**Neuro-symbolic alerting rules**
Velik, Rosemarie; Boley, Harold



**Publisher's version / Version de l'éditeur:**
https://doi.org/10.1109/TIE.2010.2044113
*IEEE Transactions on Industrial Electronics, 57, November 11, pp. 3661-3668, 2010-10-01*

# Neuro-symbolic Alerting Rules

Rosemarie Velik, *Member, IEEE*, Harold Boley

**Abstract**—Future building automation will require complex (human-like) perception and decision-making processes not being feasible with classical approaches. In this paper, we address both the perception and the decision-making process and present an alerting model that reacts to perceived situations in a building with decisions about possible alerts. Perception is based on the neuro-symbolic information processing model, which detects candidate alerts. Integrated with perception, decision-making is based on the rule model of RuleML, which computes alerts to relevant building occupants about current opportunities and risks. A general model of neuro-symbolic alerting rules is developed and exemplified with a use case of building alerts.

**Index Terms**—building automation, neuro-symbolic networks, RuleML, perception, decision-making.

## 1 INTRODUCTION

Building automation has matured over the last decades towards an indispensable contribution to everyday life. Classical approaches in building automation are concerned with simple monitoring of the environment (e.g., temperature), making this information accessible for the user, and adjusting it to predefined value ranges targeting comfort and energy preservation. Alerting and control strategies are based on input data from a small number of uniform sensors. However, future applications will target to "understand" the human users of a building and thus make it a safer, more secure, more comfortable, and more (energy-)efficient place [1], [2]. To do so, buildings will have to be equipped with a large number of diverse sensors, whose information has to be merged in order to get a robust representation of the environment, as studied in *sensor fusion* [3]. However, this research field is still quite recent and existing approaches are challenged by this abundance of data and the ways in which it should be analyzed and responded to [4], [5]. There is thus a need for new concepts to handle future demands [6]. One solution introduced recently to process and interpret such a flood of sensory information is the neuro-symbolic information processing principle [7]. This approach is inspired by how information is processed in the perceptual system of the human brain, which is able to cope with information from millions of diverse sensory receptors. On the other hand, only perceiving the environment is not sufficient. Adequate reactions in form of alerts or the activation of actuators have to be triggered depending on what is currently perceived. This calls for a decision-making process [8] which translates the perception results into an appropriate (re)action selected from a number of rule-encoded possibilities.

In this paper, we address both the perception and the decision-making process and present an alerting model that reacts to perceived situations in a building with decisions about possible alerts. By such alerts, users can be informed about opportunities (e.g., social event taking place on second floor), about safety and security relevant issues (e.g., entrance door left open during night), or they can be encouraged in energy and resource saving behavior (e.g., close windows while heating or air conditioning is on). Perception is based on the neuro-symbolic information processing model. Decision-making employs Rule Markup Language (RuleML) derivation rules, which are used by RuleML reaction rules that can alert a particular occupant or group of occupants about current opportunities and risks. The main decision to be taken is what messages, if any, to send to which users, depending on what is currently happening in the perceived environment. The concepts used will be clarified by means of concrete, easily comprehensible use cases concerning certain situations arising in an office building.

• *R. Velik is with the Biorobotics Department, Fatronik – Tecnalia, Spain and the Institute of Computer Technology, Vienna University of Technology, Austria.*
*E-mail: rvelik@fatronik.com*
• *H. Boley is with the University of New Brunswick, Faculty of Computer Science and the Semantic Web Laboratory at the NRC, Institute for Information Technology.*
*E-mail: harold.boley@nrc-cnrc.gc.ca*

*Manuscript received December 25, 2008. Accepted for publication February 2, 2010.* 

## 2 ALERTING MODEL

In figure 1, an overview of the general alerting model is given. It consists of two main modules: perception followed by decision. The task of the perception module is to classify the situation of a system. Based on the perception results, the decision module infers any alerts about risks or opportunities to be selectively communicated to a human. The alerting model economizes by using the same processing techniques for both risks and opportunities.

The architecture of the *perception module* is inspired by neuro-physiological and neuro-psychological research findings about the perceptual system of the human brain. The perception process uses sensor values from different

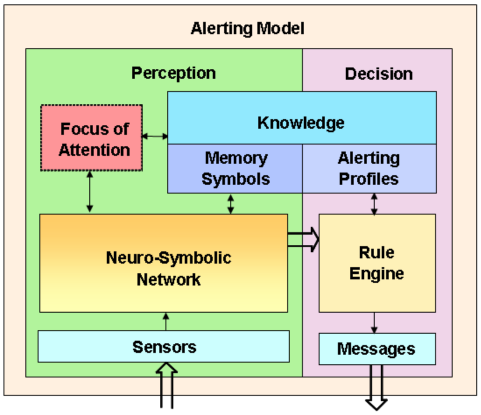


Fig. 1. Overview of Alerting Model

sensory receptors as inputs, which are then processed in a so-called neuro-symbolic network (see section 3.2). A conceptual classification of the environment encoded as activations of neuro-symbols constitutes the perception output. The perception process is additionally supported by mechanisms called memory, knowledge, and focus of attention. We will emphasize here the neuro-symbolic network as well as its interaction with memory and knowledge. For a more detailed description of focus of attention see [7].

