concepts. In Wistech, wisdom is a property of algorithms, an adaptive ability of making correct judgments to a satisfactory degree in the face of real-life constraints (e.g., time constraints, see Jankowski and Skowron 2009a,b). These decisions are made on the basis of knowledge of an agent. Thus in Wistech, wisdom is expressed metaphorically by the so called *wisdom equation* (see also Fig. 1):

$$wisdom = knowledge + adaptive\ judgment + interactions.$$

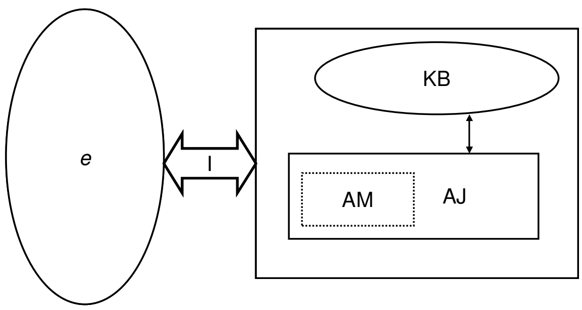

Figure 1. Illustration of the wisdom equation, where AJ denotes adaptive judgment module, AM - action (plan) module, KB - knowledge base module, I - interactions, and e - environment

Adaptive ability means the ability to improve the judgment process quality taking into account agent experience. Adaptation to the environment on the basis of perceived results of interactions and agent knowledge is needed since, e.g., agents make decisions using concepts approximated by classification algorithms (classifiers) and these approximation are changed over time as a result of classifiers acting on variable data or represented knowledge. The wisdom equation suggests also another interaction of higher order: agents making decisions based on ongoing experience, which is particular, apply possessed knowledge, which is general. So making decisions is a kind of logical interaction between general knowledge and particular experience. Vague concepts in this case help cover the gap between generality and particularity while Wistech is required to improve decision making.

Finally, let us mention software platforms supporting the development of our projects, i.e., Interactive Classification Engine (RoughICE) and TunedIT.

RoughICE is a software platform supporting approximation of spatio-temporal complex concepts in a given concept ontology acquired in the dialogue with the user and is freely available at http://www.mimuw.edu.pl/~bazan/roughice. The underlying algorithmic methods, especially for generating reducts and rules, discretization and decomposition, are outgrows of our previous tools such as RSES and RSES-lib project (see http://rsproject.mimuw.edu.pl). RoughICE software and underlying computational methods have been successfully applied in different data mining projects (e.g., in mining traffic and medical data; for details see Bazan, 2008, and references there).

TunedIT platform, launched recently by our research group, facilitates sharing, evaluation and comparison of data-mining and machine-learning algorithms. The resources used in our experiments – algorithms and datasets in particular – will be shared on TunedIT website. This website already contains many publicly available datasets and algorithms, as well as performance data for nearly 100 algorithms tested on numerous datasets - these include the algorithms from Weka, RSESlib, and the datasets from UCI Machine Learning Repository. Everyone can contribute new resources and results. TunedIT is composed of three complementary modules: TunedTester, Repository and Knowledge Base. TunedIT may help researchers design repeatable experiments and generate reproducible results. It may be particularly useful when conducting experiments intended for publication, as reproducibility of experimental results is the essential factor determining the value of a paper. TunedIT helps also in dissemination of new ideas and findings. Every researcher may upload his or her implementations, datasets and documents into the Repository, so that other users can find them easily and employ in their own research.

This paper presents a discussion on extension of RGC to IRGC. Some basic issues toward this goal are discussed. The paper is organized as follows. In Section 2, we discuss a general scheme of interaction. Some generalizations of attributes and information systems to interactive granules used in modeling interactive computations are presented in Section 3. In Conclusions, we summarize the paper contents and we present one of the main research directions for developing IRGC in the Wistech framework.

2. General interaction scheme

In this section, we recall the basic concepts from Skowron and Wasilewski (2010a,b) on the general scheme of interaction. We extend this approach by adding a discussion on computations realized by an agent or team of agents. We use the basic notation from Pawlak and Skowron (2007). In particular, by $Inf_A(x)$ we denote the signature of an object x relative to a set of attributes A, i.e., the set $\{(a, a(x)) : a \in A)\}$.

The global states are defined as pairs $(s_{ag}(t), s_e(t))$, where $s_{ag}(t)$ and $s_e(t)$ are states of a given agent ag and an environment e at time t, respectively. We now explain how the transition relation $\longrightarrow$ between global states are defined in the case of interactive computations. In Fig. 2, the idea of transition from the global state $(s_{ag}(t), s_e(t))$ to the global state $(s_{ag}(t+\Delta), s_e(t+\Delta))$ is illustrated, where Δ is a time necessary for performing the transition, i.e., when $(s_{ag}(t), s_e(t)) \longrightarrow (s_{ag}(t+\Delta), s_e(t+\Delta))$ holds. $A(t)$, $E(t)$ denote a set of attributes available to the agent ag at the moment of time t and a set of attributes used by the environment e at time t, respectively. $Inf_{A(t)}(s_{ag}(t), s_e(t))$ is the signature of $(s_{ag}(t), s_e(t))$ relative to the set of attributes $A(t)$ and $Inf_{E(t)}(s_{ag}(t), s_e(t))$ is the signature of $(s_{ag}(t), s_e(t))$ relative to the set of attributes $E(t)$[1]. These signatures are used as arguments of strategies Sel_Int_{ag}, Sel_Int_e selecting interactions I_{ag} and I_e of the agent ag with the environment e and of the environment e with the agent ag, respectively. $I_{ag} \otimes I_e$ denotes the result of the interaction product $\otimes$ on I_{ag} and I_e. Note that the agent ag can have very incomplete information about I_e as well as about the result $I_{ag} \otimes I_e(s_{ag}(t+\delta), s_e(t+\delta))$, where δ denotes the delay necessary for computing the signatures and selection of interactions (for simplicity, we assume that these delays for ag and e are the same). Hence, information perceived by ag about $s_{ag}(t+\Delta)$ and $s_e(t+\Delta)$ can be very incomplete, too. Usually, ag has only estimations of $s_{ag}(t+\Delta)$ and $s_e(t+\Delta)$ during planning a selection of the interaction I_{ag}. These estimations can next be compared with the perception of the global state $(s_{ag}(t+\Delta), s_e(t+\Delta))$ by means of attributes $A(t+\Delta)$ at time $t+\Delta$. Note that $I_{ag} \otimes I_e$ can change the content of the agent state and of the environment state. Assuming that the current set of attributes $A(t)$ is a part of the agent state $s_{ag}(t)$, this set can be changed, by adding, e.g., new attributes discovered using interactions of I_{ag} with the adaptive judgment module of ag, for example with the help of hierarchical modeling (Skowron and Wasilewski, 2010a,b). Analogously, assuming that the strategy Sel_Int_{ag} is stored in the current state of the agent $s_{ag}(t)$, it can be modified as the result of interaction. In this way, sets of attributes as well as selection of strategies for interactions can be adapted in time.

