Project ARS – The next step towards an intelligent environment

Gerhard Pratl, Peter Palensky
Institute of Computer Technology
Vienna University of Technology
Vienna, Austria
{pratl,palensky}@ict.tuwien.ac.at

Abstract
Project ARS (advanced recognition system) researches the future possibilities for building automation. Psychological models are used to deal with massive amounts of data in order to manage complex scenarios. Such a system would enable a building automation system to detect and comprehend situations that are too complex for existing solutions. This paper describes the motivation for this system, the impressive challenges and the first steps of implementing it.

1. Introduction

Imagine, the development of building automation continues like it did the last one or two decades. Coming from one sensor per home (the thermostat) we are now at hundreds of sensors in automated homes. New office buildings sometimes even have tens of thousands of “datapoints” (sources of or sinks for data) that are integrated in building automation systems. Where will we be in 10 or 50 years?
The current numbers reflect some “natural growth” of the underlying technology. Depending on the price of the systems and on the resulting or expected benefits the buildings are equipped with more or less datapoints. Currently, building automation still has its price, and the “value” of benefits in energy savings, flexibility, security or comfort leads to the above numbers. As the technology will continue to get cheaper, the number of datapoints will steadily increase.
This natural trend might experience a boost, unexpected by most of the involved parties. There is on one hand the need for more nodes – we are talking about hundreds of thousands - and there is on the other hand the possibility to come to these amazing numbers.

1.1 The need for data

Supposing there is a way to run large numbers of datapoints, what would we do with this information? Fine-grained information about buildings cannot be overestimated. Currently we might have for instance one temperature sensor per room, one occupancy sensor per room, etc. Multiple of these sensors can massively increase the quality of the acquired information. Methods like fault tolerance, gracefully degradable systems and sensor fusion only make sense if a large variety of sensors and information is available. Having large numbers of sensors and actuators, new services get in reach: The entire building, including all walls, windows, rooms, devices etc. can for instance be diagnosed and monitored on-line. Depending on the purpose of the building the integrity of the walls, the flow of temperature or other environmental aspects can be measured, logged and analyzed. The status of an entire building can be acquired and evaluated. We expect the possibility to “discover fire before it burns”: if the building is in a state that can lead to a potentially dangerous situation, the building automation system can already initiate an alarm or even take countermeasures. What is needed for this is very much data, collected in time and with a certain degree of quality and reliability.

1.2 Large numbers of sensors

The price of current building automation technology does not allow for having such large numbers of nodes. The price of the nodes is determined by the used sensor type and by the used network technology. Both aspects can and will be optimized and minimized. The traditional way to build high-quality sensors is to take one high-quality (and therefore usually high-price) piece of sensor hardware. The material for this hardware might be the high-price aspect, but it can also be the production process, guaranteed tolerances, calibration costs, seasoning, testing, production cull or additional requirements like a special working temperature of the sensor, achieved by artificially cooling or

heating the sensor hardware. Low cost sensors usually do not fulfill all these tolerances, production rates or accuracies. The trick is to take multiple cheap sensors, preferably diversified ones (based on different technologies or physical effects) and to combine multiple low-quality data to one single high quality one. Add some math, and the pool of unreliable and inaccurate toys merges to one single high-precision device. Having large numbers of such nodes it is tolerable that some of them fail, some of them get inaccurate or some of them never worked at all. It is the large number that helps.
The traditional network technology is still twisted pair cabling, reliable and established. The last years, however, showed a new player in the domain of building networks. Wireless technology, robust and self-healing ad-hoc networks, start to become more and more interesting for building automation. Especially the low-power fraction of the wireless networks is attractive, some of them even working without traditional batteries, getting their power from the environment (solar cells, kinetic energy by pressing a button, etc. [1]). Wireless networks have many advantages compared to the traditional wireline networks. "Cabling" is easy with wireless networks, there is no need for ductwork, the nodes find each other autonomously by forming a scatter network [1]. The communication media "for free" is not the only advantage. The nodes themselves are getting cheaper and cheaper [2], making it possible to get network connectivity in applications where it was previously not affordable.