The architecture of the *decision module* is inspired by the use of condition-conclusion and condition-action rules in cognitive science. The decision process takes neuro-symbols encoding perceptual results as input, which are then matched by the instantiated premises of applicable rules in a so-called rule engine. The instantiated conclusions of selected rules can trigger other rules etc., leading to a cycle of rule applications for decision-making about alerting actions as output. Rules constitute one 'half' of knowledge representations often complemented by ontological (e.g., taxonomic) knowledge as the other 'half', as well as by (mass) databases. We will write such rules in the POSL (POsitional-SLotted) presentation syntax of RuleML, which are then interchanged on the Web in XML. For an introduction to Web rules and their relation to ontologies see [9]. For an introduction to POSL and the automatic conversion of POSL to XML see [10] and go to www.jdrew.org/oojdrew/demo/translator.

While rule systems could be implemented by neuro-symbolic networks, we will consider the other mapping direction here: Neuro-symbolic networks can be described by rule systems, where each network node becomes a rule with the node inputs as premises and the node outputs as conclusions. This will permit a high-level specification of neuro-symbolic networks and the reuse of the RuleML format for network interchange.

## 3 Use Cases and Model Description

For testing and evaluating the alerting model, two test environments have been studied which provide sensor data based on what is currently going on in an office building. These sensor values are used as input data for the presented model. The first test environment is the kitchen of an office building – the *SmartKitchen* of ICT – which is equipped with different sensors [11]. The second one is a simulator that generates sensor values based on a virtual environment. It was developed to simulate sensor values in order to perceive scenarios in a virtual office environment [12]. The development took place based on the data already obtained from the SmartKitchen environment. The reason for simulating the sensor values is the cost reduction for testing in comparison to real physical installations. For our purposes, the office environment consists of a floor of the office building with a number of offices and one kitchen. The offices all have the same floor plan. The kitchen has a distinct floor plan and is used (in the simulation as well as in the real physical implementation) for informal social interaction, but also for holding official meetings. Both the SmartKitchen and the simulator were already successfully used to test the outcome of a number of related research projects [1], [2]. In the following sections, the concepts of the alerting model are illustrated both in general and by presenting concrete use cases. One use case is the "kitchen party scenario", which is discussed here in detail.

### 3.1 The Kitchen Party Scenario

The kitchen party scenario generically describes a get-together of a number of people in the kitchen not for the purpose of a meeting but as an informal gathering. Such informal gatherings benefit social networking and the quick exchange of ideas. By using the alerting model, the formation of such more or less spontaneous "kitchen parties" can be favored in a catalytic manner. For this purpose, it has to be perceived by the system that there is currently an informal gathering of a core of people in the kitchen. As consequence of this, other employees have to be selectively invited/informed about the opportunity of participating by sending them a message. It is important to precisely select under what circumstance and to whom these messages should be sent in order not to let them inflate to the point of useless spam. Furthermore, it has to be ensured that the system cannot be misused for unauthorized surveillance of employees. This problem can be mitigated on the one hand by only using data of persons that voluntarily participate in the program and on the other hand by not transmitting and logging the perceived data for any other purpose than the evaluation of the performance of the system under test. The perception and decision-making processes according to the alerting model are described in the following first in general and then for the occurrence of the kitchen party scenario. Figure 2 presents the information flow of this use case in a graphical form. A description and a rule based formulation of different parts follow in the next section.