[1] The fact that we consider only signatures over the set of attributes $E(t)$ reflects one of the basic assumptions of interactive computing, namely that interaction takes place in the environment which can not be controlled. $E(t)$ may not be known to the agent ag.

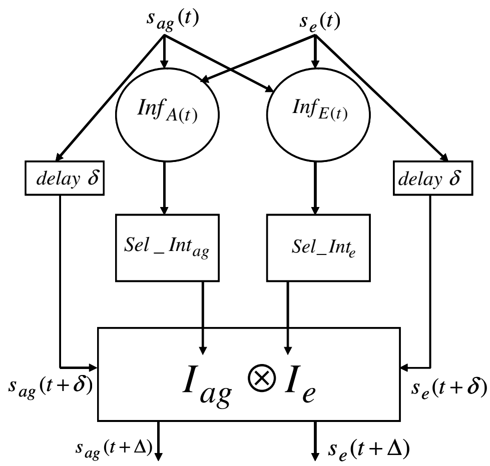

Figure 2. Transition from global state $(s_{ag}(t), s_e(t))$ to global state $(s_{ag}(t + \Delta), s_e(t + \Delta))$

Computations observed by the agent ag using the strategy Sel_Int_{ag} in interaction with the environment e can now be defined using the transition relation $\longrightarrow$ defined on global states and signatures of global states relative to the set of attributes of the agent ag. More formally, any sequence

$$sig_1, \ldots, sig_n, ... \tag{1}$$

is *a computation observed by ag in interaction with e* if and only if for some t, Δ and for any i, sig_i is the signature of a global state $(s_{ag}(t + i\Delta), s_e(t + i\Delta))$ relative to the attribute set $A(t + i\Delta))$ available by ag at a moment of time $t + i\Delta$ and $(s_{ag}(t + i\Delta), s_e(t + i\Delta)) \longrightarrow (s_{ag}(t + (i + 1)\Delta), s_e(t + (i + 1)\Delta))$[2].

Let us assume that a quality criterion Q is given, defined on computations observed by the agent ag, and let sig_1 be a given signature (relative to the agent attributes) characterizing incomplete information of ag about his own state $s_{ag}(t)$ and the environment state $e(t)$. One of the basic problems for the agent ag is to discover selection strategy in such a way that any computation (e.g., with a given length l) observed by ag and starting from any global state with the signature sig_1 and realized using the discovered selection strategy will satisfy the quality criterion Q to a sufficient degree (e.g., the target goal of computation has been reached or that the quality of performance of the agent ag in computation is satisfactory with respect to the quality criterion). The hardness of the selection strategy discovery problem by the agent ag is due to the uncertainty about the finally realized interaction, i.e., the interaction being the result of the interaction product on interactions selected by the agent ag and

[2]As usual one can consider finite and infinite computations.

the environment e. In planning the strategy, the agent ag can use (a partial) information on history of computation stored in the state.

Let us denote by $Comp(ag)$ the set of all computations observed by ag in interaction with e. Then the goal of ag (realized by adaptive judgment) is to induce an adaptive strategy $Strategy$ such that ag using this strategy will realize from $Comp(ag)$ only computations of satisfactory quality relative to Q.

One may attempt to treat this problem as a search for the winning strategy in a game between the agent ag and the environment e with highly unpredictable behavior (Shoham and Leyton-Brown, 2009). However, let us observe that the actions selected by the agent ag are initiated using adaptive judgment about the satisfiability degrees of goals, which are often complex vague concepts. It is also worthy mentioning that these goals drift in time. These are only examples of problems making the game of ag with e different from the standard models of game theory. These games are rather more like games mentioned by Wittgenstein (2001) in his discussion on natural language. Development of foundations for such games is still a great challenge. Hierarchical modeling for inducing patterns relevant for respective approximations is one of the research directions toward this goal (see, e.g., Skowron and Stepaniuk 2010, 2011).

Now, let us discuss how an agent ag can perceive other agents in the environment. The agent ag is perceiving the environment using the signatures in the current state and, possibly, the historical computations. The agent ag identifies some parts of the signatures available at time t as descriptions of other agents in the environment. Next, by hierarchical modeling, the agent ag may discover dynamic properties of these parts as behavioral patterns of agents. Let us consider an illustrative example presented in Fig. 3.

