An Introduction to Perception Based Computing

Andrzej Skowron and Piotr Wasilewski

Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

To Professor Lotfi A. Zadeh on His 90th Birthday

Abstract. We discuss basic notions of Perception Based Computing (PBC). Perception is characterized by sensory measurements and ability to apply them to reason about satisfiability of complex vague concepts used, e.g., as guards for actions or invariants to be preserved by agents. Such reasoning is often referred as adaptive judgment. Vague concepts can be approximated on the basis of sensory attributes rather than defined exactly. Approximations usually need to be induced by using hierarchical modeling. Computations require interactions between granules of different complexity, such as elementary sensory granules, granules representing components of agent states, or complex granules representing classifiers that approximate concepts. We base our approach to interactive computations on generalized information systems and rough sets. We show that such systems can be used for modeling advanced forms of interactions in hierarchical modeling. Unfortunately, discovery of structures for hierarchical modeling is still a challenge. On the other hand, it is often possible to acquire or approximate them from domain knowledge. Given appropriate hierarchical structures, it becomes feasible to perform adaptive judgment, starting from sensory measurements and ending with conclusions about satisfiability degrees of vague target guards. Thus, our main claim is that PBC should enable users (experts, researchers, students) to submit domain knowledge, by means of a dialog. It should be also possible to submit hypotheses about domain knowledge to be checked semi-automatically. PBC should be designed more like laboratories helping users in their research rather than fully automatic data mining or knowledge discovery toolkit. In particular, further progress in understanding visual perception – as a special area of PBC – will be possible, if it becomes more open for cooperation with experts from neuroscience, psychology or cognitive science. In general, we believe that PBC will soon become necessity in many research areas.

Keywords: Rough sets, granular computing, interactive computations, perception based computing, information systems, perception attributes, sensory attributes, action attributes, approximation of complex concepts, ontology approximation.

T.-h. Kim et al. (Eds.): FGIT 2010, LNCS 6485, pp. 12–25, 2010.

1 Introduction

In this paper, we discuss some basic issues of Perception Based Computing (PBC). The issue of perception was studied by many famous researchers following approaches such as structuralism, Gestaltism, ecological optics, or constructivism. Our approach is closer to the one proposed by Professor Lotfi A. Zadeh in Computational Intelligence (see, e.g., [28,29,30,31,32,11]). However, because at the moment we do not see a possibility to define fully semantics of the precisiated language proposed by Lotfi Zadeh we propose another approach. Our approach is more application oriented and is using concept ontologies expressed in natural language or small fragments of natural language (which are a bit beyond of ontologies) specific and relevant for considered application problem. In our approach, rules for reasoning under uncertainty should be adaptively discovered from data and domain knowledge. We consider action-oriented perception [1] driven by actions. Goals of initiated action help with selection of an appropriate perceptual interpretation among many ones attached to given information provided by senses/sensors. We also emphasize the role of interactive computations in PBC. Perception is a specific form of interaction of an agent with its environment (see e.g. [1,23]). Perceiving results of conducted actions is an essential part of feedback mechanism and makes adaptive change of a course of actions possible. The considered approach is wider than studied in the visual perception domain [10]. For example, by analogy to visual perception one can attempt to construct softbots acting in Web on the basis of perception characterized by their sensory measurements together with ability to perform reasoning leading from these measurements to conclusions about satisfiability of complex vague concepts used as guards for actions. The guards can be approximated on the basis of measurements performed by sensory attributes only rather than defined exactly. Satisfiability degrees for guards are results of reasoning called as the adaptive judgment. The approximations are induced using hierarchical modeling. Note that, e.g., injecting domain knowledge into a relational database engine [24] will also require approximate modeling of situation perception by domain experts or system users.

Our approach to PBC is primarily based on generalized information systems, rough sets, and granular computing [13,14,16]. Information systems are treated as dynamic granules used for representing results of interaction of attributes with the environment. Two kinds of attributes are distinguished, namely the perception attributes (including sensory attributes) and the action attributes. Sensory attributes are the basic perception attributes, other perception attributes are constructed on the basis of sensory ones. Actions are activated when their guards (being often complex and vague concepts) hold to a satisfactory degree. We show that information systems can be used for modeling more advanced forms of dynamic interactions in hierarchical modeling. The role of hierarchical interactions is emphasized in modeling of interactive computations. Discovery of structure for hierarchical modeling is still a challenge for visual perception, brain informatics or PBC [17]. However, in some applications it is possible to interactively acquire domain knowledge, e.g., in the form of ontology or a simplified description of the structure (see, e.g., [4,24]). Then, by developing approximation methods, the ontology becomes understandable by the system to a satisfactory degree [4,18]. This means that it becomes feasible to use approximated ontology in order to perform the adaptive judgment reasoning described above.