1.3 Large amounts of data

So we expect the need for large amounts of data, and we see the possibility that we will get these large amounts of data. Typical control applications – think of controlling an engine or a heating system - deal with a manageable number of datapoints, the easiest using just one physical instance that is measured and controlled by one node. The mathematics behind these algorithms is typically based on differential equations, used to optimize the control loops. Having multiple – or in our case numerous – datapoints would very quickly lead this methodology into a level of complexity that is neither solvable from the algorithmic and design aspect nor from the computational aspect. It seems like our traditional way of creating control applications is not capable of dealing with complex systems, where hundreds of thousands of measurement values flood the program that evaluates and monitors the overall system state.
Nature gives an example of a system that is capable of filtering this flood of data, while not overseeing or ignoring any vital measurement value: The human mind. People have the remarkable ability of coping with absurdly large amounts of input data without getting confused. All tactile senses, the visual cortex and other inputs feed out brains with a broad stream of data. Taking into account that the "clock rate" of our brains is – with a maximum of 100Hz for gamma waves – relatively low, it is impressive that we are able to focus on the right things at the right time. Trying to copy the capabilities of human brains (e.g. for a car autopilot or for an automated bank security system that recognize bank robberies) is a challenging task. Millions of interlinked "impressions" have to be processed in a very short time, probably 99% of the measurement data is irrelevant, and there are no strict rules that could easily be transformed into an algorithm.
Human brains are able to cope with complex scenarios, so they could give a blueprint of how to tackle the problem of monitoring and "understanding" situations like the status of a building.
Traditionally, cognitive science and computer science have met in the domain of artificial intelligence. Artificial neural networks, genetic algorithms and other developments are the result of the last decades. Bionic analogies are transformed into mathematics hoping that the artificial construct shows similar behavior. The usual way in this discipline was "bottom up". The insights of neurology were directly implemented in soft- or hardware, hoping that a sufficient amount of artificial neurons would lead to emergent "intelligence".
„On the other side" - the upper side - of the human brain, the PPP scientists (psychology, psychoanalysis and pedagogics) developed their own models of consciousness, some of them contradicting each other, but still useful models. It seems that the various mechanisms found (or modeled) on this „top-level" of the human brain, are one necessary key to build systems that can deal with complex problems. Recent research in PPP – and especially neuro-pathology [3]- has lead to great insight on how the mind might work. The usage of feelings, the balance of Ego, Super-Ego, and Id [4], memory that is gained and lost again, sub- and pre-conscious aspects, and other mechanisms of the human mind are essential ingredients for intelligent behavior [5]. A very interesting phase of the human mind is its early stage in childhood, where consciousness and self-perception is started: a baby

grabs its foot. This cocktail of motorical action and visual and tactical feedback is a massive change of the perception of the entire world: One realizes being part of the world. In general, it is the large number of diversified sensoric input that enables the brain to realize the world.
The project ARS (advanced recognition system) tries to combine a bottom-up with a top-down design method. Neurological methods find their application as well as psychological ones. Bionic approaches are successfully used in mechanics or robotics. Now, building automation borrows some knowledge from nature.