In the original table (upper part of the figure), rows are labeled by time t and $p_1(i), p_2(i), p_3(i)$ denote attribute value vectors describing properties of three agents identified in the environment ag_1, ag_2, ag_3, respectively, at time i. The table shown in the lower part of the figure, presents new structural objects. In this table, paths realized by agents ag_1, ag_2, ag_3 and observed by ag are taken as objects. Attributes in this table describe properties of such structural objects, e.g., different constraints among agents at time i (different moments or intervals of time) or behavioral patterns of agents and their interaction.

Using methods for learning models of concurrent systems from data (Pawlak, 1992; Skowron and Suraj, 1993, 1995) one can consider the problem of interaction structure discovery. This problem is illustrated in Fig. 4. It is assumed that from granules G, G_1, G_2 representing sets of paths of processes realized by agents ag, ag_1, ag_2, their models in the form of Petri nets PN, PN_1, PN_2, respectively, were induced. Then, the structure of interaction between PN_1 and PN_2 can be represented by the simplest transformation of PN_1, PN_2 into PN, e.g., a simple synchronization of Petri nets or a more complex operation.

For modeling interactions, continuous models are widely used in science and engineering. In the case of such models, a natural approach is based on searching for models in the form of differential equations (see, e.g., Bridewell et al.,

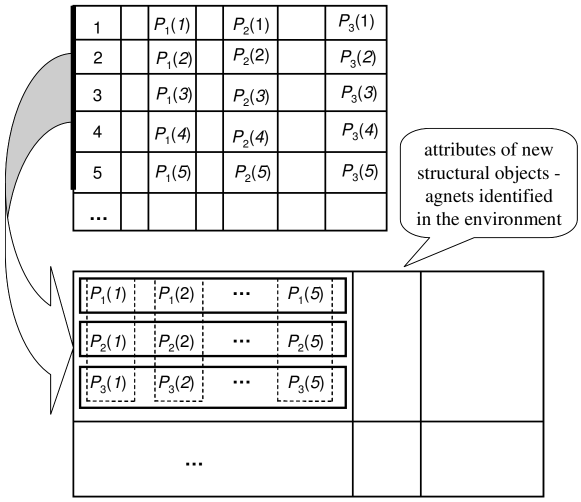

Figure 3. Properties of other agents in the environment identified by ag (dotted lines mark the global states of the observed team of agents, continuous lines mark the paths of each agent)

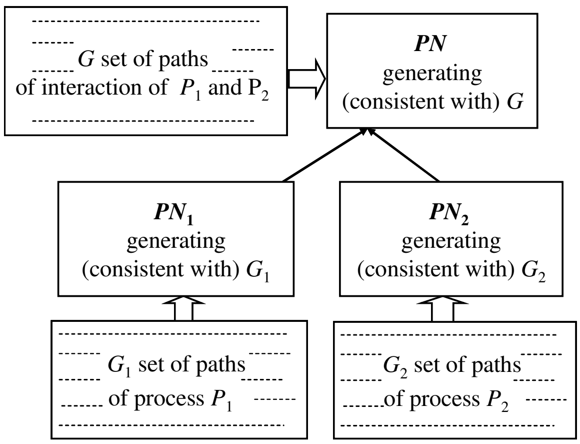

Figure 4. Discovery of interaction structure

2008; Feng et al., 2007). However, for more complex tasks it is hardly possible to give directly the analytical form of the functions describing changes in time of the agent state and the environment state. Instead of that one can search for approximations of these functions from partial and imprecise domain knowledge and experimental data using statistical or/and rough set tools (Ramsay and Silverman, 2002; Nguyen et al., 2010; Skowron and Stepaniuk, 2010, 2011; Bazan, 2008).

Consider an illustrative example of Fig. 5, showing complexity of modeling of interaction under uncertainty. In this figure, actions ac and ac' interact in time. Let us first consider a situation when the two actions are performed in isolation, i.e., without interaction. Then the action ac is initiated, if during the time period T_α, condition α holds. Next, this action is performed for the time period T_{ac}. After the action ac is finished, condition β holds for the time period T_β. Analogous conditions are shown in Fig. 5 for the action ac'. However, when these actions start to interact, then the situation becomes much more complicated. In Fig. 5, the case is illustrated when the initiation of action ac starts before ac'. Then, due to action ac it may happen that condition γ will not hold for the time necessary for initiation of ac', or even when ac' starts, the result of interaction of ac and ac' will cause that neither condition β nor δ will hold after both of these actions are finished. We see that more knowledge is needed to model the result of interaction in this case. Note also that information about the time periods, such as shown in the figure, will be in most cases not exact (crisp) but fuzzy, due to the uncertainty. However, partial and imprecise knowledge often allows us to induce models of interaction processes. These models will be in most cases nondeterministic rather than deterministic. One can expect that such models can be described by, e.g., Petri nets rather than by exact analytical formulae.

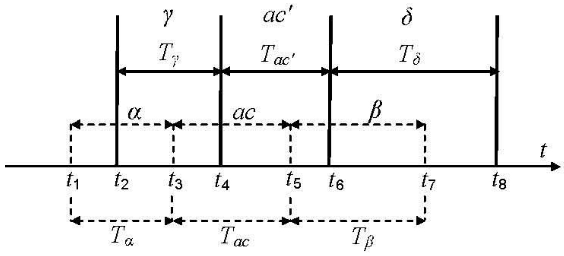

Figure 5. An illustrative example for continuous interaction

3. Generalizations of attributes and information systems: perception attributes, action attributes, information systems, nets of information systems

In this section, we present a generalization of the attribute and information system concepts (Pawlak, 1982, 1991; Pawlak and Skowron, 2007) to interactive granules. This enables using them as basic concepts in modeling interactive computations.