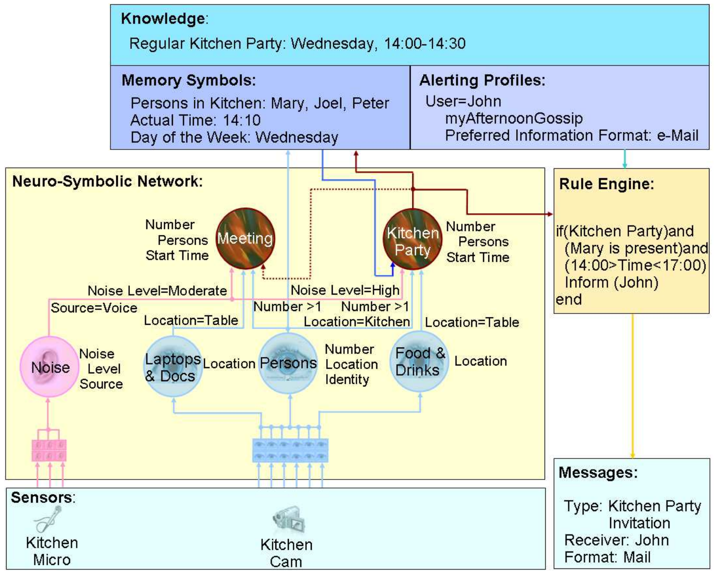


Fig. 2. Use Case Kitchen Party

## 3.2 Neuro-symbolic Network

The first main module of the alerting model is the neuro-symbolic network, which uses so-called neuro-symbols as basic information processing units. The idea for creating neuro-symbols arose from the following consideration: Information in the brain is processed by neurons. However, people do not think in terms of action potentials and firing nerve cells (low level) but in terms of symbols [13] representing, e.g., a face, a person, a melody, or a voice (high level). Neural and symbolic information processing should thus be considered as information processing in the brain at two different levels of abstraction. This raises the question of how the neural and the symbolic level are connected, which could be answered as follows: in the brain, neurons have been found which react, for example, exclusively to the perception of faces [14], [15]. This means that certain neurons or groups of neurons in the brain are responsible for the coding of certain symbolic information. As sketched in figure 3, neuro-symbols combine characteristics of neural and symbolic information processing.

Neuro-symbols represent perceptual images (symbolic information like, e.g., a person, a face, a voice, or a melody) and additionally show a number of analogies to neurons. Each neuro-symbol has a so-called activation grade with a value between 0 and 1 for indicating the degree (probability) to which the perceptual image that

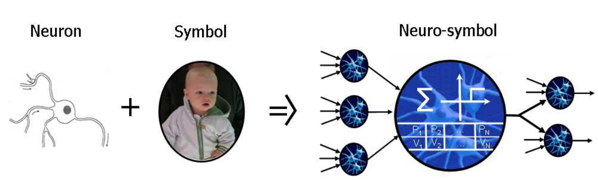


Fig. 3. Function Principle of Neuro-symbols

the neuro-symbol represents is currently present in the environment. A neuro-symbol has a number of inputs and one output. Via the inputs, information about the activation grade of other neuro-symbols is collected, among other things. These activation grades are then summed up and normalized according to the number of inputs to guarantee that the activation grade always has a value between 0 and 1. If this sum exceeds a certain threshold value, the neuro-symbol is activated and the information about its activation is transmitted to other neuro-symbols via its output. Principally also activation functions different from a pure threshold function are possible. For a discussion on how to select the threshold values and other possible neuro-symbolic activation functions see [7]. In a neuro-symbol, not only information can be processed that is received concurrently via the inputs, but also information that is coming in, asynchronously, within a certain time window or in a certain temporal succession, as in Complex Event Processing (CEP) [17]. Inputs can also be weighted differently. The purposes of weights are however different from the use of weights in artificial neural networks, and will be discussed in more detail later on in this section. Besides this, neuro-symbols can carry so-called properties, which specify the neuro-symbol in more detail. Each property can have a range of different values. An example would be the location property, which indicates where in the environment a perceptual image was perceived. The use of properties reflects the principle of population coding according to which related perceptual images are not always represented by separate neurons, but often by a group of neurons [14].

In order to perceive complex situations, neuro-symbols need to be arranged in an architecture to exchange information. To do so, the structural organization of the perceptual system of the human brain is consid-

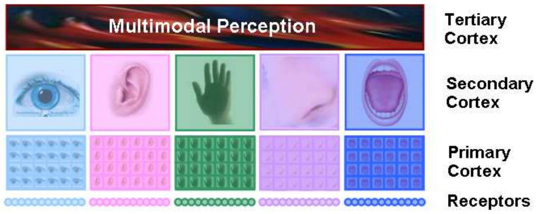


Fig. 4. Structural Organization of Perception

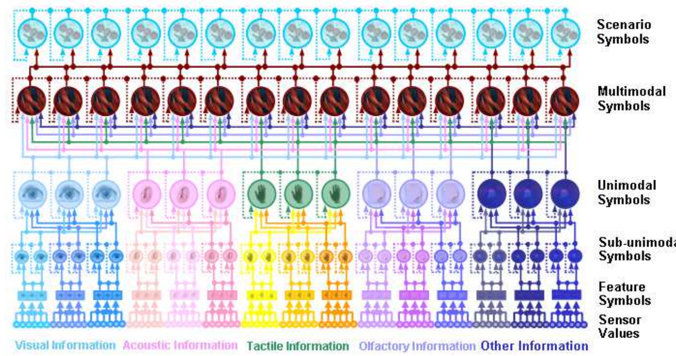


Fig. 5. Modular Hierarchical Arrangement of Neuro-symbols

ered as the archetype [15]. This system is layered as depicted in figure 4. In the brain, the starting point of perception is information from the sensory receptors of the different sense organs. The information coming from these receptors is processed in three stages. The primary cortex is responsible for the first stage, the secondary cortex for the second, and the tertiary cortex for the third one. Each sensory modality has its own region in the primary and secondary cortex. This means that in the first two steps, information of different modalities is processed separately and in parallel. In the tertiary cortex, the information of all modalities is merged and results in a unitary multimodal (modality-neutral) perception of the environment.