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 object-oriented programming. It is still in a developing 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 [7]. Both schools use the notion of *an algorithm* but with a different approach. Wegner's school uses it in the classical Turing's sense, excluding interactive systems from the scope of algorithms and introducing persistent Turing machines (PTMs) for formal description of interactive systems. Gurevich's school expands meaning of algorithms, 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 does not hold. In addition to strings or matrices, ASMs compute with non-constructive inputs as relational structures (finite graphs). PTMs can only compute with 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 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 knowledge from experience (which is also done by classical non-interacting learning algorithms), they can change themselves during the learning process in response to experience. This property creates an open space for a new technology called Wisdom technology (Wistech) [8]. It becomes inevitable for the case of intelligent agents, which 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, it is an adaptive ability of making correct judgments to a satisfactory degree in the face of real-life constraints (e.g. time constraints) [8]. These decisions are made on the basis of knowledge possessed by an agent. In Wistech, wisdom is expressed metaphorically by so called *wisdom equation*:

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

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 which are approximated by classification algorithms (classifiers) and these approximations are changed over time as a result of acting on evolving data and 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. Therefore, making decisions itself is a kind of logical interaction between general knowledge and particular experience. Vague concepts in this case help in covering the gap between generality and particularity while Wisdom technology is required to improve decision making.

From the point of view of Wistech PBC, systems should be also interactive with users and domain experts. They should allow users (experts, researchers, or students) for interactions not only for acquiring ontology but also for submitting hypotheses (e.g., about importance or interestingness of some patterns suggested by experts) to be checked by the system. These systems will be more like laboratories helping different experts to make progress in their research rather than performing fully automatic data mining or discovery. Research on visual perception supported by such systems open for cooperation with experts from neuroscience, psychology or cognitive science will become standard. This view is consistent with [5]; See, page 3 of Foreword:

> *Tomorrow, I believe, every biologist will use computer to define their research strategy and specific aims, manage their experiments, collect their results, interpret their data, incorporate the findings of others, disseminate their observations, and extend their experimental observations – through exploratory discovery and modeling – in directions completely unanticipated.*

In our approach, we further combine the-previously mentioned methods based on information systems, rough sets, and granular computing with other soft computing paradigms, such as fuzzy sets or evolutionary computing, as well as data mining and machine learning techniques. Soft computing methods, in particular rough set based methods, are necessary for adaptive approximation of complex vague concepts representing results of agent perception. They are used by agents as, e.g., action guards or invariants which should be preserved by agents. They may be related to highly structural spatio-temporal objects, e.g., functions representing changes of agent states or properties of dynamic processes [19,23].

Information systems play a special role in modeling of interactive computations based on objects called as granules [16]. Granules can be of complex types starting from elementary granules such as indiscernibility or similarity classes [15] to more complex ones such as decision rules, sets of decision rules, classifiers, clusters, time windows or their clusters, sequences of time windows or processes, agents or teams of agents.

In Section 2, we present a general scheme of interactions. The role of attributes and information systems in hierarchical modeling is outlined in Section 3. On the basis of the general scheme of interactions we introduce interactive attributes. In the definition of attribute, two components are important. The first one is defined by a relational structure and the second one is representing a partial information about the results of interactions of the relational structure with the environment. Information systems are used to represent this partial information [23]. We distinguish two kinds of interactive attributes: sensory attributes and action attributes (Section 4). Our approach generalizes the concept of information systems known from the literature, from static information systems [13,14,15] to dynamic ones. It allows us for using information systems in modeling of interactive computations. There is growing research interest in the rough set community on dynamic information systems (see, e.g., [6,9]). Here, we emphasize that embedding this research in the framework of interactive computations is crucial for many real-life applications [8].

In Conclusions, we summarize the presented approach. We also give short information about our current research projects.