We distinguish perception attributes and and action attributes.

Perception attributes. One of the main forms of interaction of an agent with the environment is perception of the environment by an agent. Moreover, this form is indispensable in the case of interactive systems. Without perception, every action of an agent in the environment would be blind, and the agent would not be able to adapt its behavior to changing conditions of the environment or to modify dynamically its course of action as a response to results of agent's actions in the environment.

In order to represent the results of perception, we need a specific class of attributes: *perception attributes*. The beginning of the perception process is in the senses in the case of living organisms or in sensors in the case of artificial agents. Senses/sensors interact with the environment. To represent the results of this interaction we use *sensory attributes*. These atomic attributes depend solely on interaction with the environment and are independent of other attributes in information system (Skowron and Wasilewski, 2011). Sensory attributes are also open attributes, i.e., if a is a sensory attribute, then a is a function not necessarily onto its value domain V_a. This formal property reflects the fact that sensors interact with the environment which cannot be controlled. It is always possible that new stimuli appear at the senses/sensors which were not perceived before. The value domains of sensory attributes are determined only by sensitivity of sensors represented by these attributes.

In order to describe formally the perception processes as interactions, let us introduce some notation. If $f : X \times Y \longrightarrow X \times Y$, then by $\pi_1[f]$, $\pi_2[f]$ we denote projections of f, i.e., $\pi_1[f] : X \times Y \longrightarrow X$, $\pi_2[f] : X \times Y \longrightarrow Y$ such that $f(x, y) = (\pi_1[f](x, y), \pi_2[f](x, y))$ for $(x, y) \in X \times Y$. Both of the previously introduced interactions, I_{ag} and I_e, can affect the global state of a given agent and its environment. By $I_e(s(t))$ $(I_{ag}(s(t)))$ we denote the global state at time $t + \Delta$ obtained from $s(t)$ by applying I_e (I_{ag}) only. Since both I_{ag} and I_e act over the time Δ they can also dynamically affect each other, the result of such interfering interaction being denoted by $I_{ag} \otimes I_e$.

As we mentioned above, perception is an example of interaction between an agent and its environment. Moreover, it is a very interesting example. It is a kind of action made by an agent, which usually does not affect the environment[3], but in which an agent is affected by its environment. In order to analyze the perception process, we should be more specific and introduce $I_{ag,a}$ - an interaction operation selected by an agent ag for performing measurement of the value of a sensory attribute a. We assume that in $s_{ag}(t)$ values are stored of sensory attributes at time t, i.e., as a part of $s_{ag}(t)$ one can distinguish (a, v), where a is a sensory attribute and v is its value at time t (or information that this value is not available). In the described model, changes of attribute values are recorded in discrete time with step Δ. For a sensory attribute a we have

[3]In the case of quantum level this assumption is not true.

that

$$s(t+\Delta) = (s_{ag}(t+\Delta), s_e(t+\Delta)) = [I_{ag,a} \otimes I_e](s(t)) = \\ = (\pi_1[I_{ag,a} \otimes I_e](s(t)), \pi_2[I_e](s(t))), \quad (2)$$

assuming that: $s_{ag}(t+\Delta)$ differs from $s_{ag}(t)$ only in a part corresponding to attribute a, i.e., a new value of a is equal to the result of sensory measurement by $I_{ag,a}$ (in a more general case $s_{ag}(t)$ may be influenced by I_e) at time $t+\Delta$. Since $[I_{ag,a} \otimes I_e](s(t)) = (\pi_1[I_{ag,a} \otimes I_e](s(t)), \pi_2[I_{ag,a} \otimes I_e](s(t)))$ therefore $\pi_2[I_{ag,a} \otimes I_e](s(t)) = \pi_2[I_e](s(t))$, i.e., $s_e(t+\Delta)$ was changed by I_e but there is no influence of $I_{ag,a}$. In other words $\pi_2[I_{ag,a}](s(t)) = I_e(s_e(t))$, i.e., $s_e(t+\Delta)$, the state of the environment e in time $t+\Delta$ being result of interaction is obtained from $s_e(t)$ by the dynamics of the environment only.

Any sensory attribute a of a given agent ag is represented by an interaction operation $I_{ag,a}$ such that the following properties hold:

1. Changes in the agent state caused by interaction of $I_{ag,a}$ with the environment are defined using a relational structure $\mathcal{R}_a$ and a set of formulas $L_a = \{\alpha_v\}_{v \in V_a}$, where V_a is the set of indexes.

2. The result of interaction of a with the environment is a pair (l, v), where l is a label and $v \in V_a$ is the index of formula α_v selected in interaction. The results of past interactions are stored in an information system $\mathcal{A}_a$ of the attribute a, in which all pairs (l, v) are stored obtained through interactions recorded so far. To different interactions different labels are assigned. Moreover, each new interaction updates the information system $\mathcal{A}_a$ by adding a new pair (l, v) resulting from the current interaction to $\mathcal{A}_a$.

3. For any formula $\alpha_v \in L_a$, its interpretation over $\mathcal{R}_a$, denoted by $\|\alpha_v\|_{\mathcal{R}_a}$, is given. The family $\{\|\alpha_v\|_{\mathcal{R}_a}\}_{v \in V_a}$ is a partition of the domain of $\mathcal{R}_a$. This property establishes bijection between L_a and V_a and allows to store v in $\mathcal{A}_a$ instead of α_v.

4. An attribute a may interact with the environment only if enabled by the agent ag. This assumption allows agent ag to select different sets of attributes for interaction with the environment in different moments of time.