The primary cortices have a topographic structure which means that spatially neighbouring receptors of sensory modalities project their information on neighbouring neurons in the primary cortex. Taking the example of the visual system of the human brain, in this first level, neurons would fire to features like edges, lines, colours, movements of a certain velocity and into a certain direction, etc. In the second level, a combination of extracted features results in a quite complex perception of all aspects of the particular modality. For the visual system, perceptual images like faces, a person, or other objects would be perceived. Finally, on the highest level, the perceptual aspects of all modalities are merged.

According to this modular hierarchical organization of the perceptual system of the human brain, neuro-symbols are structured into so-called neuro-symbolic networks (see figure 5). Sensor data are processed in different hierarchical levels into more and more complex neuro-symbols until they result in a multimodal perception of the environment. The neuro-symbols of the different hierarchical levels are labelled according to their function as feature symbols, sub-unimodal symbols, unimodal symbols, multimodal symbols, and scenario symbols.

On the feature symbol level, simple features are extracted from sensory raw data and result in the activation of certain feature symbols, which are topographic in structure can consist of a number of sub-layers. This level corresponds to the primary cortex of the brain. In the next two processing stages, sub-unimodal and unimodal symbols are derived from feature symbols. These two levels correspond to the functions of the secondary cortex of the brain. In fact, each sensory modality can consist of a number of sub-modalities like for example the somatosensory system which consists of the tactile sense, the pain sense, the temperature sense, etc. Similar to this, there can exist a sub-unimodal level between the feature symbol level and the unimodal symbol level. The multimodal and the scenario symbol layer have their analogy in the tertiary cortex of the brain. In the multimodal level, information of all unimodal symbols is combined and merged to multimodal symbols. On the scenario symbol level, different sequences of multimodal symbols can be merged to scenario symbols in order to code longer temporal sequences of events. The multimodal level and the scenario symbol level are the output levels of the perception model and transmit information about what is currently happening in the environment to the decision model.

Concerning the used sensory modalities, there can be sensor types that have an analogy in human sense organs like video cameras and microphones for visual and auditory perception, tactile floor sensors, light barriers, or motion detectors for tactile perception, and chemical sensors for olfactory perception [18]. Additionally, sensor types can be used which have no correspondence to human sense organs like sensors for perceiving electricity or magnetism [19]. By using data from different modalities, a certain degree of redundancy and therefore fault tolerance is achieved. For a discussion about various examples what sensors to use and the appearance of these sensor data see [7].

Neuro-symbols of one level can be considered as "symbol alphabet" for the next higher level. In combination with other neuro-symbols, one and the same neuro-symbol of a certain level can contribute to the activation of different neuro-symbols of the next level. Which sensors trigger which neuro-symbols and which lower-level neuro-symbols activate which neuro-symbols of the next higher level is defined by the connections between them. For this purpose, forward connections and feedback connections are possible. Forward connections always reach from one layer to the next higher layer. Feedback connections exist within a layer and go from outputs of neuro-symbols to inputs of other neuro-symbols of the same level and modality. The connections depicted in figure 5 indicate schematically what connections between neuro-symbols of different levels and modalities are principally possible. In reality, these connections do not exist in such a bus-like form but are point to point connections between particular neuro-symbols.

It was already briefly mentioned that inputs and connections, respectively, can have different weights. Concerning forward connections, weights are always positive as the information being transmitted by them originates from sensor data of perceptual images, which always increase the probability of image detection. The weights correspond to the reliability of a sensory modality. For example, if the visual modality generally provides the most accurate information, inputs from this modality will have a higher weight than inputs from other modalities. When using inputs with different weights, this has of course to be considered when normalizing the sum of all inputs of a neuro-symbol. For feedback connections, weights are negative and can reduce or inhibit the activation of a certain neuro-symbol. That way, undesired activations of neuro-symbols can be suppressed.

Connections between neuro-symbols need not have a fixed structure but can be learned from examples in a supervised learning process. This learning principle also supports the subsequent addition of sensors and neuro-symbols that were not foreseen at the beginning and the removal of neuro-symbols such as in cases where they turn out to be redundant. The major idea of the applied learning algorithm is the following: First, correlations between the sensor values and the feature symbols (and in some cases also the sub-unimodal symbols) are explicitly defined. Correlations between higher levels are then learned stage by stage during a number of learning phases starting with the sub-unimodal level and ending with the scenario symbol level. To learn correlations, examples have to be available that comprise all objects, events, and situations that shall be perceived by a certain modality and level. Examples include input data and target data. Input data are data from sensors that are triggered when certain objects, events, or situations occur in the surroundings. Target data indicate the meaning of the input data. They specify the object, event, or situation that is currently occurring and assign it to a certain neuro-symbol of the current level. For details about the learning methods used we refer to [7] and [16].

