# Proceedings of the $2^{nd}$ Workshop on

# Artificial Intelligence Techniques for Ambient Intelligence

# (AITAmI'07)

Hyderabad, India

6-7 of January, 2007

Edited by


Juan Carlos Augusto (University of Ulster)

and

Daniel Shapiro (Applied Reactivity Inc.)


(Event co-located with IJCAI'07)

# Artificial Intelligence Techniques for Ambient Intelligence (AITAmI07)

Ambient intelligence [1] (AmI) is rising as an AI-based paradigm with one of the highest potentials to impact daily human life in the near future. The broad idea is to enrich a space (e.g., a room, house, building, bus station, or a critical area in a hospital) with sensors so that the people using that space can benefit from a more flexible and intelligent environment.

Ambient Intelligence offers many expected benefits. For example, it can increase safety by monitoring lifestyle patterns or recent activities and providing assistance when a potentially harmful situation arises. It can increase comfort by making resources available in advance of demand, or by automatically managing multiple parameters of the environment such as temperature, lighting, and music.

The most well-known area of Ambient Intelligence is usually referred as "Smart Homes", in which a house is equipped to bring advanced services to its users. Recent applications include the use of smart homes to provide a safe environment for people with special needs. For example in the case of elderly people suffering from Alzheimer's disease, a system can minimize risks and ensure appropriate care at critical times by monitoring activities, diagnosing situations and advising human care-givers as available/required. Applications of this form can greatly improve quality of life.

Other applications of this concept include the use of Ambient Intelligence technology to diagnose situations where safety can be compromised in areas such as hospitals, airports, buses, trains, and underground stations. The ability to rapidly detect and react to hazards in such public environments is extremely important for diminishing the negative effects of any crisis. These environments share the characteristic of being enclosed spaces, which tend to support the necessary instrumentation. However, security minded applications can also be developed for open areas, such as parks, and individual streets.

Ambient intelligence is a multidisciplinary area that encompasses a wide range of technical and practical challenges. It includes research in agent design and distributed intelligence, broad issues in user interfaces, special problems in networking and data communication, plus design considerations involved in embedding smart systems in public spaces and buildings. This workshop emphasizes the impact that Artificial Intelligence can have on Ambient Intelligence, specifically to impact the range of services Ambient Intelligence systems can offer and to increase the flexibility such applications can provide.

Although there have been a number of workshops and conferences on related topics, none have made a strong connection between the achievements of Artificial Intelligence and the potential to improve Ambient Intelligence by borrowing/adapting solutions from this rich area.

The topic of Ambient Intelligence is strongly related to traditional AI topics such as spatio-temporal reasoning, robotics, HCI, and to knowledge representation, inference, planning and execution, which are essential in any AI application. Rather than select any single of these threads, the workshop aims to provide a broad perspective on the possibilities that AI has to make smart spaces smarter. This can include work in: Spatio-Temporal Reasoning, Causal Reasoning, Planning, Learning / Data Mining, Case-based Reasoning, Decision Making under Uncertainty, Decision Trees, Neural Networks, and Multi-Agent Systems. This list should not be regarded as exclusive, as other, newer areas are emerging that are relevant to both AI and AmI: Human Interaction with Autonomous Systems in Complex Environments, Self-adaptive systems, Context Awareness, Innovative applications of AI to Ambient Intelligence, and Agent-based approaches to AmI.

A first Workshop on Artificial Intelligence Techniques for Ambient Intelligence [2] was held in August 2006 at Riva del Garda, Italy, bringing together professionals from various European research centres. Now we are very pleased to offer this compilation of papers from the 2nd Workshop on Artificial Intelligence Techniques for Ambient Intelligence which reflects advances in the area made in Asia, Europe and North America. It contains contributions made by researchers and practitioners, and includes two key notes, and fourteen papers which have been selected by our Program Committee. All these contributions report on interesting advances in the area both at a theoretical and application level. We want to thank the authors for their contribution to this exciting event. Special thanks to our supportive Program Committee members who gave shape to the workshop by helping with the selection of papers and by providing useful advise to the authors of accepted papers on how to improve their final versions. Dr. Arne Schuldt and Dr. Christos Goumopoulos also made valuable contributions during the review process.

We would like to give our special thanks to IJCAI, represented by Dr. Carles Sierra, who allowed us to organize AITAmI'07 as a co-located event with IJCAI'07. We would also like to thank the local organization of IJCAI'07 for their technical assistance to produce this proceedings and to organize various aspects of the event at Hyderabad.

We are looking forward to meet you all in a fascinating place of the world for a stimulating event.

## References

[1] *The European Union report, Scenarios for Ambient Intelligence in 2010*, available at `ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf`

[2] Proceedings of the 1st Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI06). Co-located event of ECAI 2006. Riva del Garda, Italy. Edited by Juan C. Augusto and Daniel Shapiro. 29th August 2006.

*JC. Augusto and D. Shapiro*
*January 2007*

# Organization

## Program Chairs

Dr. Juan Carlos Augusto, University of Ulster, U.K.
Dr. Daniel Shapiro, Applied Reactivity, Inc., U.S.A.

## Program Committee

Boris de Ruyter (Philips Research, The Netherlands)
Marc Böhlen (State University of NY, USA)
Andreas Butz (University of Munich, Germany)
Vic Callaghan (University of Essex, UK)
Jingde Cheng (Saitama University, Japan)
Carlo Combi (University of Verona, Italy)
Diane Cook (Washington State University, USA)
Anind Dey (Carnegie Mellon University, USA)
Monica Divitini (Norwegian University of Science)
Christer Fernstrom (Xerox Research)
Michael Freed (NASA-Ames, USA)
Björn Gottfried (University of Bremen, Germany)
Hans Guesgen (University of Auckland, New Zealand)
Pertti Huuskonen (Nokia - NRC Tampere, Finland)
Achilles Kameas (Computer Tech. Institute, Greece)
Henry Kautz (U. of Washington, USA)
John Krumm (Microsoft Research, USA)
Jianhua Ma (Hosei University, Japan)
Johan Plomp (VTT Electronics)
Hartmut Raffler (Siemens AG)
Abdul Sattar (Griffith University, Australia)
Mukundan Sasikumar (CDAC Mumbai, India)
Kostas Stathis (Royal Holloway - U. of London, UK)

# Table of Contents

# AI Techniques for Ambient Intelligence

# Keynotes

# Help from a Stranger – Media Arts in the context of Ambient Intelligence*

**Marc Böhlen**
MediaRobotics Lab
Department of Media Study, University at Buffalo
marcbohlen@acm.org

## Abstract

*Because of the proximity between technically engaged media art practices and Ambient Intelligence research agendas, media arts can deliver contributions to Ambient Intelligence research in ways that Ambient Intelligence as a technical discipline do not.*

## 1 Introduction

Ambient Intelligence (AmI) is likely the most ambitious and fare reaching version of Mark Weiser's [Weiser, 1991] original vision of ubiquitous computing. AmI combines almost all of the subfields of the 'anywhere-anything-anytime' computing paradigm: ubiquitous computing itself, unobtrusive hardware, ad hoc networking, intelligent systems, robust and secure systems, context awareness, intelligent and intuitive interfaces [Shadbolt, 2003] and, with renewed emphasis, social interactions.
If AmI is not foremost about technology, but rather about people as a telling quote from Philips AmI Research states "ultimately however, Ambient Intelligence is not about technology but about people, because it is not Ambient Intelligence that will shape the future of ordinary people, it is ordinary people who will shape the future of Ambient Intelligence" [Philips, 1999], then 'people studies' need to be reconsidered for more effective AmI. This means reconsidering some of the fundamental concepts upon which AmI is contingent.