2. Perceptive Consciousness

As described above, today's control engineering has achieved well-established means for controlling processes where a set of inputs is controlled to meet predefined output requirements. Typically this is very limited, compared to nature. A closed-circuit control system considers a number of inputs that is way below the number of sensors that are, for example, found on the skin of a human hand.
In systems as we see them today, every sensor has its dedicated function in an application and is used exclusively for this application (e.g. a temperature sensor to measure the room temperature). Adding an additional sensor for the same functionality is not considered, mainly because of cost reasons. Also it is uncommon to use one sensor for different applications (e.g. for heating control as well as for fire alarms). The advantages are obviously cost savings and a reduced complexity of the control circuitry: if there is only one sensor to report the room temperature, it has to be taken as true; if there were more sensors, each reporting different temperatures, additional logic to determine the most reasonable room temperature is needed.
Disadvantages are, however, manifold. The system has an extremely limited view of the world that it interacts with. An HVAC system (Heating, Ventilation and Air Condition) will perfectly adapt the room climate to meet human requirements, but it will do so no matter whether a person is actually present in the room (it might even try to cool the room in the case of a fire). Failure of a sensor results in malfunction that cannot be compensated by other parts of the system. All system behaviour has to be programmed into it without the system being able to adapt to modified requirements. And even if there were more sensors to provide additional information, it is hard to benefit from this information, due to the previously mentioned increased complexity.
We see the need to extend the existing concept of control systems to the next level, where a system is able to behave in a more sophisticated way, reacting upon personal needs of the user, adapting to a changed environment or requirements and – as the first and most vital step – being able to handle sensory information in the order of many thousands sensors in different domains, providing heavily redundant and maybe contradictorily information.
To handle this vast amount of information we use the model of the biologic mind, as far as it is known today. The goal of *project ARS (Artificial Recognition System)* is to design such a system. In order to manage the complexity of diverse and redundant sensory information, we implement *Perceptive Consciousness* [6] that is able to extract important information and abandon irrelevant information. This is done by using the concept of *symbolization* [7]. In the human mind the term symbolization can be defined as the interface level between the neuron level and the psyche. The symbolization is largely responsible for the way in which we see the real world. This research field was developed by psychologists over the past decade [8][4][9] and the ARS project is based on this work: instead of working with the information that each sensory provides, we introduce a layer that condenses sensor information into symbols. A symbol is a piece of sophisticated information that the system knows and has means to process; in the first layer, which is based purely on sensory information, these symbols are called microsymbols (Fig. 1).

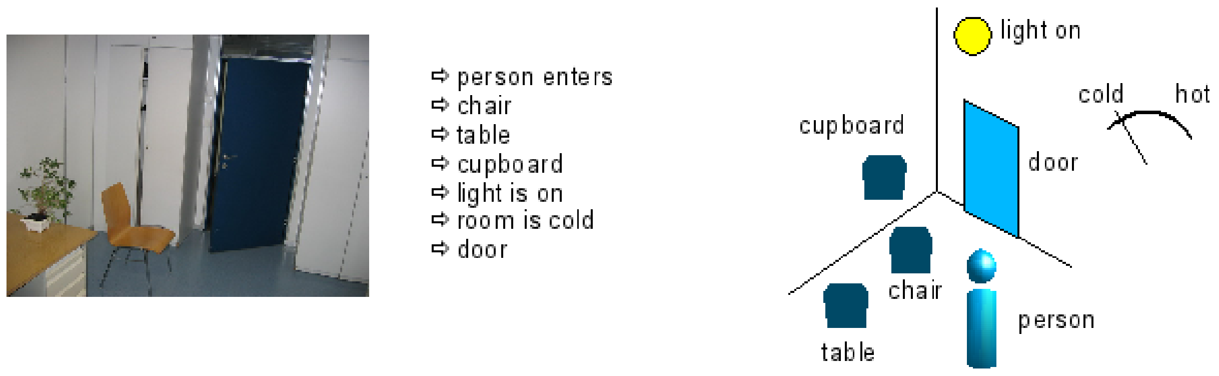

Fig. 1: The real world transferred into the world of microsymbols

They are constructed out of lots of single pieces of sensor information, thus condensing the sensor information. A symbol (or microsymbol) is only little information, but with a lot of "weight", meaning that it represents a more complex concept. As an example we look at the surveillance application introduced later in this paper: Tactile sensors mounted in the floor of a building provide information about people and their movements (a carpet that provides this functionality has been announced in [10]). This information is constantly processed and condensed into microsymbols called *footprint*. The next module of the ARS system is not bothered with processing tactile sensor information, it only receives the information that a new microsymbol footprint has been created together with its whereabouts. As said before, a symbol can be some magnitudes of data volume less than the sensory information that it is based on, but it has more "weight" – assuming that the system is able to process a footprint microsymbol.

Symbolization continues by constructing symbols that are based on microsymbols and even higher-layer symbols that are based on other symbols. A central symbol in the surveillance application is the symbol *person*, which represents a human being with all the properties that are relevant for the functionality of the application. In this case these properties would be the position of the person, the way he or she has taken so far, the identity, and so on. Based on person-symbols, the system can then provide information about the occupancy of rooms, about the activities of persons and their interactions.

Symbols can therefore be *associated* with other symbols (a footprint belongs to a person). These associations are of different kind and build the base for controllable complexity in the system. We will have a closer look on this topic in the next section.