In our representation as perception rules, neuro-symbols become predicates defined via a rule conclusion by a rule condition. A rule condition is composed of a conjunction of other such neuro-symbol predicates. These predicates use the grade from (0,1) as their single argument. The rules sum up the grade arguments of the condition predicates and, if the sum exceeds the threshold, assign it to the grade argument of the conclusion predicate. Properties of neuro-symbols become optional extra arguments in the form of key-value slots.

For example, the neuro-symbol for a kitchen party can be represented by a predicate `kitchenParty` defined by POSL rules including the following camera- and microphone-based ones[1]:

```
kitchenParty(?grade;start->?S;num->?N) :-
  kitchenCam(?g1;people->?N),
  greaterThan(?N,1),
  kitchenCam(?g2;food->?F;drinks->?D;
             location->table),
  kitchenMicro(?g3;noise->?V;source->voice),
  isEqual(?V,high),
  add(?grade,?g1,?g2,?g3),
  greaterThan(?grade,THRESHOLD).
```

For the neuro-symbol “kitchen party”, properties are amongst others the number of people participating in the party and the time at which the party started. Input arguments of the neuro-symbol could be the conditions that at least two people are perceived in the kitchen, that food and drinks are present on the kitchen table, and that the noise level of voices is high.

The kitchen party scenario should not be mistaken for a meeting scenario (see also section 3.3). For this purpose, an inhibitory feedback connection can exist from the output of the neuro-symbol “kitchen party” to the neuro-symbol representing the meeting scenario.

```
meeting(?grade;start->?S;num->?N) :-
  kitchenParty(?g1;start->?S;num->?N),
  kitchenCam(?g2;people->?N),
  greaterThan(?N,1),
  kitchenCam(?g3;papers->?P;laptops->?L;
             location->table),
  kitchenMicro(?g4;noise->?V;source->voice),
  isEqual(?V,moderate),
  add(?grade,?g1,?g2,?g3,?g4),
  greaterThan(?grade,THRESHOLD).
```

In our rule representation, neuro-symbolic networks become rule chainings where the conclusion of a rule can be connected to one of the conditions of another rule, etc., in any layered or feedback topology. Each connection is established by the use of the same variable name in the grade arguments of conclusion and condition predicates. The grade variable in a conclusion thus represents the output of its neuro-symbol, and – when equated to the grade variable in a condition – also represents an input of its neuro-symbol. For example, the above rules for the predicates `kitchenParty` illustrate a rule chaining since the conclusion of the `kitchenParty` rule is used as the first of four neuro-symbol conditions of the `meeting` rule. Thus, the output variable `?grade` of `kitchenParty` will be equated to an input `?g1` of `meeting`.

As will be shown in section 3.3, neuro-symbols can not only receive information from sensory receptors or other neuro-symbols, but also from information stored in the knowledge module. To represent this fact, decision rules are used. Decision rules are similar to the rules just mentioned but use n-ary knowledge predicates, which do not represent neuro-symbols, in the conclusion. They

1. The rule is simplified here for presentation purposes. The “`:-`” separates a conclusion from conditions. Arithmetic predicates such as `add` bind their first argument to the result. Predicate arguments can be preceded by a name via a “`->`”. Variables are prefixed by a question mark. `THRESHOLD` is assumed to be a global neuro-symbolic threshold constant.

use condition predicates that are either neuro-symbolic or knowledge predicates in the condition. For example, it could be stored in the knowledge module that the kitchen party scenario is most likely to occur between 2 and 3 pm. In this case, the predicate `kitchenParty` can be defined by a POSL rule whose condition uses, in addition to the neuro-symbolic predicates, the knowledge predicate `inTimeInterval`, for a time between 2 and 3 p.m., as follows:

```
kitchenParty(?grade;start->?S;num->?N) :-
  . . .
  greaterThan(?grade,THRESHOLD),
  inTimeInterval(?S,1400,1500).
```

### 3.3 Memory Symbols, Alerting Profiles, and Knowledge

As described before, based on sensor data from cameras, microphones, etc., the test environment is equipped with, neuro-symbolic information processing is performed in the neuro-symbolic network. This results in the activation of certain neuro-symbols on the different hierarchical levels. Finally, the multimodal and the scenario symbol level provide the output information of the perception module. However, bottom-up sensor data processing alone is not always sufficient to unambiguously perceive what is currently going on in the environment. In the brain, perception is not only based on sensor information but also on stored knowledge and information about what happened before. This knowledge can be factual knowledge, e.g., that objects generally fall down, context knowledge, e.g. that certain objects and events generally occur at certain places or at a certain time of day, or the expectation that certain events or situations are likely to happen after certain other ones. Knowledge can facilitate perception if sensor data are ambiguous or could be assigned to different objects, events, or situations [20]. It also allows a certain degree of fault tolerance [7]. Inspired by this concept, knowledge in the introduced model can influence the activation grade of neuro-symbols. To realize this principle, so-called memory symbols interact with stored knowledge.