AmI is historically part of an ongoing discourse on our relationship with technology. Born from the Industrial Revolution's promise for a life of plenty and leisure, technology was become synonymous with progress and improving life. Information processing technologies, including AmI, are part of this larger narrative, firmly embedded in the history of Western culture. The German existentialist philosopher Martin Heidegger, seeking a fundamental understanding of technology as a civilizational phenomenon, understood the essence of technology as a 'Gestell' (standing reserve) [Heidegger 1957]. By this he meant a potential, a prepared and efficient framework onto which a more complex process can be mapped. While this is useful for making life easier, (in not having to grind one's wheat every day but instead buying bread from the store), it prevents people from participating in the complete cycle of events (planting, growing, and harvesting, etc). So there is a 'price' to pay for the removal of tedious labor. Reacting to this dilemma is not without precedent. Cybernetics [Pickering, 2002], [Pias, 2004], and in particular the Second-order Cybernetics movement generated one of the most notable attempts of integrating technology into daily life and putting technology in the service of people and society as a whole. It is also well known that the movement failed from the perspective of the engineering sciences, because it could not make good on its promises, because the simple feedback mechanisms would not scale to complex systems, and because it could not build an operational technical framework, amongst other reasons. The failure of cybernetics and the rise of robotics and AI, and in their wake ubiquitous computing and AmI as technical practices capable of delivering real world solutions, do not render Heidegger's critique nor the Cybernetics movement obsolete. On the contrary, the need to reflect on the role of technology in society and its impact on our environment is as pressing today as it ever was.

## 2 Help from a Stranger

Since the engineering sciences are generally not the domain where knowledge about 'people' is a priority, realms from the social sciences are called upon for assistance. Indeed, Human Computer Interaction (HCI), Sociology, Ethnography all have their established place in this endeavor. However, I am interested in a different source of contribution, a strange and unlikely candidate; the arts, and in particular the field where artists fiddle with technology, commonly known as media arts. Specifically, I am interested in discussing qualitative aspects of mediation between people and ambient systems that are currently not part of the AmI framework.

### 2.1 Media Arts

Media arts are a new and evolving field that is far from unified. The observations below are intended only as a cursory introduction for AmI professionals.

In general one could say that media arts comprise all art forms where technical media come into play. Media arts, as many traditional art forms, tend to be craft-based and thinking often occurs more through things than through abstractions of things. Because of the novelty and the lack of a canon, media arts are a highly experimental field of inquiry. Artists are traditionally not accountable to disciplinary boundaries in the same way as other professionals are. Artists 'experiment' with technologies, often learning while doing, and apply technologies in unusual ways; sometimes in ways for which the technologies were not conceived (and ill prepared to perform). Furthermore, media arts inherit from a lineage of 20th century art movements a sharp focus on personal subjective experience in lieu of universal truth. A common trait of many media art works is an elaborate concern with the effects technologies have on people in their daily lives. As such, media arts are nominally not very different from AmI. However, there are differences. Media arts often suffer from lack of technical sophistication. From an engineering perspective, many of the works are hobbyist hacks. Also, as the sub discipline of media arts, 'physical computing', continues to demonstrate, there is a notable emphasis and rich variety in creating customized outputs from computational systems. However, the computational processes themselves are often not part of the inquiry. Coding is usually performed in scripting languages, data acquisition and processing as well as motor control are reduced to minimum complexity, and the projects are often too fragile to be reliably reproduced.

On the other hand, and this is of importance here, the media arts have a differentiated approach in defining gray scales in the relationship between people and technology, in the circumstances under which we experience technologies, and in the ways in which a given technology may receive new agency. There is a predisposition to contextualize the work outside of the art practice where it originates. This generates, for example, an appreciation for the mutual dependencies between technical and aesthetic design. For media artists, technology is a not about task completion, and interaction does not limit itself to information processing. The concept of a 'user' does not exist in the media arts, and few artists are inclined to perform formal evaluation methods or 'user studies' on their works. Artists are often more inclined to create problems than to effectively solve problems. Contrary to its reputation, media arts are not a pleasure industry. A few examples may shed some light on this.

### 2.1.1 Examples

Arthur Gansen's “Machine With Chair” [Fig. 1] is an elaborate mechanical contraption that does nothing more than rotate a wooden chair through 360 degrees in free space and place it back on the same spot from which it was taken. Gansen's work is part of a history in creating non-utilitarian devices and strange objects that can be used for absolutely nothing. Gansen's work is generally perceived as beautiful in the traditional sense of aesthetics. It is indeed strangely wonderful to watch a mundane object being moved elegantly through space for no apparent reason other than to see it nicely executed.

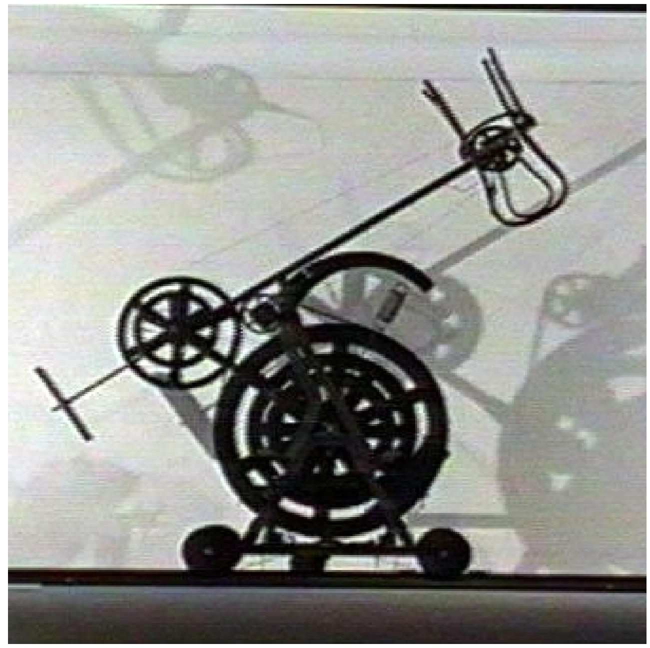


Fig.1 Machine with Chair

Mike Bonanno of the artist collective the YESMEN has a very different idea of what to do with technologies. In one particularly well-known event, Bonanno purchased dozens of Barbie and G.I. Joe dolls, switched the voice boxes of Barbie and G.I. Joe dolls and returned them to store shelves. All of a sudden the G.I. Joes would say “Let's go shopping today!”, and the Barbies would bark “Dead men tell no lies!”, or something along those lines. With this very simple technical and logistic intervention, Bonanno was able to critique gender stereotypes projected onto children's toys and make his message known to a wide audience in ways that other consumer protests never achieved.

Erwin Driessen & Maria Verstappen interpret 'situated interaction' in a literally hands-on way. The two artists developed a massaging robot; a robotic gizmo that crawls over your back and gives you a gentle tickle-like massage [Fig. 2]. The tickle robot also builds a 3D map of your back as it performs its massaging operations. The device delivers a pleasurable bodily experience that a data-centric approach to mapping the body could not capture. While the work is fully functional, it is preposterous and quixotic at once – machinic intimacy without empathy that delivers canned affection on demand.

Christa McPhee “borrowed”, from a sympathetic scientist, a mobile carbon flux measurement system [Fig. 3] used for environmental monitoring in global climate change studies on the tallgrass prairie in Kansas [McPhee, 2004]. Instead of analyzing the collected data as an environmental scientist might, she recorded the noises of the apparatus performing the measurements themselves, resulting in a kind of ambient 'measurement noise'. She also made the complete data set

Fig. 2 Tickle

Fig. 3 Slipstreamkonza

from the autonomous robotic system, unfiltered and raw, available to viewers such as not to "remove" anything, imagining that something unknown might be lingering in the vast lines of floating-point data. McPhee resisted numerical methods of summarizing or mining the data in any way. For her the data itself became a physical object of interest, not something that should be replaced by a convenient numerical summery.