3. Knowledge

To properly fulfil it function, a system has to have knowledge about the information it retrieves from its sensors, it needs to have knowledge about the correlations between the information, and it needs a lot of information that does not originate from sensory input. Human perception of the real world that surrounds him is only based to a small amount on data by his sensors and greatly depends on the interpretation and previous knowledge. The system we implement is of course strongly limited in its capabilities compared to the human mind; it is only able to perceive information that it already knows, learning of new concepts is only planned for the next step. Therefore we define a set of symbols and microsymbols and do not implement means to extend this set (for now). This is one part of the knowledge that the system has: it knows persons and their properties and it knows, for example, how to update the position of a person, if a new footprint has been detected. This knowledge is used for the perceptive awareness module of the system, but there are other types of knowledge that the system implements: according to the human brain [4] we implement a crude form of semantic memory, procedural memory and episodic memory. The semantic memory is described by the definition of the symbols and their associations. Procedural memory contains the actions that the system can take to interact with the real world. Again, these actions are static in the first step, but will adapt to meet new or modified requirements in future versions. Episodic memory provides the system to perceive sequences of events that last over a certain period of time. A recognized episode (or *scenario*, as we also call it) is usually followed by an action of the system. This is due to the fact that we want the system to react to a scenario it has recognized.

The knowledge in the system together with its perception of microsymbols is combined into an image of the real world. A key issue are the associations between what comes from the outside (the perception) and what the system knows (it memory). The system perceives information about changes in the state of the real world and associates this information using its knowledge. The resulting image of the real world is described in the next section.

4. Inner and Outer World

Whatever we think of the real, physical world that surrounds us is strongly based on the information that our sensors (visual, aural, haptic and so on) provide us. Taking this information, we create an image of what we think is around us. The ARS system works similarly: it creates an *inner representation of the outside world*. This inner representation contains all the information about the outside world that is relevant for the system to fulfil its functionality. Other information is disregarded, since it is not important.

Although the ARS system is not embodied by any means, it still has an *inner world*. This is represented by all the components, sensors, actuators and wiring that define the interface between the real world and the perception that the system has of it. Similar to the needs that arise in a mammal when its inner milieu is not in the constrained limits that it needs to survive, the ARS system represents its inner world by gathering information about the state of its components, e.g. broken sensor or interrupted communication to actuators. Of course it does not have any means to repair itself – it can only inform maintenance personnel about its state – still it can react on failures by adapting its perception and using the sensory redundancy that is built in: if the position of a person comes into the system by tactile sensors, light barriers and cameras and a set of tactile sensors fails, the system is still able to determine the position of the person.

5. Architectural Challenges

The system, proposed and implemented in the ARS project, follows a traditional way of engineering science. Behavior and functionality is modeled – in this case for the first time also by using newest results of psychoanalysis and neuropsychology – and then implemented by means of machinery. The task of this machinery is acquisition and processing of environmental data that comes from cameras, microphones and various other sensors.

The first steps in implementing the ideas of ARS are done in a similar traditional way. Acquiring, storing and processing data is the natural task of a database. This tool is sufficient to try out the first functionalities of ARS, to explore “white spots” and to describe the entire system. The usage of a database leads to the usage of standardized data structures, formal languages for specification, to SQL, ODBC, processes and operating systems. It is the home ground of informatics and engineers, built upon common and comprehensible languages. This is fine for describing the functionality, for specifications and for exchanging ideas. The implementation of a data base system that has to process millions of sensor input values per second is, however, another thing.

It is the expected complexity that makes this project special. Sure, the hope to better understand the human mind by having a computational model is still there. We should not forget that the ARS software is 100% transparent and observable by its users and operators. Unlike the human mind, we can have a look at the used symbols, the firing neurons and the flow of data without interfering – an advantage over working with real brains that cannot be overestimated. Especially neuropathology and psychopathology might greatly benefit from this system. Mental disorders like autism could be simulated and provoked with and in the system, parts of its “consciousness” or ability for symbolization could selectively be disabled to examine changes in the perception of the world. But still, it is complexity that makes this project – from the engineering point of view – a special one. The challenges that arise therefrom are manifold.