2 Interactive Computations

In this section, 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 the 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 Figure 1, 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 the set of attributes available by agent ag at the moment of time t and the set of attributes used by 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 agent ag with the environment and 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 the result $I_{ag} \otimes I_e(s_{ag}(t+\delta), s_e(t+\delta))$ only, where δ denotes the delay necessary for computing the signatures and selection of interactions (for simplicity of reasoning we assume that these delays for a and e are the same). Hence, information perceived by a about $s_{ag}(t+\Delta)$ and $s_e(t+\Delta)$ can be very incomplete too. Usually, the agent ag can predict only estimations of $s_{ag}(t+\Delta)$ and $s_e(t+\Delta)$ during planning selection of the interaction I_{ag}. These predictions can next be compared with the perception of the global state $(s_{ag}(t+\Delta), s_e(t+\Delta))$ by means of attributes

[1] By considering only signatures over some set of attributes $E(t)$ we reflect one of the basic assumptions of interactive computing that interaction takes place in the environment which can not be controlled. $E(t)$ may not be known for an agent ag.

$A(t+\Delta)$. Note that $I_{ag} \otimes I_e$ can change the content of the agent state as well as 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, for example by adding new attributes discovered using I_{ag}, for example with the help of hierarchical modeling discussed previously. Analogously, assuming that the description of the strategy Sel_Int_{ag} is stored in the current state of the agent $s_{ag}(t)$ this strategy can be modified as the result of interaction. In this way, sets of attributes as well as strategies for selecting interactions can be adopted in time.

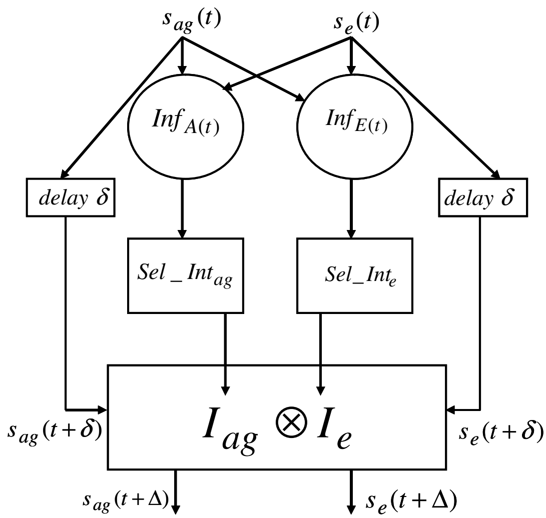

Fig. 1. Transition from global state $(s_{ag}(t), s_e(t))$ to global state $(s_{ag}(t+\Delta), s_e(t+\Delta))$

Computations observed by the agent ag using the strategy Sel_Int_{ag} in interaction with the environment e can now be defined with a help of the transition relation $\longrightarrow$ defined on global states and signatures of global states relative to the set of attributes of agent ag. More formally, any sequence $sig_1, \ldots, sig_n, \ldots$ 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 there is given a quality criterion over a quality measure defined on computations observed by the agent ag and let sig_1 be a given signature (relative to the agent attributes). One of the basic problems for the agent ag is to discover a strategy for selecting interactions (i.e., 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 to a satisfactory

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

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 agent ag and the environment e. In planning the strategy, the agent ag can use (a partial) information on history of computation stored in the state. One may treat the problem as searching for the winning strategy in a game between the agent ag and the environment e with a highly unpredictable behavior.

3 Information Systems in Hierarchical Modeling

A hierarchical modeling of complex patterns (granules) in hierarchical learning (see e.g., [4]) can be described using the rough set approach based on information systems. In such description a construction of every model is described/made on the basis of a particular information system. The result of construction of an information system from a given level of hierarchical modeling is built from information systems from lower levels of its hierarchy. This is made by constructing of new attributes on the basis of already known attributes.

We use the standard rough set notation [13,14,15]. In particular, by $\mathcal{A} = (U, A, \{V_a\}_{a\in A})$, we denote an information system with the set of objects U, the set of attributes A, and the set of attribute values V_a for $a \in A$. By (U, C, d) we denote a decision system with the decision attribute d.

Attributes in information systems can be divided into two classes: *atomic attributes* and *constructible attributes*. Atomic attributes are basic in the sense that their values depend only on some external factors (with respect to a given information system) and are independent on values of other attributes. Atomic attributes can be both closed and open attributes.