Memory symbols have the function to store information about occurring objects, events, and situations or consequences of them that are relevant for future perceptions. Memory symbols receive information from multimodal and scenario symbols, extract and store certain important features of this information, and transmit it to the knowledge module. The knowledge module contains rules defining what influence these former perceptions have on the current ones and can increase, decrease, or inhibit the activation of certain neuro-symbols. Interaction again takes place at the multimodal level and the scenario level.

Memory symbols can be represented as facts storing the current grade of a predicate as its single argument, augmented by possible properties such as a timestamp as its slots. They can be used as condition predicates of rules some of whose other condition predicates (re)invoke a neuro-symbol.

In case of the kitchen party scenario, when considering only sensor data, this scenario could in some cases be mistaken for a meeting scenario. In both cases, a group of people is present in the kitchen. Differences are that in a meeting, it is more likely that people are seated regularly around the table, that they have papers or laptops to read and tools to write with them, that the number of people talking at the same time is smaller, that the overall noise level is lower, and that they have less food and drinks. However, in some situations, the borderline between a kitchen party and a meeting might not be evident to the perception system. In such a case, stored knowledge can help to resolve an ambiguous perception. One example would be that the system knows that on Wednesdays around 2 p.m., certain people generally meet for a break and have coffee and cake together. In this case, influenced by this knowledge, the activation grade of the neuro-symbol representing the kitchen party scenario would be increased and (via feedbacks) the activation grade of the neuro-symbol representing the meeting scenario would be decreased – ideally in a way that the former rises above the threshold value and the latter falls below it.

As already mentioned, properties of the neuro-symbol representing the kitchen party scenario are the number of people participating and the time when the party started. Further properties could be the identity of the persons participating. Neuro-symbols, however, only store information about what is currently perceived in the environment. It might occur that some persons are not identifiable at a certain moment, because they stand with the back to the video cameras and do not talk to be identified by their voice. This problem can be solved by using memory symbols. Memory symbols can store the identity of persons who have entered the kitchen and have been identified until they leave the kitchen again.

Knowledge is an important factor not only on the perception side but also on the decision side. For this purpose, the knowledge module contains general facts and rules necessary for deciding in what cases to inform users about activities going on in the building. The represented facts and rules are relevant for all users of the building or at least a large subset of them.

To enable decisions concerning particular users, so-called *alerting profiles* are employed. The *alerting profile* of each user represents that user with properties representing the various opportunities and risks he/she wants to be alerted about. Alerting profiles are represented in the form of facts and rules, as explored earlier for expert finding [21] and eTourism [22]. These user-specific alerting profiles are employed in the conditions of alerting rules whose other conditions are the currently perceived opportunities and risks in the building. When such derivation rules have decided that a particular user is to be alerted, a top-level reaction rule is triggered to send him/her the actual alerting message.

Based on the perception that a party is currently emerging in the kitchen, the identity of persons participating, the time of day, etc., certain other persons should now be informed about this fact and invited to join. This decision is made in the rule engine, which employs rules and facts from the knowledge module and the alerting profiles. In the knowledge module, general facts are stored, which concern all employees. Concerning the kitchen party scenario, it does not make sense, for instance, to invite further employees to the party if a meeting of the whole office staff will start in a different room in ten minutes. In the alerting profiles, rules and facts for particular users are stored. Such a profile can store, for example, that a user John only wants to be informed about a potential kitchen party if it takes place between 2 and 5 p.m., if the probability that the event really is a kitchen party rather than a meeting is higher than 90%, and if a certain person, Mary, is participating in it. The probability for a kitchen party scenario equals the activation grade of the neuro-symbol representing the kitchen party scenario. In this case, John can have a rule for the predicate `myAfternoonGossip` in his profile.

```
myAfternoonGossip(?grade;start->?S;num->?N) :-
  kitchenParty(?g1;start->?S;num->?N),
  matchOccupant(?g2;room->kitchen;find->Mary),
  add(?grade,?g1,?g2),
  greaterThan(?grade,THRESHOLD)
  greaterThan(?g1,0.9),
  inTimeInterval(?S,1400,1700).
```

For example, a top-level reaction rule in John's profile can specify his alerting preference via emails (rather than pop-up windows, automated mobile phone calls, etc.). Once the above `myAfternoonGossip` predicate is fulfilled, the action of the alerting rule will send John an alerting message about the kitchen party giving its grade, start time and number of participants (including Mary).