There are also other reasons for adding the relational structure $\mathcal{R}_a$ and the set of formulas L_a to the description of the sensory attribute. We would like to include in the sensory attribute a possibility that after the sensory measurement some initial data preprocessing is performed, like discretization or symbolic value grouping. Then the formulas in L_a describe over $\mathcal{R}_a$ the granulated results of measurements, such as intervals of reals or sets of symbolic values, and the indexes of formulas in L_a denote these granulated objects.

Enabling of sensory attributes is an example of interaction of sensory attributes with the agent possessing these attributes. Sensory attributes also interact with other components of the agent, e.g., during searching for new relevant patterns for approximation of complex concepts in hierarchical learning (Bazan, 2008; Bazan and Skowron, 2005b; Bazan et al., 2006a,b,c).

In Fig. 6, we illustrate basic features of sensory attributes.

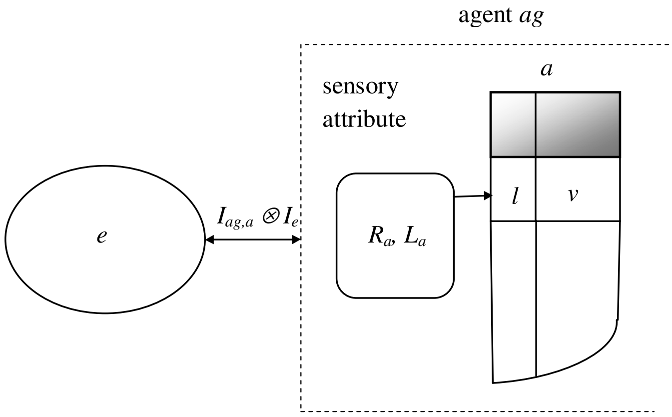

Figure 6. Sensory attribute. e denotes the environment, $\mathcal{R}_a, L_a$ - relational structure of sensory attribute a and set of formulas assigned to a, respectively, l is a label of the environment state currently perceived by a, v is the index such that $\alpha_v \in L_a$ was selected in interaction of a with the environment. In the shadowed area the results of past interaction are stored, the interaction of a with the environment e is not changing e (the changes of e are caused by dynamics of the environment only). In the agent state only a row with label l and v was added and represents the result of sensory attribute a measurement.

In the next steps (of perception), some new attributes can be introduced on the basis of information presented by sensory attributes. These are perception constructible attributes and we refer to them as *complex perception attributes*. They correspond to complex perceptual representations constructed in the process of perception postulated in cognitive science (Thagard, 2005; Bara, 1995; Cartwright, 2000). Complex perception attributes can be used in searching for patterns or structural properties of perceived objects (Skowron and Wasilewski, 2011). They also seem to be indispensable in solving of classification problem for complex concepts in the case of newly perceived objects. Complex perception attributes serve as a kind of bridge between knowledge stored in an agent and results of perception given by sensory attributes. For the same reason, they are needed in approximation of complex vague concepts referring to an environment perceived by a given agent and responsible for activating actions.

Action attributes. Every action is made on the basis of knowledge possessed by an agent, results of agent's perception of the environment, some previously established objectives, and should lead to specified goals. Every goal of an action in the environment should have observable characteristics. In a process of planning, i.e., a selection of the chain of actions (orders to be followed by effectors) leading to an established goal, it is also very important to consider an expected state of the environment as an anticipated result of actions. This result of actions is predicted on the basis of some agent's knowledge and by comparing to observables of specified goals it can be used in selection of an optimal action/plan (a course of actions). In the case of dynamical interactions, the state of the environment anticipated by a given agent can be used for comparison with the state of the environment actually perceived during (or after) the effecting of action in order to modify an action made (or planned) by a given agent, i.e., anticipated state of the environment is an element of feedback mechanism. So, two elements are essentially connected to every action: the goal of an action and the expected state of the environment, matching (possibly partially) the observables of the goal.

In Artificial Intelligence actions are parts of *production rules* called also *IF-THEN* rules. In rough sets such rules are represented by decision rules. Thus, in rough set analysis of interactions, attributes used for representation of actions are decision attributes. We refer to these attributes as *action attributes*. It follows from the discussion above that action attributes should be compound. A value of an action attribute should not only contain information on elementary actions (or a chain of elementary actions) but also should contain information on the specified goal and expected perceptual results of a given action (chain of actions). These attributes can be constructed in many ways. In the process leading to selection of interaction $I_{ag,a}$ by Sel_Int_{ag}, where an action attribute a represents solely a given action/actions, this attribute becomes also condition attribute and is used together with attributes representing knowledge and perception of ag for determining expected observable results in the environment. These anticipated results are compared with observable characteristics of specified goals and decisions on selection of interactions are made on the basis of their similarity or whether anticipated results match sufficiently the observable properties of goals. More advanced approach can use history of interactions for action prediction. Anticipated results of action predicted at time t can be compared to perceived states of the environment at time $t + \Delta$ being a result of interaction $[I_{ag,a} \otimes I_e](s(t))$. This comparison is used to modify an action in time $t + \Delta + \lambda$, where λ is the time needed for making comparison and plan modification, when perceived results are too far away from anticipated ones. The basic features of action attributes are illustrated in Fig. 7.

These notes clearly show that some methods of comparison of anticipated states of the environment with observable characteristics of specified action goals and the perceived states of the environment are needed as indispensable in selection or modification of actions.