5.1 Undebuggability of software

Anatomists say that the human brain is the most complex system that we know. It seems natural that we will have to expect a similar level of complexity when we try to “copy” the capabilities of this organ by implementing a machine. Supposing that this machine is a classical computer there are two ways of achieving this level of software complexity:

We have one large program, that is based on heuristic ideas. This program is designed according to some certain idea. The programmer is supposed to know how to do this, how the program should

finally work, what data structures it uses, what procedures and functions, etc. If this way is chosen, we can – according to [11] - expect a piece of software that will “never work right”. The expected size of the code alone will very likely make it undebuggable, untestable and incomprehensible. There were already large software projects (like SDI – Strategic Defense Initiative – in the 80ies) that were not started because people realized that they will be too complex to work in real life.
The second approach could be to build the system upon small and simple pieces of software that are – in its isolated state – testable, debuggable and comprehensible. The expected power and complexity would arise of simply taking thousands or millions of these pieces, that are networked and that can cooperate in order to solve the overall and common given task. An example for this would be a traditional artificial neural network [12], that consists of simple “nodes”, where only the number of these nodes leads to the desired computing power. The verification of such software is not much easier than with the first case, unless you have a formal mathematical proof of the correct scalability (i.e. taking n+1 nodes does not lead to undefined states if n nodes do not have undefined states, etc.) or some means of on-line verification of such a system [13].

5.2 The semantic gap

The semantic gap is a metaphor of computer architecture. It describes the distance between the application program and the hardware. While a program might be based on certain types of data structures (e. g. objects, trees, etc.), the hardware itself just – in the case of a von Neumann machine – uses bitfields without support for higher-level structures. A universal machine can implement any program and even any other machine, but the closer the semantic gap, the more performance you get. Suppose an x86-based PC-style machine, that runs a 68000 emulation. Within this emulation you start a Java virtual machine that executes the final program. Everybody will agree that these 2 nested emulations decrease performance, compared to running the program on a native Java processor. The same happens when an artificial neural network is run on a von Neumann machine. The nature of the ANN is transponded into matrix operations that are then executed on a universal machine. This has nothing to do with the parallel nature of ANNs, the only aspect of real neural networks that is still there in ANNs is the comfortable graph-based model that eases the specification of ANNs. Any algorithm of cognitive science and artificial intelligence that is run on a universal machine leads to a gigantic semantic gap, much larger than those of ordinary programs. The nature and the model these algorithms are based on are far away from how they are implemented and executed.

5.3 The “von Neumann Bottleneck”

Similar to the semantic gap, the von Neumann architecture has a second disadvantage (beside its huge advantage of being universal). Data gets in touch with the central processing unit (CPU) only during the act of computation, which typically happens in a serial manner. All data is stored in some memory (main memory, hard discs, etc.) and sequentially fetched by the processor for processing, also I/O data from “outside” is transported serially over the system bus (Fig. 2). Even the program lays in memory and must be fetched by the processor. Memory and processor are connected – or separated – by the system bus, the “von Neumann bottleneck”.

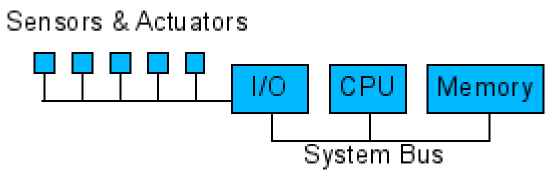

Fig. 2: the von Neumann bottleneck

This separation surely makes things easier but also prohibits possible parallel computation. Real brains on the other hand have no separation of data storage and data processing. Both are combined in neurons, that are in charge of remembering as well as of computing. There is no need to fetch some information from somewhere, it is already “there” – right where it is needed. The ARS architecture has an extreme bottleneck, if all information is stored in a real database. It is not possible to store and retrieve thousands of sensor values per second to and from the database. The injection of sensory data

into the system must happen via parallel paths. The same applies to its storage and computation, the system can only scale to where we want it by means of parallel architectures.
The ARS project is still in the phase where a traditional hardware, combined with a database, is sufficient to discuss and research the various ideas and concepts. In a not-to-far future, however, this architecture will not be able to host and to execute the ARS concept. There will be the need of a two-fold application of parallel structures. The algorithms must be defined in an as parallel-way as possible, and the hardware itself must support parallel streams of data and distributed computation. A simplified view of the dataflow in ARS is given in Fig. 3. Be aware that the number of sensors is expected to be hundreds of thousands or millions and the number of microsymbols might still be tens of thousands.