At another level of filtering the system has to avoid situational spam by, for instance, not informing users who already participate in the party, who are currently not in their office, or who currently are in another meeting. This can partly be determined by the perception module and partly be derived by the rule engine from stored knowledge, e.g., about the agendas of different users. As a further step towards high-precision alerting, unread alert messages could also be eliminated again from the mail boxes of users in case their validity has expired.

### 3.4 OO jDREW Rule Engine

The decision module is based on the rule interpreter OO jDREW [23] www.jdrew.org/oojdrew, a deductive reasoning engine for the RuleML Web rule language, which also permits the interchange of neuro-symbolic networks of the perception module. OO jDREW can execute positional rules of RuleML both bottom-up and top-down. It also implements object-oriented extensions to RuleML, which include slots (as used here) as well as order-sorted types and object identifiers. OO jDREW is written in the Java programming language and available via Java Webstart and for LGPL download. The alerting rules described here benefitted from earlier implementations of NBBizKB, FindXpRT, and eTourPlan in OO jDREW as well as from the Rule Responder use case for symposium planning [24].

## 4 Neuro-symbolic Alerting versus Artificial Neural Network Approaches

In the alerting model, a neuro-symbolic network was introduced as central processing element of sensor information. Between neural networks and the proposed concept of neuro-symbolic networks, certain similarities can be found. In both cases, weighted input information is summed up and compared to a threshold in the basic processing units. Both combine basic processing units to perform complex tasks and process information in parallel. Nevertheless, besides these similarities, there exist crucial differences between neural networks and neuro-symbolic networks. Unlike in neural networks, where information is represented in a distributed and generally not interpretable form via weights of connections, every single neuro-symbol has a certain interpretable semantic meaning as each neuro-symbol represents a certain perceptual image. Neuro-symbols can contain properties, which specify a perceptual image in more detail. This allows the correct merging of information and offers a mechanism for error detection. In artificial neural networks, only the structure and function of a single nerve cell serves as biological archetype. In contrast to this, in neuro-symbolic networks, also the structural organization of the perceptual system of the human brain is used as archetype for their architecture. Whereas weights in neural networks are altered by a learning algorithm to achieve a mapping of input values to output values, in neuro-symbolic networks, weights are used to consider different reliabilities of sensor modalities. In contrast to neural networks, in neuro-symbolic networks, learning is performed in several steps and phases for different sensory modalities and hierarchical levels. Neuro-symbols comprise mechanisms to process information arriving asynchronously within a certain time window or in a certain succession. Information exchange between neuro-symbols is event-based meaning that information is only processed if a new input message is received. This allows it to reduce the communication and information processing effort.

## 5 Conclusion and Future Work

In this article, we presented an alerting model consisting of a perception module and a decision module in order to inform users about activities going on in a building, which are currently relevant for particular users. Perception is based on the neuro-symbolic perception module being represented by rules. Decision-making is based on the rule model of Derivation RuleML. The test environment for the model was a floor of an office building. The described use cases show that the alerting model has a great variety of potential applications to inform users about opportunities and risks.

The presented approach offers a flexible framework adaptable to very different domains and environments such as homes, offices, and public buildings (stations, airports, stadiums, shopping malls, museums, etc.). Due to its modular hierarchical organization, its overall structure always remains the same, irrespective of the concrete sensor types used, the perceptual images and scenarios to be detected, the stored knowledge about the environment and its users, and the types of alerts.

Besides the kitchen party scenario, which was illustrated in detail here, many other scenarios are conceivable. Examples that are in preparation for future work are in the field of energy end resource saving, safety and security, entertainment, and increase of comfort. Besides for alerting, it is also planned to use the neuro-symbolic rule concept for the control of actuators.

Concerning the perception part of the model, the use of unsupervised learning in addition to supervised learning will be the subject of further investigations. Concerning the decision part of the model, which is currently based on explicitly defined rule profiles, one future research topic will be the investigation of learning the behavior of users to automatically extract rule profiles. Another issue currently studied is the transfer of neuro-psychological and neuro-psychoanalytical findings about emotions and drives to autonomous decision agents.