The approach discussed allows us to define the semantics of formulas from L_a relative to the set of states $State_e$ of the environment, perceived by a. A state

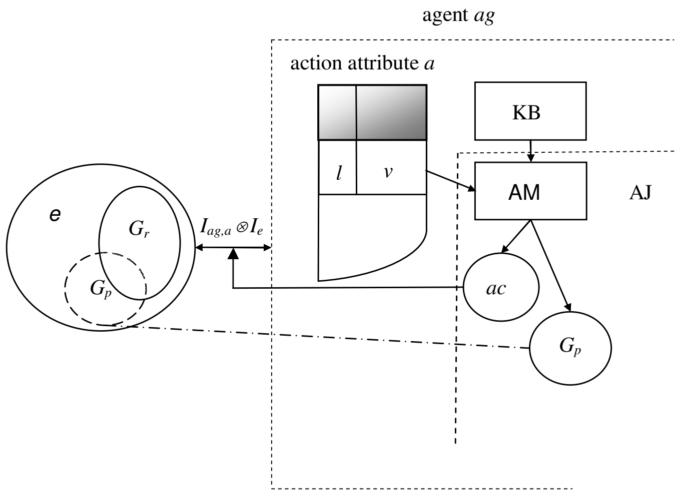

Figure 7. Action attribute. On the basis of current information on current state of agent ag and of the environment e, the action attribute a is selecting an action ac and predicts changes of the environment caused by ac, represented by granule G_p; l, v mean the same as in Fig. 6. AJ denotes the adaptive judgment module with the action submodule denoted AM. The action attribute a selects an action ac to be performed (using module AM, knowledge base KB, and the measurement results stored by sensory attributes). Changes in e caused by ac in the form of granule G_p are predicted, too. The selected action ac determines the interaction $I_{ag,a}$ of agent ag with the environment. Note that reaction of the environment may be unpredictable and granule G_r, representing change of e in effect of $I_{ag,a} \otimes I_e$ may be different from the predicted, described by granule G_p.

$s_e \in State_e$ satisfies $\alpha_v \in L_a$ relative to a, in symbols $s_e \in \|a = v\|_{State_e}$ if and only if the interaction of a with the environment state s_e updates $\mathcal{A}_a$ by a pair (l, v), where l is the label of s_e. Observe that all states from $State_e$ updating in interaction with agent ag (defined by sensory attribute a) the information system $\mathcal{A}_a$ by a pair (l', v), where l' is any label, are indiscernible by the agent ag relative to the sensory attribute a[4]. Using the defined semantics of the descriptor formulas one can define semantics of compound formulas in a standard way (see, e.g., Pawlak, 1982, 1991; Pawlak and Skowron, 2007). This allows us to perform inductive reasoning on sets of objects not yet perceived, e.g. to make an estimation of the inclusion degree between two granules defined by formulas, when their satisfiability is known on a data sample only (Skowron and Stepaniuk, 2010).

[4]This is the reason why we use notation $s_e \in \|a = v\|_{State_e}$ instead of $s_e \in \|\alpha_v\|_{State_e}$.

The main generalization of attributes presented in this section concerns their dynamic structure rather than static, as in the existing approaches, i.e., they interact as special granules with the environment. Let us note that other interactions may be performed with other parts of agent ag possessing the attributes.

Now, we would like to shortly describe a generalization of information systems. Information systems in interactive computations should be dynamic granules rather than static ones. Any sensory attribute a defines dynamic information system $\mathcal{A}_a$ with one attribute, recording the results of interactions of a with the environment. These information systems are updated when new interactions are performed. Such information systems may be fused into a new information system. This fusion may be based on labels of rows of fused data tables representing information systems. For example, assuming that any considered label has as a component the moment of measurement ending, in one row of the new data table rows are included from component data tables labeled by the same time moment. The new information system is updated when information systems from which this system was obtained are updated.

Note also that the fusion of information systems corresponding to sensory attributes may be more complex. So, the fusion may lead to new information systems with structural objects (Skowron and Wasilewski, 2010a,b, 2011; Skowron and Szczuka, 2009) or to nets of information systems linked by different constraints. For example, a family of parameterized sensors may model a situation when the sensors are enabled by a judgment module for recording features of video at different moments of time in probing the environment. This makes possible collecting the necessary features of the environment for activating the relevant higher level action. Parameters may be related, e.g., to positions of camera. This is closely related to the approach from Noë (2005, p. 1):

... perceiving is a way of acting. Perception is not something that happens to us, or in us. It is something we do. Think of blind person tap-tapping his or her way around a cluttered space, perceiving the space by touch, not all at once, but through time, by skillful probing and movement. This is, or at least ought to be, our paradigm of what perceiving is. The world makes itself available to the perceiver through physical movement and interaction.

The last example suggests that the sensory attributes may be fused using some parameters such as time of enabling or position of sensors. Certainly, for more complex actions it is necessary to use a net of such parameterized sensors in which sensory attributes are linked by relevant constraints (Noë, 2005). Hierarchical modeling may also lead to nets of information systems constructed over information systems corresponding to sensory attributes. Nodes in these networks may be linked using different information, like behavioral patterns or local theories induced from information systems in nodes, and their changes when information systems are updated. In the first case the reader may recognize some analogy to theory of information flow (Barwise and Seligman, 1997).

Conclusions

We discussed some basic issues of IRGC including general interaction scheme, dynamic granules such as attributes and information systems. These granules are important components in modeling interactive computations. Foundations of reasoning methods for computations based on interacting granules still belong to challenges. This is, in particular, caused by uncertainty under which agents should perform reasoning about granules in the environment and their interactions. Uncertainty about the results of interactions may be caused by unpredictable dynamics of the environment or by only partial information about it, extracted from interacting granules representing, e.g., structures of objects. In the last case, it is possible to construct nondeterministic soft models rather than exact analytical ones. Intelligent (systems) agents should interact with the environment regarding their goals, which are often complex vague concepts (belonging to ontologies of such concepts or fragments of natural language) as it is done by human experts in decision making. This requires development of new methods of reasoning. For some applications it was already possible to induce approximations of such concepts from data and domain knowledge using hierarchical modeling. Intelligent agents behaving under existing constraints, should also be adaptive for improving their behavior on the way to targets. The reasoning under uncertainty for real-life problems characterized by the above properties is called adaptive judgment in Wistech software(Jankowski and Skowron, 2009a,b; Skowron and Wasilewski, 2010a,b, 2011). These reasons make the adaptive judgment quite different from the existing approaches to reasoning under uncertainty (Jankowski and Skowron, 2009b).