## 3 Reflecting on Technical Practices

The above-sketched artworks are more than academic pleasantries and technology critique. Not that the HCI and AmI communities do not perform self-critique. Agre set the stage for this with "Computation and Human Experience" [Agre, 1997], with the central tenet of a "critical technical practice". Sengers and others [Sengers, 1999], [Boehner et al, 2005] have refined Agre's vision into a HCI methodology.

Other interesting input comes from Nijholt. He quotes Cohen who has spent time in a smart room in the "off" state (when the system was not receptive), and reports on the very odd feelings of being in a weird void [Nijholt, 2004]. New forms of alienation are formed when smart technologies go idle. Subsequently Nijholt suggests we build a relationship with ambient spaces and perceive them as social actors, without, however, specifying any details of what to do with this social actor.

Furthermore, Dourish and others have argued that the historical context that equated mind exclusively with computational processes (and from which the Weiser vision of ubiquitous computing emerged) is insufficient as a lasting conceptual framework of ambient intelligence and that information processing should be considered a cultural category [Dourish, 2005].

It is also well known that HCI researchers have put considerable effort into design strategies based explicitly on social and ethical norms. The iHome project at the University at Aarhus [Iversen, 2004] *rasion d'être* is to not make the home a carbon copy of the office as some large companies [De Ruyter and Aarts, 2004] have. Iversen proposes to revitalize utopian ideals in the design of technology and implement them with systems that foster collective experience through participatory design principles as opposed to individualistic experiences. However, the researchers do not accept the possibility of decoupling usefulness from technical invention as the media arts do. And this brings me back to the media arts.

*Media arts have built a qualitative but nonetheless operational craft of recontextualization in engaging sentient systems with people. Despite their technical limitations, media arts have contributed to critical diversity in our machine culture. And that is what should make them attractive for AmI researchers.*

## 4 Technically Engaged Media Arts

The list of artists working in close proximity to engineering paradigms has grown significantly in the last years. While it is not possible to describe this trend in detail here, the following paragraphs depict two select AmI related art works originating from the author's studio/lab that have found resonance in both the art as well as the engineering science communities.

### 4.1 A Nature Interpretation Center with Second Thoughts (UNSEEN)

"Unseen" is a nature interpretation center with second thoughts; an autonomous outdoor knowledge-mixing system that dynamically proposed expertise on plants and shared this with its visitors[1].

Nature interpretation centers are a romantic expression of the desire to understand and experience nature without giving up the comforts of civilization. Interpretation centers attempt to shore up this deficit through visual effects. Following trends in news and entertainment TV, they offer seductive media shows depicting portraits of wildlife busily eating, hunting, cleaning, and so forth—in contrast to the reality where usually nothing much happens.

A public garden offers an interesting conditioning of the natural environment for those interested in querying this cultural malaise. Midway between untouched, pristine land and controlled construction, public gardens are established forms of colonialized wildlife. Following the Linnaean tradition, marked trees and labeled plants promise clear classifications with no secrets. Paved paths and directional cues prevent accidental disorientation and exposure to unstructured spaces. There is no room and no need for questions.

Unseen proposes a very different approach to the experience of being in the outdoors. Set in the Reford Gardens of Grand-Métis on the Gaspé Peninsula of eastern Québec during the summer of 2003, our multi-camera real time machine vision system observed select plants indigenous to the region [Fig. 4] and shared its results with visitors.

Fig. 4 viewing box with one of four observation cameras

Working with unstructured spaces creates encounters with several levels of engineering and design challenges. Since our system was placed in a large garden in the middle of a forest one-half a mile from the nearest power outlet, we had to address a variety of adverse conditions: line voltage fluctuations, temperature variations, and high levels of humidity. Since our system was designed for public access, we had to anticipate peak traffic during weekends and allay anxieties about surveillance technologies.

The presence of high-bandwidth data collection machinery in public spaces of leisure can make people feel uncomfortable. While most people have grown accustomed to the presence of surveillance technologies in urban areas, they expect the outdoors to be unspoiled by surveillance technologies. The challenge consisted in finding ways to counter such anxieties and still be able to collect useful data over the course of the summer. The solution was simple but effective. We placed the cameras all below (human) eye level and cast the view downwards toward the ground. This posture of downcast stance signaled to visitors that the vision system was watching not them, but the garden.

Fig. 5 plant motion over time

The Dogwood, the Wild Sarsaparilla, the Harebell, the Foamflower, the Wild Columbine, the Garden Columbine, the Alpine Woodsia, the Lowbush Blueberry and the Canadian Burnet were under continued observation during the entire summer. Borrowing from data analysis and classification techniques, the system searched for, found and tallied instances of these plants. The system had a variety of display scenarios, amongst others one in which the change over time was visible. Such an image slowly intensified as the movements under the camera accumulated [Fig. 5]. This mode of seeing is unique to the machine. Using differential images, one can reduce the numeric representation of change to minimal information. The method was quite sensitive and could register the flight path of a large bee, for example.

Short texts, constructed from a large database of acknowledged expert sources, depicted factual data on the plants and on computer screens in a small hut adjacent to the garden. Over the course of the summer, however, the flavour of the texts changed. As the initially sparse garden grew luscious, the system followed the changes and altered its opinion, as it were, on the plants. The texts it created shifted from descriptive to hypothetical, and, having second

[1] Technical details on the vision and text creation system are described elsewhere [Böhlen2004].

thoughts, confronted the visitor with imagined future understandings of plant life, suggesting that a walk in the woods really is more than just an act of information processing. Unseen was an expert system driven by the very objects it observed. It was an open invitation to look again, with fresh eyes, human and synthetic, at a simple garden [Böhlen, Tan 2004].

### 4.2 Synthetic Hissy Fits – Amy and Klara

"Amy and Klara" are an experiment in probing the limits of the mimetic approach in robot design [Böhlen, 2006], in particular with regards to the use of synthetic speech. The project premiered at ISEA2006 in San Jose.

The logical consequence of the desire to mimic humans in synthetic systems is to mimic human idiosyncracies. The perfection of imperfection might become a useful heuristic, a negative feedback component in the ongoing pursuit of synthetic beings. A case in point is foul language. When humans learn a new language they usually acquire a odd mix of bare essentials and examples of foul language. This is interesting as foul language circumvents the unknown new language and connects the speaker directly back to known territories. While culturally specific in the boundary conditions that control its use, foul language links us more directly to our bodies than other forms of speech.

Fig. 6 Amy and Klara

In order to experiment with foul language in synthetic systems, the MediaRobotics Lab has built a set of robotic agents housed in cute pink boxes named Amy and Klara. Their ontologies are formed by and limited to reading and analyzing on-line trivia of life style magazines such as Salon dot com. There is no claim to universality or completeness in this borrowed simplistic epistemology. What is listed in this continuously updated collection of fashion, politics and celebrity trash is considered significant by the two robots. That which is mentioned repeatedly receives more computational weight than topics only listed once. Items that reach a critical threshold of numerically constructed significance become material for discussion. Amy and Klara share their statically weighted text summaries with each other via Text to Speech and Automated Speech Recognition. The results from the speech recognizer as well as the physical transmission of utterances from speaker to microphone are error prone; miscommunication is unavoidable. If the robots choose different topics and 'disagree' in their statistical evaluation, they begin to call each other names.

Both robots are equipped with video cameras and able to see each other. An adaptive histogram based hue detection algorithm allows them to detect the other box's pink even under varying lighting conditions. With a parametric disposition to be agitated by pink (and not knowing that they are themselves pink), they are primed for trouble when set next to each other. The fact that Klara has a thick German accent only increases the potential for misunderstanding. This game invariably ends in a rather ugly exchange of expletives that often leaves people watching the two robots wondering about machine intelligence[2].