Constructible attributes are complex attributes defined from other attributes, or more exactly inductively defined from atomic attributes. If b is a constructible attribute, then for any object x and already defined attributes $a_1, a_2, \cdots, a_m$: $b(x) = F(a_1(x), a_2(x), \cdots, a_m(x))$, where $F : V_{a_1} \times V_{a_2} \times ... \times V_{a_m} \longrightarrow V_b$ and elements of V_b are constructed on the basis of values values from V_i for $i = 1, .., m$.

Sensory attributes can serve as typical examples of atomic attributes. Values of sensory attributes are results of measurement conducted by sensors, so they depend only on the environment and are independent on values of other attributes. Constructible attributes defined from sensory attributes can represent higher-order results of perception, when some patterns are identified, perceptual granules are created *etc.*

We generalize the concept of attributes used in rough set theory [13,14,15,20]. In hierarchical modeling, the attribute value sets can be compound, i.e., they can be of higher order types represented, e.g., in the power set hierarchy [19,20]. The types are fixed in the case of atomic attributes and it should be properly constructed for constructible attributes. Note that elements of the attribute value sets can be complex objects such as signals or images. We also assume that for any attribute a together with its attribute value set V_a there is assigned a relational structure $\mathcal{R}_a = (V_a, \{r_i\}_{i \in I})$, where r_i are relations over Cartesian products of V_a. Examples of relational structures for attributes will be given later. Together with a relational structure $\mathcal{R}_a$ we consider a set of formulas $\mathcal{L}_a$ with interpretation in V_a, i.e., to any formula $\alpha \in \mathcal{L}_a$ there is assigned its meaning $\|\alpha\|_{\mathcal{R}_a} \subseteq V_a$. Moreover, for any attribute a there is distinguished a subset of formulas $L_a \subseteq \mathcal{L}_a$ defining a partition of V_a (defined using the semantics $\| \cdot \|_{\mathcal{R}_a}$). The result of interaction of any atomic attribute with the environment can be described as a selection of formula from L_a. Then values of attribute a, interpreted as the results of measurements by this attribute, can be identified with the index of the selected formula. One can observe that using this approach information systems can be interpreted as the result of a finite number of interactions with the environment of the relational structure defined by the Cartesian product of relational structures corresponding to attributes from information system.

Constructible attributes can be constructed in many ways [23]. One of them is based on introducing a relational structure on value domains of atomic attributes [23]. Because of the space limit we discuss one illustrative example only related to hierarchical modeling. Note that this modeling process is often hierarchical and interactions occur between constructed granules by a given agent on different hierarchical levels and between the levels. Granules are constructed on different levels by means of actions performed on granules from lower levels and then the quality of the granules constructed on higher levels is evaluated. In the case the quality is not satisfactory, new actions on higher levels should be activated in searching for construction of relevant granules. Observe that the concepts initiating actions are often drifting in time. Hence, the searching process (or the discovery process) for relevant granules should be adaptive. Moreover, the satisfiability of the approximated concepts is usually not binary but can be expressed as satisfiability to a degree only. Hence, mechanisms for conflict resolution between different concepts voting for initiating different actions should be developed/learned/discovered (analogously to voting mechanism between rules in the case of rule based classifiers). Another problem is related to propagation of satisfiability degrees of concepts through hierarchical levels. Here, the rough set approach proposes to use approximate reasoning schemes discovered from data or acquired from domain experts [4,18].

General operations on information systems can be defined as products with constraints (see [19] for more details). Definitely, one can consider at the next level of modeling sequences of time windows as structures and construct information systems or decision tables over such structural objects. Observe that in this case the indiscernibility (similarity) classes are sets of paths over time windows. One may, e.g., induce concurrent systems defining such sets of paths [21].

Discovery of relevant attributes on each level of the 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 this concepts and their complements [4]. 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 as well as on other levels of the hierarchy can be found using different ontologies. These ontologies can be described by different sets of formulas and possibly by different logics. 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 the patterns constructed on higher levels are not relevant for the target task the top-down interaction should inform lower levels about necessity of searching for new patterns.

4 Interactive Attributes

There are two basic types of interaction between an agent and an environment: the influence of the environment on an agent and an opposite influence of an agent on its environment.

We need a specific class of attributes to represent interactions of an agent, namely *interactive attributes*, divided into two classes: *perception attributes* and *action attributes*.