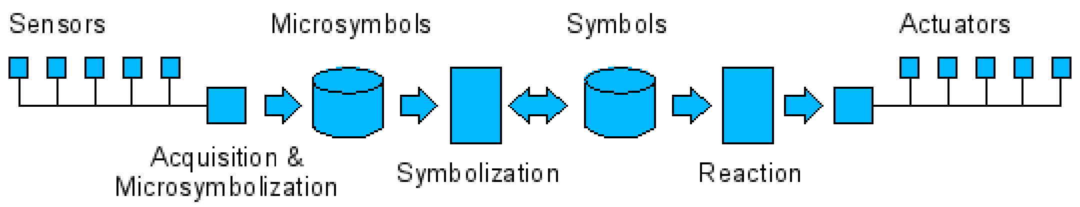

Fig. 3: a simplified software view of the ARS system

Fig. 3 shows the software view of the current implementation, based on field area networks for data acquisition, databases for storing data and programs that condense this data into real information that can be interpreted and that can be reacted on. Note that the real system looks still like Fig. 2, all software, all data is tied and chained to the architectural deficiencies of the von Neumann architecture.

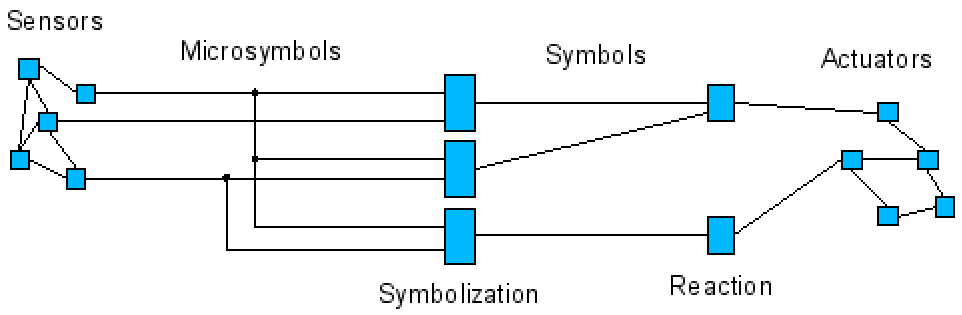

Fig. 4: a dataflow model of the ARS system

Fig. 4 shows the data flow in a more parallel way. There is no separate storage for symbols, the data is merged with its computation. The data flow results that data is in the state of the components or implicitly in the communication between the components [14].Even the microsymbols are created "in the field", by the sensors themselves. Such an architecture means a radical change of traditional data processing systems. The application of hardware implementations of ANNs (digital, analog and hybrid, [15]), the usage of data flow processors [16], [17] and other non-standard hardware can lead to a situation where the software model and the hardware implementation look the same: a closed semantic gap.

6. Applications

In order to demonstrate and test the ARS system we have chosen four applications from different domains. These applications shall be seen as examples of the capabilities of the system; they are prototypes, which can easily be extended to additional functionality.

Application 1: Human Surveillance System

In this application we implement a human surveillance system. By making use of light barriers and detectors, pressure sensors in the floor, door contacts, and stereo cameras the system is enabled to know where people are in a building. People are considered to be anonymous, which means that the system has no additional knowledge about their identity. That is to say, unless a person is furthermore provided with an identification mechanism (e.g. due to authentication at a security door). The system is able to provide information about a person's current and past location, so that the path of a person through a building can be tracked and monitored.

Application 2: Child Safety System
The second application we consider is a child safety system. The system recognizes when a particular person is actually a child, and can monitor and guard the actions of the child. When it appears that the safety of the child may be compromised due to a hazardous situation, the system alerts a (human) supervisor.

The decision that a particular person is actually a child is based on diverse sensor information, similar to other mechanisms in the system. This includes the use of camera images that enable the decision regarding the height and shape of a person, as well as information from light barriers that are mounted at different heights, and weight information obtained from pressure sensors to support the decision. Example situations that are classified as hazardous are: an open fire, a hot stove plate, a cupboard with open doors, or a child climbing on table. There are additional conditions and criteria that have to be taken into consideration. For example, a situation is only classified as hazardous if a child is alone and unattended (meaning that no adult is in the same vicinity). Other conditions include the facts that the fire has to be burning, or that the stove is indeed hot.