Amy and Klara and their provocative synthetic hissy fits warn us not to expect too much from intelligent machines. But they do not fall into the pessimists' dystopian dead-end. They counter the rhetoric of the gentle intelligent machine with a critique of normative uses of synthetic speech and linguistic imperfection. Some might find it inappropriate to have robots curse at each other. But if foul language is out of bounds for machines, then what about other taboos? Will we map all our taboos onto robots once they look, sound or smell like we do? Many linguistic taboos are derived from taboos in religion, sex and mental and physical ailments directly related to the constraints of being human. Since machines lack our bodily functions, the corresponding taboos really need not hold. Also, we should expect some of our own taboos to be invalidated by machines. We should not be surprised if they invent new curse words particular to the experience of being machine and having speech. Languages human in origin will be altered and amended by their use in machines in similar ways as popular culture alters and adds over time, to the chagrin of purists, to the standardized English dictionary. There will be new figures of speech. Languages no longer in use by humans might be kept artificially alive in machines.

## 5 Conclusion

Because of the breadth of its technical scope, its historic predecessors and its emphasis on rich social interactions, AmI has a unique position amongst the engineering sciences to conceive and deploy robust intelligent systems that make life not just easier, but richer and foster new forms of participation. Because of its proximity to AmI research and its

[2] Details on this work, including the design strategy of generating the unscripted exchange of foul language is forthcoming in: "Robots with Bad Accents".

diversity in machine culture, media arts are now positioned to contribute to bona fide research in AmI and related fields.

## Acknowledgments

This work is supported in part by the Department of Media Study and the Humanities Institute of the University of Buffalo, SVOX AG Switzerland and FONIX Speech. Special thanks to Natalie Tan and Stefani Bardin for their suggestions.

## References

[Agre, 1997] Agre, P., Computation and Human Experience. Cambridge: Cambridge UP, 1997.

[Böhlen, 2004] Böhlen, M., A Patient Autonomous Knowledge Sharing System for Public Outdoor Spaces, American Association for Artificial Intelligence (AAAI2004), Spring Symposium on Interaction between Humans and Autonomous Systems over Extended Operation, Stanford, California.

[Böhlen and Tan, 2004] Böhlen, M., Tan, N., Garden variety pervasive computing, Pervasive Computing, IEEE, Volume 3, Issue 1, Jan.-March 2004 Page(s): 29 - 34.

[Böhlen, 2006] Böhlen, M., "Towards Machinic Maledicta", CHI2006, Workshop on Abuse of Interactive Technologies, Conference on Human Factors in Computing Systems, ACM SIGHCHI, (workshop paper), Montreal 2004

[Boehner et al, 2005] Boehner, K., Kaye, J., Sengers P., Critical Technical Practice as a Methodology for Values in Design. CHI 2005 Workshop: Quality, Value(s) and Choice: Exploring Wider Implications of HCI Practice. April 2004.

[Bonnano, 1989] Bonnano, M., The Barbie Liberation Organization, Text by Brigitte Greenberg, Associated Press. www.etext.org/Zines/UnitCircle/nonfiction/blo.html

[De Ruyter and Aarts, 2004] de Ruyter, B. and Aarts, E. 2004. Ambient intelligence: visualizing the future. In Proceedings of the Working Conference on Advanced Visual interfaces. AVI '04. ACM Press, New York, NY, 203-208.

[Dourish et al, 2005] Dourish, P., Brewer, J., and Bell, G. 2005. Information as a cultural category. interactions 12, 4 (Jul. 2005), 31-33.

[Driessen and Verstappen, 1999] Driessen, E., & Verstappen, M., The Tickle Robot and Tickle Salon, http://www.xs4all.nl/~notnot/tickle/TICKLEcat.html

[Gansen, 1995] Gansen, A., Machine with Chair http://www.arthurganson.com/pages/Sculptures.html

[Heidegger, 1957] Heidegger, M. Die Frage nach der Technik in: Martin Heidegger Gesamtausgabe, Bd7, Vortraege und Aufsaetze, Frankfurt a. M., Klostermann, 2000.

[Iversen et al, 2004] Iversen, O. S., Kanstrup, A. M., and Petersen, M. G. 2004. A visit to the 'new Utopia': revitalizing democracy, emancipation and quality in co-operative design. In Proceedings of the Third Nordic Conference on Human-Computer interaction. NordiCHI '04, vol. 82. ACM Press, New York, NY, 171-179.

[McPhee, 2004] McPhee, C., Slipstreamkonza Semiotics: Towards a Telemimetic Sublime in the Data Landscape in: Proceedings of the 4th International Converence on Computational Semiotics in Games and New Media, University of Split, Croatia, 2004.

[Nijholt et al, 2004] Nijholt, A., Rist, T., and Tuijnenbreijer, K. 2004. Lost in ambient intelligence?. In CHI '04 Extended Abstracts on Human Factors in Computing Systems. CHI '04. ACM Press, New York, NY, 1725-1726.

[Philips, 1999] Ambient Intelligence - Changing lives for the better, www.research.philips.com/technologies/syst_softw/ami/background.html

[Pias, 2004] Claus Pias, C., ed.,Cybernetics - Kybernetik. The Macy-Conferences 1946-1953, Band 1: Transactions/ Protokolle, Band 2: Documents / Dokumente, Diaphanes Verlag, Zürich 2004

[Pickering, 2002] Pickering, A., Cybernetics and the mangle: Ashby, Beer and Pask, Social studies of science, 2002, vol. 32, no3, pp. 413-437.

[Sengers 1999] Sengers, P., Practices for Machine Culture: A Case Study of Integrating Artificial Intelligence and Cultural Theory. Surfaces. Volume VIII, 1999.

[Shadbolt, 2003] Shadbolt N., Ambient Intelligence, IEEE Intelligent Systems archive, Volume 18, Issue 4 (July/August 2003), ISSN:1541-1672

[Weiser, 1991] Weiser, M. (1991), The computer for the twenty-first century, Scientific American 265(3), 94 -104.

# Machine Learning, Reasoning, and Intelligence in Daily Life: Directions and Challenges

**Eric Horvitz**
Microsoft Research
Redmond, Washington USA 98052
horvitz@microsoft.com

## Abstract

In my invited presentation, I will discuss how technical developments and trends are providing a fertile substrate for creating and integrating machine learning and reasoning into multiple applications and services. I will review several illustrative research efforts on our team, and focus on challenges, opportunities, and directions with the streaming of machine intelligence into daily life.

## 1 Reflections on Trends and Directions

Over the last decade, technical and infrastructural developments have come together to create a nurturing environment for developing and fielding applications of machine learning and reasoning—and for harnessing automated intelligence to provide value to people in the course of their daily lives. These developments include (1) technical advancements in machine learning and reasoning, (2) the growth in CPU and memory capabilities within commonly available devices and platforms, (3) the connectivity, content, and services provided by the evolving Web, and (4) the increasing availability of data resources, including corpora of behavioral data collected via inexpensive sensors, and through ongoing interaction with software and services.

### 1.1 Panoply of Applications and Services

Opportunities for integrating applications of machine intelligence into the daily lives of people are growing with the increasing popularity of computing systems, the widening diversity of web services, the growing popularity of portable devices that contain general-purpose operating systems, and ongoing innovations in human-computer interaction—including the increasing prowess of speech recognition, handwriting, and sketch-understanding interfaces.

Various examples of the integration of automated learning and reasoning into daily life have been appearing as implicit and explicit extensions to traditional systems and services, and in prototypes and systems that provide qualitatively new kinds of experiences. I will review several projects and efforts undertaken by our team that highlight directions and approaches to introducing potentially valuable machine learning and reasoning into the daily lives of people.