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

In order to represent results of perception, we need a specific class of attributes: *perception attributes*. The beginning of the perception process is in 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 from other attributes in information system. Sensory attributes are also open attributes, i.e. if a is a sensory attribute, then a is a function with values in its value domain V_a. This reflects the fact that sensors interact with the environment which can not be controlled. Always it is possible that new stimuli appear to 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 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 $(x, y) \in X \times Y$. A global state at time t, $s(t) = (s_{ag}(t), s_e(t))$, consists of the whole state of a given agent ag and its environment e. Let us recall that the agent ag does not have to posses complete information about these states and usually it does not. In Section 2, by I_{ag} and I_e we denote the influence operation of a given agent ag on its environment e and the opposite influence operation of the environment e on an agent ag, respectively. Both interactions 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 $t+\Delta$ obtained from $s(t)$ by applying I_e (I_{ag}) only. Since both I_{ag} and I_e last in time Δ they can also dynamically affect each other, a result of such interfering interaction is denoted by product $I_{ag} \otimes I_e$.

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 do not affect the environment [3] but in which an agent is affected by its environment. In order to analyze 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 sensory attribute a. We assume that in $s_{ag}(t)$ are stored values 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 timing with Δ. For a sensory attribute a we have that

$$\begin{aligned} s(t+\Delta) &= (s_{ag}(t+\Delta), s_e(t+\Delta)) = [I_{ag,a} \otimes I_e](s(t)) = \\ &= (\pi_1[I_{ag,a} \otimes I_e](s(t)), \pi_2[I_e](s(t))), \end{aligned} \tag{1}$$

assuming that: $s_{ag}(t+\Delta)$ differs from $s_{ag}(t)$ only on 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. In Figure 2, we illustrate the basic features of sensory attributes.

In the next steps (of perception), some new attributes can be introduced on the basis of information presented by sensory attributes in the ways described in Section 3. These are perception constructible attributes and we will refer to them as *complex perception attributes*. They correspond to complex perceptual representations constructed in the process of perception postulated in cognitive science [27], [2]. Complex perception attributes can be used in searching for patterns or structural properties of perceived objects (see Section 3). They also seem to be indispensable in solving 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 environment perceived by a given agent and responsible for activating actions. Therefore complex perception attributes are also indispensable from the point of view of Wistech.

In rough set analysis of interactions, actions are represented by decision attributes. We refer to these attributes as *action attributes*. It follows from the

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

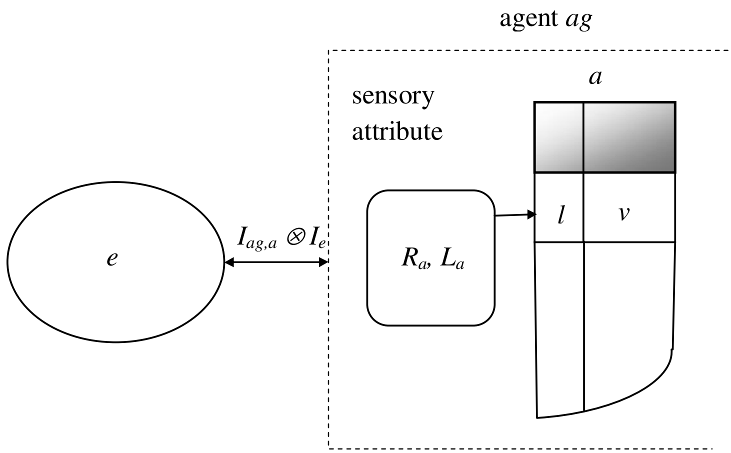

Fig. 2. Sensory attribute. e denotes the environment, $\mathcal{R}_a, L_a$ - relational structure of sensory attribute a and set of formulas assigned to a, respectively, l is a label of the environment state currently perceived by a, v is the index such that $\alpha_v \in L_a$ was selected in interaction 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 only a row with label l and v was added and represents the sensory attribute a measurement.

discussion made above, that action attributes should be somehow compound. A value of an action attribute should not only contain some information about elementary actions (or a chain of elementary actions - this difference is unessential) but also contain information about the specified goal and expected perceptual results of a given action/a chain of actions. These attributes can be constructed in many various ways. In the process Sel_Int_{ag}, i.e., the process leading to a selection of interaction $I_{ag,a}$, where an action attribute a is representing solely a given action/actions. The attribute a becomes also condition attribute and is used together with attributes representing knowledge of the agent ag 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 about selection of interactions are made on the basis of their similarity or whether anticipated results match enough observable properties of goals. More advanced approach can use history of interaction 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 time needed for making comparison and planning modification, in the case when perceived results are too far away from anticipated ones. The basic featutres of action attributes are illustrated in Figure 3.