Application 3: Geriatric Care System
The third application is concerned with a geriatric system to care for elderly people. In this case the system is able to recognize when an elderly person collapses or faints. Such a collapse shall be recognized. If the supervised person is asleep, two possibly dangerous situations shall be recognized: first, if the room temperature drops down to an alarming level (e.g. because someone has forgotten to close the window in winter), and second, if there is a burning candle in the room.
As additional functionality for improved comfort the system shall identify and track the location of predefined objects, such as keys, glasses, and books.

Application 4: Theft Protection
The fourth application shall supervise the whereabouts of classified things. In a given room the system monitors a set of object (e.g. books in a library) that may not be removed. In case a person takes such an object away, the system shall inform a (human) supervisor.

These applications share a common scenario, in the sense that we define a number of rooms on a floor that has a layout, which is identical for all applications. The sensors that are used are also identical, and mounted in the same position. In this way the symbolization mechanism shares a common set of symbols (although not all symbols have to be present in all applications).

7. System Description

The implementation of the ARS system is based on a database-backed, modular design, in which different components can run on different machines in order to provide sufficient performance where necessary. Since the vast amount of sensors would require considerable installation costs, we use a simulator that creates a virtual world (a floor in a building), in which the sensors and actuators interact with the persons that use the building. The simulator creates the persons and animates them, and it derives the sensor response to the changes in the virtual world. Additionally to the simulated world, we have a real-world installation with a limited set of sensors. This is intended to provide us with information about how to adapt the simulation to be as close as possible to the real world.
All sensor information is stored in a database. This way the ARS system has access to not only the current data, but also to the history of data. The symbols and microsymbols that the system creates are as well stored in the database. This is the task of the perceptive consciousness part of the system: it analyzes the incoming sensor data and creates microsymbols in the database. An association engine is then responsible for the higher layer associations. It resolves questions like: Which footprint belongs to which person? Who is the person that is shown on the camera image and how does this camera image update the position of the person?
An additional knowledgebase contains the necessary information about the outside world and helps to fill in the gaps that are not perceived by the sensor subsystem. For example, if a light barrier is

interrupted, the system assumes that a person has passed by. It has however no knowledge of the size of the person. Therefore the knowledgebase contains a default value for the size of an average person. If the same person is later seen on a camera image, the system can update this default size by the actual size of the person.
Obviously the system has only a limited understanding of the world it perceives. The fact that cameras are installed does not automatically imply that the system is able to process all the information in a way that is similar to a human operator observing and evaluating a camera image. For example, suppose that a dog enters the room. This could possible be perceived as a "person" (or, at best, as a "child"), since the system has no initial concept of a dog. Hence, the system is bound to make incorrect decisions if it is confronted with facts or images that are outside the scope of its capabilities. This system attribute is intentional, since it does not form part of the task that needs to be fulfilled. If we introduce a new application, which makes it necessary to distinguish animals from persons, the knowledge of the system will have to be extended. The ARS system is not intended to run as a standalone system responsible for making final decisions, it is rather intended as support for human supervisor: if the system is able to recognize dangerous situations for a child with a reliability of 80% and then request a human supervisor to make the final decision, and if it is able to fulfil this task for a huge group of children simultaneously, we see a good chance for the system to be accepted.