An example of an implicit integration of ambient learning and reasoning is work by our team to create a probabilistic action prediction and prefetching subsystem that is now embedded deep in the kernel of the latest version of Microsoft's Windows operating system. The predictive component, operating within a component in the Vista operating system called Superfetch, learns, by watching sequences of application launches over time, to predict a computer user's application launches. These predictions, coupled with a utility model that captures preferences about the cost of waiting, are used in an ongoing optimization to prefetch unlaunched applications into memory ahead of their manual launching. The implicit service seeks to minimize the average wait for applications to be ready to use after launching.

Moving from the depths of operating system kernels to the infrastructure of a city, there is great opportunity for collecting data and learning predictive models from constellations of sensors embedded throughout a large-scale region. The JamBayes traffic forecasting service [Horvitz *et al.*, 2005] serves as an example of an explicit extension of ambient intelligence to familiar digital views of maps that display the flow of traffic in urban areas. The JamBayes client, operating on desktop systems and portable devices, accesses a service that employs machine learning and reasoning about the context-sensitive flow of traffic. The system overlays predictions about *future* traffic conditions on a digital traffic flow map. The system combines multiple variables that consider key contextual signals as well as the dynamics of flow across a greater urban area to predict when free-flowing traffic will likely become jammed and how long it will be until current jams melt away. A companion system developed by our team named ClearFlow provides users with context-sensitive routing based on reasoning about the current and future flows on road segments within a regional traffic system.

JamBayes also provides a qualitatively new functionality—surprise detection and forecasting. We have been pursuing research on surprise forecasting, aimed at developing and fielding sensing and reasoning systems that have the ability to detect and to alert users when *current or future events will likely surprise them*. Such reasoning promises to provide significant value to people in the course of daily life as it explicitly considers the differences in computational forecasts and the expectations of people with regards to important outcomes. JamBayes' surprise modeling component considers a user's context-sensitive expectations about current and future traffic, and infers when situations that it is inferring would likely be interpreted as anomalous from the user's perspective. Users can configure the system to generate alerts when particular kinds of surprises occur, based on context. In operation, the surprise analysis lets people know when current or projected flows on routes that they have expressed interest in would likely surprise them as jammed or as free, based on a model of context-sensitive user expectations.

Let us now move to systems that learn and reason about a computer users activities and that can provide qualitatively new experiences for users. In the Web Montage prototype [Anderson and Horvitz, 2002], data collected in the background about a user's desktop activity is analyzed via machine learning, and predictive models are constructed that are used to generate personalized portals, pages that are composed by making automated clippings from multiple web sources. The system builds models of the cost of navigation and the value of content to triage information based on time of day and ongoing activities, and this content is arranged and laid out via an optimization of expected value.

Another example of overlaying intelligence to provide new kinds of experiences is the MemoryLens effort [Horvitz *et al.*, 2004], which has been exploring the use of machine learning and reasoning to model aspects of human memory. The LifeBrowser prototype developed within the MemoryLens effort employs inferences about the memorability of events and items to provide new kinds of browsing and searching experiences such as variable density timelines. It also uses the inferences to extend familiar search functionality by integrating a backbone of memorable personal events and activities into search results to assist with browsing and searching of information.

Other examples of new functionalities and services leverage components that reason about peoples' focus of attention and interruptability [Fogarty *et al.* 2005; Horvitz *et al.*, 1999; Horvitz *et al.* 2003b; Iqbal and Bailey, 2006], as well as about people's current and future presence and availabilities [Horvitz *et al.*, 2002]. We have explored the use of these components in systems that can reason about both the urgency of incoming communications and the status of a user's workload and focus of attention. Such systems have the ability to weigh the cost of deferring communications with the cost of interrupting the user. Examples of prototypes constructed in this realm include the Priorities system [Horvitz *et al.*, 1999], the larger, cross-platform alerting system named Notification Platform [Horvitz *et al.* 2003], and several of our Bestcom efforts.

The Notification Platform continues to build and triage information from multiple sources within a unified inbox, and reasons about the best devices and modalities to employ. The related Bestcom effort (for *best-means communications*) centers on assisting people to establish communications with one another. Research in this realm highlights the potential for machine learning and reasoning to play a significant role with enhancing communications in our daily lives.

Within Bestcom research, we have investigated the promise of systems that have the ability to understand communication preferences, the intent or goals of a communication, current and future availabilities, and context. Prototypes explore how agents, working on behalf of a contactor and contactee, can negotiate about the best time and type of communication to undertake, and then execute connections. In the general Bestcom methodology, a contactor requests a communication with a contactee (the contactor "Bestcoms" the contactee), and an optimization is executed based on identity, context, goals, connectivity, and devices available now and in the future. Actions may range from a real-time voice call to a rescheduling of a voice or videoconference for a later time. Various prototypes have been created, including systems that perform smart routing of calls based on context, executing on a portable device (*e.g.*, the Bayesphone [Horvitz, *et al.*, 2005], and systems employing a larger infrastructure, such as the Bestcom-ET system fielded internally at Microsoft [Horvitz *et al*., 2003c].

Machine learning and reasoning can provide value with the overall coordination of people in the course of daily life. The Coordinate system [Horvitz *et al.*, 2002] was initially developed to support the Notification Platform and Bestcom efforts, but the work led to stand-alone presence and availability forecasting services that provide people with new kinds of awareness. Coordinate continues to collect data about the presence and activities of people at different locations and devices, and employs machine learning and reasoning to perform availability forecasting. The system can pass its inferences to computational agents, or provide to people (who have been granted access privileges), such predictions as the time until a user will return to their office, read email, regain access to a networked computing system, or finish a conversation that is currently in progress.

### 1.2 Rise of Intention and Preference Machines

Moving from particular examples to trends, we are seeing the use of machine learning, inference, and decision making to drive the creation of *preference machines* and *intention machines* in multiple domains. Preference machines include the set of systems referred to as recommender systems, employing collaborative filtering to predict the preferences of users about different sets of items, content, and experiences based on partial information about activities, demographics, and preferences. Such systems typically leverage predictive models constructed from the activities or assessments of a large set of users. The world has come to know of such systems as recommendations in online commerce situations.

Intention machines are services that employ models that predict peoples' activities. Such work includes predictive models used in the Microsoft operating system, and numerous other projects. Intention machines have come to touch the lives of many millions of people via web search. The logs of queries and page accesses serve as a rich case library for building predictive models. Recent work on constructing predictive models from such log data is highlighted in [Downey *et al.,* 2007].

Microsoft has been employing machine learning based on result click-through data to continue to enhance the functions used to rank results associated with queries. Other models are used to identify overall satisfaction with results. Learning and reasoning is also being used to optimize the presentation of advertisements. Such research is undoubtedly going on at other companies providing Web search, and services associated with targeted advertising.

Such machines will see even more exotic uses, including the use of intention and preference machines in geocentric services. As an example, our team has been building intention and preference machines with data that we collected over several years from volunteers who participated in the Microsoft Multi-User Location Survey (MSMLS). We have been exploring the uses of the data in learning and reasoning systems, including the construction of a system that can predict and then harness drivers' likely destinations given partial driving trajectories [Krumm and Horvitz, 2006]. Beyond geocentric intention machines, we have been exploring the feasibility of building geocentric preference machines, that perform geocentric collaborative filtering: given sets of sensed destinations of multiple people and the sensed destinations of a particular driver, what places, never visited before by a driver, might that interested to that driver, and might that driver be informed (*e.g.,* by hearing a paid advertisement) when he or she is approaching such destinations.

Intention machines and preference machines are becoming important and increasingly common assets in business. Competitive pressures will lead to an ongoing polishing of these models.

## 2 Challenges and Opportunities

For all of the promise and opportunity, integrating machine intelligence into daily life faces an array of interesting challenges and opportunities. I will pause to highlight several challenges.