In [23], an example of highly interactive cognitive architecture ACT-R [26] with a number of interactions between its different parts was discussed.

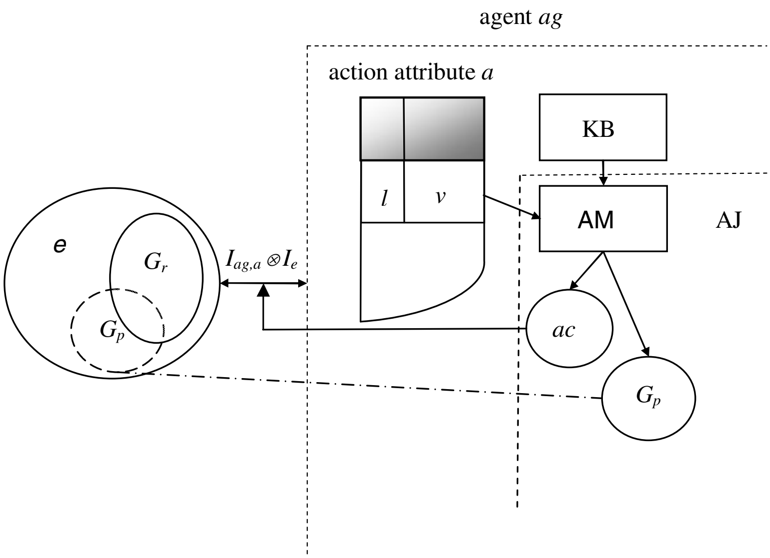

Fig. 3. Action attribute. On the basis of the current information about the current state of the agent ag and the state of the environment e, the action attribute a is selecting an action ac and predicts changes of the environment caused by ac which are represented by granule G_p. l, v have meaning as in Figure 2. AJ denotes the adaptive judgment module with the action submodule denoted by AM. The action attribute a is selecting an action ac to be performed (using module AM, knowledge base KB contents, 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 $I_{ag,a}$ with the environment from the side of the agent ag. Note that reaction of the environment may be unpredictable and the granule G_r representing change of e as the result of $I_{ag,a} \otimes I_e$ (on the component of the environment) may be different from predicted described by granule G_p.

5 Conclusions

We discussed the role of interactive computations in PBC. The fundamental role of information systems in modeling interactive computations was emphasized. Some more advanced representations of sensory measurements such as sets of information systems, clusters of information systems, or relational structures over information systems can be also considered. The role of these more complex structures for adaptive judgment and their relationships to the existing approaches (see, e.g., [3]) will be considered elsewhere.

We stressed the necessity of further development of the adaptive judgment methods for approximate reasoning about constructed granules as a crucial step in understanding of interactive computations. In particular, the necessity of using inductive strategies relevant for new features discovery, extension of approximation spaces, conflict resolution, tuning of quality measures, discovery of approximate reasoning schemes from data and domain knowledge, adaptive judgment based on beliefs, adaptive learning of concepts on the basis of histories of computations for prediction of actions or plans, hierarchy discovery (different levels with features and structural objects), reasoning by analogy, learning protocols for cooperation or competition, coalition formation are all example of tasks in which adaptive judgment is involved.

Acknowledgements. Authors would like to thank Professor Dominik Ślęzak for his critical comments helping to improve the paper. The research has been supported by the grants N N516 077837, N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland.