8. Conclusion

Project ARS is in an early stage, where the foundations of its general architecture are laid. A team of 10 scientists is working on a feasible way to implement the ideas of ARS in a real computer system. The architectural and technological challenges are – as shown – significant. One important aspect that will require consideration is “learning”. Established technologies like databases, ANNs or GAs (genetic algorithms) can be used to implement learning, but learning is much more than “remembering things”. The connection between memories and especially forgetting memories are essential features of a real mind.
The broad stream of input data forces the design to follow a distributed way. Multiple input streams, distributed processing and parallel computation are necessary in order to get the desired computation power.
All in all, the entire proposed system can be seen as a black box. It is fed with massive amounts of data and outputs an analysis or reactions. What happens inside can actually be described as an algorithm, so it seems to be nothing exciting. The true power of the ARS system lies in its model of how to perceive the environment and of how to deal with information. An analogon to this black box metaphor is “Fuzzy Logic”. A fuzzy computation can entirely be described as an algebraic or algorithmic process, as a trivial sequence of mathematical operations. The power of fuzzy logic is that after “fuzzyfication” a mathematically complex situation becomes very easy. It is transformed into a simplified world, where decisions can easily be done. “Defuzzyfication” brings these decisions back to the real world, and - “magically” - fuzzy logic solves complex problems with simple models and decisions. The same happens within ARS: A complex and potentially incomprehensible situation is transformed into a psychological world, where scenarios, images, feelings and other information and mechanisms are easier to comprehend.
The application of ARS to building automation, however, leads to further challenges. The acquisition of massive amounts of data requires new concepts in networking. Sophisticated network management, self-healing structures and low-cost infrastructure are just three of many important aspects. This is why the project ARS is now split into two teams, namely ARS PA (for psycho analysis), dealing with the upper layers, and ARS PC (for perceptive consciousness), dealing with the lower layers that are described in this paper.
The research project is expected to take several years, the involvement of industrial partners guarantees regular intermediate results that can be commercially exploited. Further research and publication will be done, especially on the ARS PA branch and implementation issues.

References

[1] S. Mahlknecht, ”Energy-Self-Sufficient Wireless Sensor Networks for Home and Building Environment”, Ph.D. Thesis, Vienna University of Technology, 2004

[2] Official Zigbee Homepage: www.zigbee.org page visited on 2005-02-11

[3] O. Sacks, “The Man Who Mistook His Wife For a Hat”, Summit Books/Simon & Schuster, Inc., New York, 1987

[4] M. Solms, O. Turnbull, “The Brain and the Inner World: An introduction to the neuroscience of subjective experience”, Karnac Press, 2002

[5] E. Brainin, D. Dietrich, P. Palensky, C. Rösener, “Neuro-bionic Architecture of Automation Systems - Obstacles and Challenges”, in Proceedings of the 7th IEEE Africon Conference, ISBN 0780386051, 2004

[6] C. Tamarit Fuertes, “Automation System Perception”, Ph. D. Thesis, Vienna University of Technology, 2003

[7] G. Russ, “Situation Dependent Behavior in Building Automation”, PhD Thesis, Vienna University of Technology, 2003

[8] K. Kaplan-Solms, M. Solms, “Clinical Studies in Neuro-Psychoanalysis”, International Universities Press, Inc., Madison, CT, 2000

[9] A. R. Damasio, “Descartes' Error. Emotion, Reason and the Human Brain”, G. P. Putnam's Son, New York, 1994

[10] Vorwerk & Co. Teppichwerke, “Zukunft braucht Visionen“, http://www.vorwerk-teppich.de/sc/vorwerk/template/Thinking_Carpet_deu.html, visited on 2005-02-09

[11] C. Cherniak, “Undebuggability and cognitive science”, in Communications of the ACM, Volume 31 Issue 4, April 1988

[12] D. W. Patterson, “Artificial Neural Networks: Theory and Applications”, Prentice-Hall, 1996

[13] J. Schumann, P. Gupta, and S. Nelson, “On Verification & Validation of Neural Network Based Controllers”, in Proceedings of Engineering Applications of Neural Networks EANN 03, September 8-10, 2003, Costa del Sol, Spain, 2003

[14] R. Moore, B. Klauer, K. Waldschmidt, “What computer architecture can learn from computational intelligence-and vice versa”, 23rd EUROMICRO Conference '97 New Frontiers of Information Technology, Budapest, Hungary, 1997

[15] T. Schonauer, A. Jahnke, U. Roth, H. Klar, „Digital Neurohardware: Principles and Perspectives“, Proceedings of Neuronale Netze in der Anwendung NN'98, 101-106,Magdeburg, Germany, 1998

[16] A. H. Veen, “Dataflow machine architecture” ACM Computing Surveys (CSUR), Volume 18 Issue 4, December 1986

[17] L. Bic, R. L. Hartmann, “AGM: a dataflow database machine”, ACM Transactions on Database Systems (TODS), Volume 14 Issue 1, March 1989