### 2.1 Learning and Supervision

Designs for introducing intelligent reasoning into the world often depend critically on acquiring a case library of rich data that can be used to build predictive systems. The construction of case libraries may require the labeling of hidden states, such as the ground truth of intentions or preferences. In some situations, users may have to engage in explicit tagging activities or *supervision*. In other situations, the target states of interest are tagged by definition at logging time. As an example, the core predictive models of Coordinate employ in-stream supervision, where states and transitions that capture user presence and availability are logged automatically. In many cases, it is possible to tag intentions and preferences with *in-stream supervision,* a term I use to refer to the tagging situations in an implicit manner, in the course of activity.

As another example of in-stream supervision, consider the methods employed in the Lookout calendaring and scheduling agent [Horvitz, 1999]. Lookout performs in-stream supervision to build two models without cost to the user. Lookout computes the likelihood that someone reviewing email might like to perform a scheduling task based on the content of the email message being focused on. The system collects and labels cases for building a model of a user's intentions in the course of a user's normal activity. To collect cases, Lookout runs in the background and notes when users examine an email message and then turn, within a time horizon to a calendar view or scheduling task. Positive and negative examples of messages, and the details of message headers and bodies, are stored along with the observed action in a case library.

Lookout also uses in-stream supervision to learn about the ideal timing of actions. The system watches behind the scenes and records the amount of time that people focus on email messages before moving onto calendaring tasks or onto other messages. Lookout builds a case library of messages and dwell times collected through such in-stream supervision and constructs a predictive model that provides real-time recommendations about how long the system should wait before engaging the user, given properties of the message (such as the message length) at the user's focus of attention. Such a predictive model, learned without human intervention, provides Lookout with an awareness of attentional patterns of users. The subsystem gives Lookout an ability to courteously withhold potentially distracting engagements while a user reviews a message.

A similar approach to supervision is used in the Priorities system for prioritizing email by urgency. Priorities learns to infer the expected cost of delayed review for each incoming email message. Classifiers can be constructed based on explicit supervised learning as well as on in-stream supervision, considering a user's activities. For example, messages that are deleted without being read are assigned a lower urgency than messages read soon after they arrive.

In-stream supervision methods do not have to be fully automatic and operate as complete sleuths. Priorities research also explored a middle ground of allowing users to become more involved with in-stream supervision. In versions of Priorities users could inspect and modify in-stream supervision policies. Such awareness and potential modification allows the in-stream supervision to become a grounded collaboration between the machine and user. The system also allows users to inspect the case libraries before invoking the modification or regeneration of predictive models. By collaborating in such a mixed-initiative manner *about* the process and accuracy of the tagging process, in-stream supervision can be made more accurate.

A number of systems have employed more costly probes for hidden states. For example, the BusyBody system learns and then uses personalized models that predict the cost of interruption of a computer user based on the user's activity and context [Horvitz *et al*., 2004b]. During a training phase, BusyBody makes intermittent requests for the current cost of interruption. We have investigated the prospect of minimizing user tagging effort by automatically tagging a user's cost of interruption via a proxy such as the delays in responding to notifications. Developing a grounded collaboration on tagging by coming up with shared tagging rules is appears to be promising for this application.

Beyond in-stream supervision, unsupervised and semi-supervised learning show promise for reducing the cost of training systems. In another approach, we have been exploring the harnesssing of active learning in guiding the allocation of supervision efforts, including an approach we refer to as *selective supervision*—the application of decision-theoretic methods to triaging cases for explicit tagging [Kapoor *et al.*, 2007]. This research includes efforts to reason in an ongoing manner about the acute costs of probing users versus the long-term benefits of enhanced performance of a system in an environment over time, with models systems that perform lifelong learning models.

On a related challenge, users may wish to use a system right away, before training the system. In one approach, pre-trained *generic* models can be made available and users can select and use the most appropriate model immediately. Such generic models would typically not be able to provide the accuracy of a personalized model. However, they can provide some value immediately. The case libraries of such models can be extended or washed out over time with new user-specific data. In other cases, a model mixture approach can be used, where the output of the initial model is weighted together with the output of a personal model that is growing with addition of new labeled cases over time.

The Microsoft Outlook Mobile Manager [Microsoft Corporation, 2000], a product derived from the Priorities effort, provides users with pre-built predictive model for email urgency. In the training process, user-specific features were removed, so the system operates, for example, on features expected to have overall universality, such as structural aspects of email, including length of messages and the number of people on the recipients list. As a user provides cases to the system, a personal model is constructed and is continually revised with the addition of new cases. As the quantity of cases grows in the personal the model, the model output is weighted more heavily with the output of the pre-built model, until the initial predictive model is completely washed out.

Challenging areas of research include developing a better understanding of the best approaches to constructing generic models that can provide valuable, usable initial experiences with intelligent applications and services, but that allow for efficient adaptation downstream with a user's explicit training efforts or in-stream supervision. Research may lead to deeper insights about setting up systems for "ideal adaptability" given expectations about the nature of different kinds of environments, and adaptations, given the targeted users and uses.

### 2.2 Criticality of Mixed-Initiative Capabilities

Intelligent systems with the ability to support a mix of machine and human initiatives to address problems at hand are especially critical for applications of ambient intelligence—where solutions, support, recommendations, and warnings might be offered in stream with ongoing activities [Horvitz 2007]. There is a great opportunity for developing systems that understand how to work in a collaborative manner with users, where the system has skills in recognizing opportunities for problem solving and in understanding which aspects of problems the machine versus the person might best solve. Mixed-initiative interaction would also benefit by providing systems with ability to infer subtleties of cognitive state of people so as guide the "if and when" interventions. A set of principles of mixed-interactive interaction and the value of harnessing decision-theoretic principles for guiding action under uncertainty for guiding mixed-initiative interaction are presented in [Horvitz, 1999]. Work is progressing on mixed-initiative user interaction on multiple fronts, including explorations into efficient interfaces and interactions for correcting recognition errors [Shilman et al., 2006].

### 2.3 Mental Models, Transparency, and Control

In bringing systems and services based on machine learning and reasoning into the world, we face a challenge of making the behavior of machine learning and reasoning systems understandable to people. Understanding and addressing concerns with the transparency, trust, and controllability of intelligent systems by lay people is a multidimensional and challenging research area. Concerns about how a system functions and how its functioning can be modified will likely increase as systems move into roles where they interact with users on daily activities that people care deeply about, and as the responsibilities of systems shift from gentle recommendations to higher-stakes actions

There has been little work to date on laypersons' perceptions and overall mental models about the operation of learning and reasoning systems, and about the influence that different mental models about automated reasoning may have on usage, acceptance, and effective configuration and control of systems by users. There is an opportunity to better understand the mental models and to construct explanations to make the workings and conclusions of a system more transparent and understandable.

People may wish to understand the basis for automated actions, both to be aware of the operation of systems, and to build confidence in the actions and policies executed in different settings. Such understanding can allow people to interpret why, in retrospect, different actions were taken. Knowledge about the operation and control of an intelligent system would allow users to understand if, when, and how behavior might be changed to better suit a user's wishes or expectations.

It may be useful for systems to employ explicit explanation subsystems [Suermondt, 1992] or even to modify reasoning with alternate approximations so as to enhance the transparency or understandability of the deliberation of reasoning systems [Horvitz *et al.,* 1989].

### 2.4 Privacy, Data, and Machine Intelligence

With the advent of increasingly ubiquitous sensing and reasoning, and the rise of preference and intention machines in multiple arenas, people will seek to understand and to control how personal data is being used. Overall, developing approaches to addressing potential concerns about the privacy of data harnessed for learning and reasoning will be useful in the creation and adoption of applications of machine intelligence. We are studying research on enhancing privacy in several ways.