Acknowledgements

We thank the anonymous reviewers for the valuable comments and suggestions, which helped us to improve the presentation.

The research has been partially supported by the grant N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland, the National Centre for Research and Development under grant SP/I/1/77065/10 by the Strategic scientific research and experimental development program "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information", and by the project "Interactive Computations in Layered Granular Networks", realised with Homing Plus programme of Foundation for Polish Science, financed from the European Union, Innovative Economy Programme.

References

Bara, B.G. (1995) *Cognitive Science. A Developmental Approach to the Simulation of the Mind.* Lawrence Erlbaum Associates, Hove.

Barwise, J. and Seligman, J. (1997) *Information Flow: The Logic of Distributed Systems.* Cambridge University Press, Cambridge.

BAZAN, J. (2008) Hierarchical classifiers for complex spatio-temporal concepts, *Transactions on Rough Sets IX*. **LNCS 5390**, Springer.

BAZAN, J., KRUCZEK, P., BAZAN-SOCHA, S., SKOWRON, A. and PIETRZYK, J.J. (2006a) Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: S. Greco, Y. Hato, S. Hirano, M. Inuiguchi, S. Miyamoto, H.S. Nguyen and R. Słowiński, eds., *Proceedings of the 5th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2006), Kobe, Japan*. **LNAI 4259**, Springer, Heidelberg, 418–427.

BAZAN, J., KRUCZEK, P., BAZAN-SOCHA, S., SKOWRON, A. and PIETRZYK, J.J. (2006b) Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: *Proc. of IPMU'2006, Paris, France*, Éditions E.D.K., Paris, 2650–2657.

BAZAN, J., PETERS, J.F. and SKOWRON, A. (2005) Behavioral pattern identification through rough set modelling. In: D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko, and X. Hu, eds., *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005), Regina, Canada, Part II*. **LNAI 3642**, Springer, Heidelberg, 688—697.

BAZAN, J. and SKOWRON, A. (2005a) Classifiers based on approximate reasoning schemes. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron and M. Szczuka, eds., *Monitoring, Security, and Rescue Tasks in Multi-agent Systems (MSRAS'2004)*, Springer, Heidelberg, 191–202.

BAZAN, J. and SKOWRON, A. (2005b) On-line elimination of non-relevant parts of complex objects in behavioral pattern identification. In: S.K. Pal and S. Bandoyopadhay, eds., *Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence (PReMI'05)*, **LNCS 3776**, Springer, Heidelberg, 720-–725.

BAZAN, J., SKOWRON, A. and SWINIARSKI, R. (2006c) Rough sets and vague concept approximation: From sample approximation to adaptive learning. *Transactions on Rough Sets V: Journal Subline*. **LNCS 3100**, Springer, 39–63.

BREIMAN, L. (2001) Statistical modeling: The two cultures. *Statistical Science*, **16**(3), 199–231.

BRIDEWELL, W., LANGLEY, P., TODOROVSKI, L. and DZEROSKI, S. (2008) Inductive process modeling. *Machine Learning*, **71**, 1–32.

CARTWRIGHT, J. (2000) *Evolution and Human Behavior: Darwinian Perspective on Human Nature*. MIT Press.

FENG, J., JOST, J. and MINPING, Q., eds. (2007) *Network: From Biology to Theory*. Springer, Heidelberg.

FREGE, G. (1903) *Grundgesetzen der Arithmetik vol. 2*. Verlag von Hermann Pohle, Jena.

GABBAY, D. (1977) *Fibring Logics*. Oxford University Press.

Gabbay, D. and Pirri, F. (1997) Introduction. In: D. Gabbay, F. Pirri, eds., *Combining Logics. Special Issue of Studia Logica, Studia Logica*, **59**, 1–4.

Gell-Mann, M. (1994) *The Quark and the Jaguar – Adventures in the Simple and the Complex.* Brown and Co.

Goldin, D., Smolka, S. and Wegner, P. (2006) eds. (2006), *Interactive Computation: The New Paradigm.* Springer, Heidelberg.

Hastie, T., Tibshirani, R. and Friedman, J.H. (2008) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, Heidelberg.

Jankowski, J. and Skowron, A. (2009a) Rough Granular Computing in Human-Centric Information Processing. In: K.A. Cyran, S. Kozielski, J.F. Peters, U. Stańczyk and A. Wakulicz-Deja, eds., *Man-machine Interactions.* Springer, Heildelberg, 23–42.

Jankowski, J. and Skowron, A. (2009b) Wisdom Technology: A Rough-Granular Approach. In: M. Marciniak and A. Mykowiecka, eds., *Bolc Festschrift.* **LNCS 5070**, Springer, Heildelberg, 3–41.

Leśniewski, S. (1929) Grundzüge eines neuen Systems der Grundlagen der Mathematik. *Fundamenta Mathematicae*, 14:1–81.

Miikkulainen, R., Bednar, J.A., Choe, Y. and Sirosh, J. (2005) *Computational Maps in the Visual Cortex.* Springer, Heidelberg.