## REFERENCES

[1] W. Burgstaller, *Interpretation of Situations in Buildings*, PhD thesis at the Vienna University of Technology, 2007.
[2] G. Pratl, *Processing and Symbolization of Ambient Sensor Data*, PhD thesis at the Vienna University of Technology, 2006.
[3] W. Elmenreich, ”A Review on System Architectures for Sensor Fusion Applications,” Software Technologies for Embedded and Ubiquitous Systems, pp. 547-559, Springer Berlin / Heidelberg, 2007.
[4] P. Wide, ”The Electronic Head: A Virtual Quality Instrument” IEEE Trans. Industrial Electronics, vol. 48, no. 4, pp. 766-769 , Aug 2001, doi:10.1109/41.937408.
[5] R. Velik, R. Lang, D. Bruckner, and T. Deutsch, ”Emulating the Perceptual System of the Brain for the Purpose of Sensor Fusion,” Proc. Conference on Human System Interactions, 2008.
[6] R. Velik, G. Zucker, ”Autonomous Perception and Decision Making in Building Automation,” IEEE Trans. on Industrial Electronics, this issue, 2010.
[7] R. Velik, *A Bionic Model for Human-like Machine Perception*, VHS-Verlag, 2008.
[8] P. Vadakkepat, P. Lim, L.C. De Silva, Liu Jing and Li Li Ling, ”Multimodal Approach to Human-Face Detection and Tracking,” IEEE Trans. Industrial Electronics, vol. 55, no. 3, 1385-1393, 2008, doi:10.1109/TIE.2007.903993.
[9] H. Boley, ”Are Your Rules Online? Four Web Rule Essentials,” Proc. Advances in Rule Interchange and Applications, International Symposium, RuleML, pp. 7-24, 2007.
[10] H. Boley, ”POSL: An Integrated Positional-Slotted Language for Semantic Web Knowledge, RuleML Working Draft,” http://www.ruleml.org/submission/ruleml-shortation.html, May, 2004.
[11] S. Goetzinger, *Scenario Recognition based on a Bionic Model for Multi- Level Symbolization*, Master thesis at the Vienna University of Technology, 2006.
[12] H. Hareter, G. Pratl, and D. Bruckner, ”A Simulation and Visualization System for Sensor and Actuator Data Generation,”’ Proc. 6th IFAC International Conference on Fieldbus Systems and their Applications, pp. 56-63, 2005.
[13] R. Velik, ”From Single Neuron-firing to Consciousness - Towards the True Solution of the Binding Problem,” Neuroscience and Biobehavioral Reviews, doi:10.1016/j.neubiorev.2009.11.014, 2009.
[14] E.B. Goldstein, *Wahrnehmungspsychologie*, Spektrum Akademischer Verlag, 2002.
[15] A.R. Luria, *The Working Brain – An Introduction in Neuropsychology*, Basic Books, 1973.
[16] R. Velik and D. Bruckner, ”A Bionic Approach to Dynamic, Multimodal Scene Perception and Interpretation in Buildings,” International Journal of Intelligent Systems and Technologies, vol. 4, no. 1, 2009.
[17] A. Paschke, H. Boley, ”Rules Capturing Events and Reactivity,” Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, May, 2009, ISBN:1-60566-402-2.
[18] T. Bucher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. von Seelen, ”Image Processing and Behavior Planning for Intelligent Vehicles,” IEEE Trans. Industrial Electronics, vol. 50, no. 1, pp. 62-75, Feb 2003, doi:10.1109/TIE.2002.807650.
[19] H.N. Chow and Y. Xu, ”Learning Human Navigational Skill for Smart Wheelchair in a Static Cluttered Route,” IEEE Trans. Industrial Electronics, vol. 53, no. 4, pp. 61350-1361, June 2006, doi:10.1109/TIE.2006.878296.
[20] J.M. Wolfe and K.R. Cave, ”The Psychophysical Evidence for a Binding Problem,” Neuron, vol. 24, pp. 11-17, Sept. 1999.
[21] J. Li, H. Boley, V.C. Bhavsar, and J. Mei, ”Expert Finding for eCollaboration Using FOAF with RuleML Rules,” Proc. Montreal Conference of eTechnologies, pp. 53-65, 2006.
[22] T. Dema, *eTourPlan: A Knowledge-Based Tourist Route and Activity Planner*, Master Thesis, Faculty of Computer Science, University of New Brunswick, September, 2008.
[23] M. Ball, H. Boley, D. Hirtle, J. Mei, and B. Spencer, ”Implementing RuleML Using Schemas, Translators, and Bidirectional Interpreters,” W3C Workshop on Rule Languages for Interoperability, Position Paper, April, 2005.
[24] B.L. Craig and H. Boley, ”Personal Agents in the Rule Responder Architecture,” Proc. Rule Representation, Interchange and Reasoning on the Web, International Symposium, RuleML, pp. 150-165, Oct, 2008.

**Rosemarie Velik** is a senior researcher at the Biorobotics Department of Fatronik - Tecnalia, Spain. Before that, she was assistant professor at the Vienna University of Technology, Institute of Computer Technology. She was honored with the Promotion Sub Auspiciis Praesidentis Rei Publicae due to her first-rate study performances during her whole scholastic career. Her current main research fields are biorobotics, neural and cognitive computation, and ambient intelligence.

**Harold Boley** is adjunct professor at the University of New Brunswick, Faculty of Computer Science and Leader of the Semantic Web Laboratory at the NRC, Institute for Information Technology. His current research interests are in the area of Semantic Web knowledge representation combining rules and ontologies.