**Protected sensing and personalization.** In many applications, it is feasible to perform machine learning and reasoning *within a protected shroud of privacy*, where the sensed data about activities and content of people is kept in the local control of their owners, *e.g.,* on stores within users' personal machines. For example, a geocentric intention machine with the ability to predict someone's destination while they are driving can be learned from personal GPS data that has been collected, stored, and processed locally. As another example, local analysis of potentially sensitive data within a system designed to protect sensing and personalization can be used to enhance web search; in work on the PS prototype, a comprehensive index of a users email, documents, and web search activity is constructed locally. A large list of web results is requested from a Web search engine, and this list of, *e.g.,* several hundred results, is reranked with local models that consider relationships between the content of the web pages and information in a user's personal store [Teevan *et al.*, 2005].

Protected sensing and personalization could be employed in conjunction with predictive models constructed from volunteers and with third-party content. For example, a predictive model about destinations can constructed from data built from multiple volunteers, and then incorporated and used locally within a user's shroud of privacy. A large cache of advertising content might be intermittently downloaded within a user's system, but matched privately within a user's local privacy shroud to do local targeted advertising based on location, web pages being viewed, and other content and activities.

**Learning and harnessing preferences about privacy.** The legal and organizational language used to define policies on the usages and access of personal data are often limited to particular conceptions of privacy for purposes of clarity, expediency, and universal application. However, "privacy" is not a simple, nor universal concept. A recent study of the sensitivities that people might have with sharing different types of information with people in different groups revealed that there are significant variations in preferences among people [Olson *et al.,* 2005]. These results highlight the potential value of performing additional investigation of preferences about allowable uses of personal data in reasoning systems—and the application of learned insights to the design of expressive representations, interfaces, and controls that can allow people to custom tailor policies for the sharing of data with people and applications.

As a community, we need to better understand the varying preferences of people with regard to the use of their personal data in learning and reasoning systems. Such understanding includes the study of potentially time-varying sensitivities about the use of data and the trades that people may be willing to make in cases where valuable services are enabled when data is shared with organizations and people. Machine learning itself can be useful in probing preferences about privacy.

**Partial revelation.** Personal data can be transformed via such processes as anonymization, summarization, abstraction, and obscuration (via such techniques as the controlled introduction of errors) in pursuit of making the sharing of the date more acceptable with people, organizations—or a specific reasoning system hosted by a particular organization. There is an opportunity to study peoples' preferences about sharing data that has undergone different transformations with different kinds reasoning systems. Such preferences can be coupled with analyses that provide insights about losses in the fidelity of reasoning with different transformations so as to provide guidance about the most appropriate matching of transformations to end uses.

**Restricted rights.** There is opportunity to develop methods that annotate user data with privacy metadata that restricts the usage of the data to specified uses. Developing schemas and infrastructure to support data rights management for restricting the use of personal data would likely be a significant undertaking, including the development and widespread adoption of standards. Such a privacy infrastructure, perhaps developed as part of more comprehensive efforts to introduce standards for representing higher-level semantics on the Web, could be enabling for developing intelligent applications that employ sensitive data in a trusted manner.

## 3 Summary

I discussed trends that are leading to a nurturing environment for creating valuable applications of machine intelligence. I presented several examples of implicit and explicit applications of automated intelligence, including applications that are enabling new services and experiences. Then, I reviewed several challenges, touching on topics in the realms of learning, mixed-initiative interaction, undertandability and control, and the privacy of data resources used in machine learning and reasoning. I hope that my reflections are helpful to the community of researchers pursuing the tantalizing vision of enhancing the lives of people via the fielding of applications of machine intelligence.

## Acknowledgments

I am indebted to the researchers and engineers on the Adaptive Systems & Interaction team at Microsoft Research, as well as to my colleagues and students at other centers who contributed insights to the projects and ideas that I have highlighted here.

## References

[Anderson and Horvitz, 2002] Corin R. Anderson and Eric Horvitz. Web Montage: A Dynamic Personalized Start Page. *Proceedings of WWW, World Wide Web Conference*, pages 704-712, Honolulu, HI, May 2002.

[Downey *et al.*, 2007] Doug Downey, Susan T. Dumais, and Eric Horvitz. Models of Searching and Browsing: Languages, Studies, and Applications. *Proceedings of IJCAI*, Hyderabad, India, January 2007.

[Fogarty *et al.*, 2005] James Fogarty, Andrew J. Ko, Htet Htet Aung, Elspeth Golden, Karen P. Tang, and Scott E. Hudson. Examining Task Engagement in Sensor-Based Statistical Models of Human Interruptibility. *Proceedings of CHI*, pp. 331-340, Portland, OR, April 2005.

[Horvitz et al. 1989] Eric Horvitz, David Heckerman, Keung Chi Ng, Bharat Nathwani, Heuristic Abstraction in the Decision-Theoretic Pathfinder System, *Proceedings of Symposium on Computer Applications in Medical Care*, Washington DC, IEEE Press: Silver Springs, MD, November 1989.

[Horvitz, 1999] E. Horvitz. Principles of Mixed-Initiative User Interfaces. *Proceedings of CHI, Conference on Human Factors in Computing Systems*, pages 159-166, Pittsburgh, PA, May 1999.

[Horvitz *et al.*, 1999] Eric Horvitz, Andy Jacobs, David Hovel. Attention-Sensitive Alerting. *Proceedings of UAI, Conference on Uncertainty and Artificial Intelligence*, pages 305-313, Stockholm, Sweden, July 1999.

Horvitz, *et al.*, 2002] Eric Horvitz, Paul Koch, Carl.M. Kadie, and Andy Jacobs. Coordinate: Probabilistic Forecasting of Presence and Availability. *Proceedings of UAI, Conference on Uncertainty and Artificial Intelligenc*e, pages 224-233, Edmonton, Canada, July 2002.

[Horvitz *et al.* 2003] Eric Horvitz, Carl M. Kadie, Timothy Paek, and David Hovel. Models of Attention in Computing and Communications: From Principles to Applications, *Communications of the ACM*, 46(3):52-59, March 2003.

[Horvitz *et al.*, 2003b] Eric Horvitz and Johnson Apacible. Learning and Reasoning about Interruption. *Proceedings of the International Conference on Multimodal Interfaces*, pages 20-27, October 2003, Vancouver, Canada.

[Horvitz *et al.*, 2003c] Eric Horvitz, Johnson Apacible, Muru Subramani., Raman Sarin., Paul Koch., J.J. Cadiz, Atilla Narin, and Yong Rui. Experiences with the Design, Fielding, and Evaluation of a Real-Time Communications Agent, *Microsoft Research Technical Report* MSR-TR-2003-98, December 2003.

[Horvitz *et al.*, 2004] Eric Horvitz, Susan T. Dumais, and Paul Koch. Learning Predictive Models of Memory Landmarks, *Proceedings of CogSci, Annual Meeting of the Cognitive Science Society*, Chicago, IL, August 2004.

[Horvitz *et al.*, 2004b] Eric Horvitz, Paul Koch, and Johnson Apacible. BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption. *Proceedings of CSCW, Conference on Computer-Supported Collaborative Work*, pages 507-510, Chicago, IL, November 2004.

[Horvitz *et al*., 2005] Eric Horvitz, Johnson Apacible, Ramin Sarin, and Lin Liao. Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service, *Proceedings of UAI*, July 2005.

[Horvitz *et al.*, 2005] Eric Horvitz, Paul Koch, Raman Sarin, Johnson Apacible, and Muru Subramani. Bayesphone: Precomputation of Context-Sensitive Policies for Inquiry and Action in Mobile Devices. *Proceedings of User Modeling*, pages 251-260, Edinburgh, Scotland, July 2005.

[Horvitz, 2006] E. Horvitz. Reflections on Challenges and Promises of Mixed-Initiative Interaction, *AAAI Magazine* 27:4, December 2006.