Nguyen, H.S., Bazan, J., Skowron, A. and Nguyen, S.H. (2006) Layered learning for concept synthesis. *Transactions on Rough Sets I: Journal Subline.* **LNCS 3100**, Springer, 187–208.

Nguyen, H.S., Jankowski, A., Peters, J.F., Skowron, A., Stepaniuk, J. and Szczuka, M. (2010) Discovery of Process Models from Data and Domain Knowledge. In: J. T. Yao, ed., *Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation*, pp. 16–47, IGI Global.

Nguyen, S.H., Nguyen, T.T. and Nguyen, H.S. (2005) Rough set approach to sunspot classification. In: D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko and X. Hu, eds., *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005), Regina, Canada, Part II.* **LNAI 3642**, Springer, Heidelberg, 263-–272.

Noë, A. (2005) *Action in Perception.* MIT Press.

Pawlak, Z. (1982) Rough sets. *International Journal of Computing and Information Sciences*, **18**, 341–365.

Pawlak, Z. (1991) *Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers.

Pawlak, Z. (1992) Concurrent versus sequential – the rough sets perspective. *Bulletin of the EATCS*, **48**, 178–190.

Pawlak, Z. and Skowron, A. (2007) Rudiments of rough sets. *Information Science*, **177**, 3–27.

PEDRYCZ, W., SKOWRON, A. and KREINOVICH, V., eds. (2008) *Handbook of Granular Computing*. John Wiley and Sons, Hoboken.

POGGIO, T. and SMALE, S. (2003) The mathematics of learning: Dealing with data. *Notices of the AMS*, **50**(5), 537–544.

POLKOWSKI, L. and SKOWRON, A. (1996) Rough mereology: A new paradigm for approximate reasoning. *Int. J. of Approximate Reasoning*, **15**(4), 333–365.

RAMSAY, J.O. and SILVERMAN, B.W. (2002) *Applied Functional Data Analysis*. Springer, Heidelberg.

SHOHAM, Y. and LEYTON-BROWN, K. (2009) *Multi-agent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press.

SKOWRON, A. and STEPANIUK, J. (2008) Rough sets and granular computing: Toward rough-granular computing. In: W. PEDRYCZ, A. SKOWRON and V. KREINOVICH, eds., *Handbook of Granular Computing*, John Wiley & Sons, 425–448.

SKOWRON, A. and STEPANIUK, J. (2010) Approximation Spaces in Rough–Granular Computing. *Fundamenta Informaticae*, **100**, 141–157.

SKOWRON, A. and STEPANIUK, J. (2011) Rough Granular Computing Based on Approximation Spaces. *Information Sciences*, doi:10.1016/j.ins.2011.08.001.

SKOWRON, A. and SURAJ, Z. (1993) Rough sets and concurrency. *Bulletin of the Polish Academy of Sciences*, **41**, 237-–254.

SKOWRON, A. and SURAJ, Z. (1995) Discovery of concurrent data models from experimental tables: A rough set approach. In: *Proceedings of First International Conference on Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, 288–293.

SKOWRON, A. and SZCZUKA, M. (2009) Toward interactive computations: A rough-granular approach. In: J. Koronacki, S. Wierzchon, Z. Ras and J. Kacprzyk, eds., *Advances in Machine Learning Vol. II, Dedicated to the Memory of Professor Ryszard Michalski*. Springer, Heidelberg, 23–42.

SKOWRON, A. and WASILEWSKI, P. (2010a) Information systems in interactive computing. In: A. WAKULICZ-DEJA, ed., *Proceedings of Decision System Support Conference, Zakopane*, Institute of Informatics, Silesian University (in print).

SKOWRON, A. and WASILEWSKI, P. (2010b) Information systems in modeling interactive computations on granules. In: M. Szczuka, M. Kryszkiewicz, S. Ramanna, R. Jensen and Q. Hu, eds., *Proceedings of the 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2010)*. **LNAI 6086**, Springer, Heildelberg, 730–739.

SKOWRON, A. and WASILEWSKI, P. (2011) Information Systems in Modeling Interactive Computations on Granules. *Theoretical Computer Science*, **412** (42), 5939–5959.

SYCARA, K. (1998) Multi-agent systems. *AI Magazine*, **19**(2), 79–93.

THAGARD, P. (2005) *Mind: Introduction to Cognitive Science* (2nd ed.). MIT Press.

The Rough Set Exploration System (RSES) (no date) `http://logic.mimuw.edu.pl/~rses` (as of 5 September 2011).

The Rough Set Interactive Classification Engine (RoughICE) (no date) `http://www.mimuw.edu.pl/~bazan/roughice` (as of 5 September 2011.

The RSES LIB PROJECT HOMEPAGE (no date) `http://rsproject.mimuw.edu.pl` (as of 5 September 2011.

The TunedIT PLATFORM HOMEPAGE (no date) `http://tunedit.org/`.

VAPNIK, V. (1998) *Statistical Learning Theory.* John Wiley & Sons.

WITTGENSTEIN, L. (2001) *Philosophical Investigations. The German text, with revised English translation. Third edition.* Translated by G.E.M. Anscombe, Blackwell.

ZADEH, L.A. (1965) Fuzzy sets. *Information and Control*, **8**, 333–353.

ZADEH, L.A. (1999) From computing with numbers to computing with words - From manipulation of measurements to manipulation of perceptions. *IEEE Transactions on Circuits and Systems*, **45**, 105–119.

ZADEH, L.A. (2001) A new direction in AI - toward a computational theory of perceptions. *AI Magazine*, **22**(1), 73–84.

ZADEH, L.A. (2006) Keynote lecture at the 5th International Conference on Rough Sets and Current Trends in Computing, RSCTC, Kobe, Japan, November 6-8, 2006.