References

1. Arbib, M.A.: The Metaphorical Brain 2: Neural Networks and Beyond. Willey & Sons, Chichester (1989)
2. Bara, B.G.: Cognitive Science. A Developmental Approach to the Simulation of the Mind. Lawrence Erlbaum Associates, Hove (1995)
3. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press, Cambridge (1997)
4. Bazan, J.: Hierarchical classifiers for complex spatio-temporal concepts. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 474–750. Springer, Heidelberg (2008)
5. Bower, J.M., Bolouri, H. (eds.): Computational Modeling of Genetic and Biochemical Networks. MIT Press, Cambridge (2001)
6. Chakraborty, M.K., Pagliani, P.: Geometry Of Approximation: Rough Set Theory: Logic, Algebra and Topology of Conceptual Patterns. Springer, Heidelberg (2008)
7. Goldin, D., Smolka, S., Wegner, P. (eds.): Interactive Computation: The New Paradigm. Springer, Heidelberg (2006)
8. Jankowski, J., Skowron, A.: Wisdom technology: A Rough-granular approach. In: Marciniak, M., Mykowiecka, A. (eds.) Bolc Festschrift. LNCS, vol. 5070, pp. 3–41. Springer, Heidelberg (2009)
9. Khan, M.A., Banerjee, M.: A study of multiple-source approximation systems. In: Peters, J.F., Skowron, A., Słowiński, R., Lingras, P., Miao, D., Tsumoto, S. (eds.) Rough Sets XII. LNCS, vol. 6190, pp. 46–75. Springer, Heidelberg (2010)
10. Maar, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W.H. Freeman, New York (1982)
11. Mendel, J.M., Wu, D.: Perceptual Computing: Aiding People in Making Subjective Judgments. John Wiley & IEEE Press (2010)
12. Newell, A.: Unified Theories of Cognition. Harvard University Press, Cambridge (1990)
13. Pawlak, Z.: Rough sets. International Journal of Computing and Information Sciences 18, 341–356 (1982)
14. Pawlak, Z.: Rough sets. In: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Science 177, 3–27 (2007); Rough sets: Some extensions. Information Science 177, 28–40 (2007); Rough sets and boolean reasoning. Information Science 177, 41–73 (2007)

16. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, Chichester (2008)
17. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the AMS 50(5), 537–544 (2003)
18. Skowron, A., Stepaniuk, J.: Informational granules and rough-neural computing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neural Computing: Techniques for Computing with Words, pp. 43–84. Springer, Heidelberg (2003)
19. Skowron, A., Stepaniuk, J.: Approximation spaces in rough-granular computing. Fundamenta Informaticae 100, 141–157 (2010)
20. Skowron, A., Stepaniuk, J.: Rough granular computing based on approximation spaces (Extended version of [19] submitted to the special issue of Theoretical Computer Science on Rough-Fuzzy Computing)
21. Skowron, A., Suraj, Z.: Discovery of concurrent data models from experimental tables: A rough set approach. In: Proceedings of First International Conference on Knowledge Discovery and Data Mining, pp. 288–293. AAAI Press, Menlo Park (1995)
22. Skowron, A., Wasilewski, P.: Information systems in modeling interactive computations on granules. In: Szczuka, M. (ed.) RSCTC 2010. LNCS, vol. 6086, pp. 730–739. Springer, Heidelberg (2010)
23. Skowron, A., Wasilewski, P.: Information systems in modeling interactive computations on granules (Extended version of [22] submitted to the special issue of Theoretical Computer Science on Rough-Fuzzy Computing)
24. Ślęzak, D., Toppin, G.: Injecting domain knowledge into a granular database engine – A position paper. In: CIKM 2010, Toronto, Ontario, Canada, October 26-30 (2010)
25. Sun, R.: Prolegomena to Integrating cognitive modeling and social simulation. In: Sun, R. (ed.) From Cognitive Modeling to Social Simulation, pp. 3–26. Cambridge University Press, Cambridge (2006)
26. Taatgen, N., Lebiere, C., Anderson, J.: Modeling paradigms in ACT-R 29. In: Sun, R. (ed.) Cognition and Multi-Agent Interaction. From Cognitive Modeling to Social Simulation, pp. 29–52. Cambridge University Press, Cambridge (2006)
27. Thagard, P.: Mind: Introduction to Cognitive Science, 2nd edn. MIT Press, Cambridge (2005)
28. Zadeh, L.A.: Computing with words and perceptions – A paradigm shift. In: Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2009), Las Vegas, Nevada, USA, IEEE Systems, Man, and Cybernetics Society (2009)
29. Zadeh, L.A.: Generalized theory of uncertainty (GTU) – principal concepts and ideas. Computational Statistics & Data Analysis 51(1), 15–46 (2006)
30. Zadeh, L.A.: Precisiated natural language (PNL). AI Magazine 25(3), 74–91 (2004)
31. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)
32. Zadeh, L.A.: From computing with numbers to computing with words – From manipulation of measurements to manipulation of perceptions. IEEE Transactions on Circuits and Systems 45(1), 105–119 (1999)