[Kapoor *et al.*, 2007] Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning, *Proceedings of IJCAI*, Hyderabad, India, January 2007.

[Krumm and Horvitz, 2006] John Krumm and Eric Horvitz. J. Krumm and E Horvitz. Predestination: Inferring Destinations from Partial Trajectories, *Proceedings of Ubicomp*, pages 243-260, Irvine, CA, September 2006.

[Microsoft Corporation, 2000] Outlook Mobile Manager. http://research.microsoft.com/adapt/omm/

[Olson *et al.*, 2005] Judith S. Olson, Jonathan Grudin, and Eric Horvitz: A Study of Preferences for Sharing and Privacy. *Proceedings of CHI*, pages 1985-1988, Portland, OR, April 2005.

[Shilman et al., 2006] Michael Shilman, Desney S. Tan, and Patrice Simard. CueTIP: A Mixed-Initiative Interface for Correcting Handwriting Errors. *Proceedings of Symposium on User Interface Software and Technology*, pages 323-332, Montreux, Switzerland, November 2006.

[Suermondt, 1992] Henri J. Suermondt. Explanation in Bayesian Belief Networks. *Ph.D. Dissertation*, Department of Computer Science and Medicine, Stanford University, Stanford, CA.

[Teevan *et al.* 2005] Jaime Teevan, Susan T. Dumais, Eric Horvitz. Personalizing search via automated analysis of interests and activities. *Proceedings of SIGIR, Conference on Information Retrieval*, pages 449-456, Salvador, Brazil, August 2005.

# AI Techniques for Ambient Intelligence

# Research Papers

# Towards a Context Theory for Context-aware systems

**Richard Dapoigny**
Université de Savoie,
LISTIC/Polytech'Savoie B.P. 806,
F-74016 ANNECY Cedex, France
richard.dapoigny@univ-savoie.fr

**Patrick Barlatier**
Université de Savoie,
LISTIC/Polytech'Savoie B.P. 806,
F-74016 ANNECY Cedex, France
patrick.barlatier@univ-savoie.fr

## Abstract

The basic goal of context-aware systems is to make software aware of the environment and to adapt to their changing context. Building context-aware software is a complex task due to a lack of appropriate formal model in dynamic environments. While significative formalizations have been proposed, context models are often expressed either through logical formalisms or with ontology-based approaches. In this paper, we propose a model which combines the strengths of both approaches while trying not to carry their specific weaknesses into the resulting formal framework. The formal model relies both on a knowledge representation with ontologies and on a logical reasoning with Dependent Record Types (DRT) based on intuitionistic type theory and the Curry-Howard isomorphism. This logic modelling can be applied to any kind of process-based applications.

## 1 Introduction

Much research has been done in the area of context modelling, representation and reasoning in the past few years. It concerns a number of research communities ranging from human-computer interaction to pervasive computing. Unfortunately, there is no consensus about context definition, representation or modelling. Therefore, an epistemologically adequate and sound model is required which includes mechanisms for context integration and usage. There is a number of context definitions in the literature and it is obvious that they are greatly influenced by the target application framework. From an ontological perspective, the concept of context belongs to the class of "moment universal" [Costa *et al.*, 2006], which means that the context existentially depends on other concepts to exist. Since contexts are subsumed by the concept of property, we should speak of *context-of* rather than simply *context*. In this paper, we focus on the context of physical processes. It is an important subclass of the real-world applications in which any physical process can be modelled by incorporating contextual information. This subclass includes processes such as engineering and control applications, Web engineering, clinical tasks [Peleg *et al.*, 2003] or business processes [Balabko and Wegmann, 2003].

There are a number of issues that need to be addressed in context aware computing, including how context information is modelled and how we can reason about context information. To cope with these issues, most prominent context models developed in pervasive computing and artificial intelligence, are either logic-based [McCarthy, 1993; Giunchiglia, 1992] or ontology-based [Strang *et al.*, 2003]. After considering recent comparative studies[1] of significant techniques in context modelling [Strang and Linnhoff-Popien, 2004; Mügge *et al.*, 2006], it appears that, first the ontology-based approaches are the most appropriate and second that logic-based and ontology-based approaches are rather complementary[2]. In the rest of the paper, we suggest a context theory which integrates both a context logic based on Intuitionistic Type Theory, and a domain ontology to enable the systematic development of context-sensitive applications.

## 2 Context in AI

### 2.1 The need for a Context Model

The issue of context has been emerged in various areas of AI, including knowledge representation, natural language processing, and intelligent information retrieval. In context-aware systems, a well-known definition is expressed as follows "a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task" [Dey and Abowd, 2000]. But such a definition requires first to categorize explicitly the concept of context. The strong link between contexts and situations is highlighted in [Giunchiglia, 1992] where three different possibilities are detailed (a fourth one is the combination of the three basic ones). In the first one, a given situation is related to multiple contexts, each being a different approximate of the situation. We argue that these contexts are different because they correspond to different goals that we have in mind, speaking for a *context-of* definition. In his one-to-one correspondence between situation and context, the author gives an example involving a time evolution of a physical process. Considering first that a physical process has a global goal and second, that this goal can be divided into sub-goals, each of them being achieved successively, it results that, at a given time a single *context-of* is available (i.e., the *context-of*

[1] based on general requirements

[2] especially on distributed composition and partial validation

the related sub-goal). The third possibility, in which a single context corresponds to many situations, argues for a hierarchical property of contexts. If we can merge all contexts corresponding to a situation, provided that the situation is itself divided into sub-parts each of them related to a context, then we can conclude that context must offer hierarchical properties, which gives some evidence for an ontological definition.

### 2.2 Related Works

Ontology-based approaches are a promising framework to specify concepts and their interrelations [Uschold and Grüninger, 1996; Guarino, 1998]. They are particularly suitable to project parts of the information describing and being used in our daily life onto a data structure utilizable by computers. The development of context models based on ontologies take advantage of their knowledge sharing, logic inferencing and knowledge reuse capabilities. In [Wang *et al.*, 2004], the authors have created an upper ontology which captures general features of basic contextual entities and a collection of domain specific ontologies with their features in each subdomain. The CoBrA system [Chen *et al.*, 2003] provides a set of ontological concepts to characterize entities such as persons, places or several other kinds of objects within their contexts. It implements a broker-centric agent architecture to provide runtime support for context-aware systems. A recent survey of the most significant context modelling approaches [Strang and Linnhoff-Popien, 2004] argued that ontology-based approaches seem one of the most promising assets for context modelling.

In logic-based models, contextual information is introduced, updated and deleted in terms of facts or inferred from a set of rules. In the early ninety, the most significant general foundations for reasoning about contexts have explored logics rooted in the Situation Calculus. In [McCarthy, 1993], contexts are first class objects and the formula $ist(c, p)$ asserts that the formula $p$ is true in the context $c$. Lifting rules are able to relate the truth in one context to the truth in another one. In addition, operations such as entering and leaving contexts were introduced. This work was the starting point for a formalization in propositional logic [Buvac *et al.*, 1995] extended to first-order languages [Buvac, 1996]. In [Ghidini and Giunchiglia, 2001; Giunchiglia, 1992], the formalization relies on two assumptions, namely the locality principle where reasoning happens inside a context and the principle of compatibility where relationships can occur between reasoning processes in different contexts. The approach known as Multi-context Systems considers a context as a subset of the complete state of an individual. Contexts are seen as partial theories which can interact with each other through bridge rules allowing to infer a conclusion in one context from a premise in another one. More recently, an approach of contexts exploring type-theoretic ideas has been proposed in [Thomason, 1998][Thomason, 1999]. Three primitive syntactic constructions are derived from type theory, namely identity, functional application and lambda abstraction. Built on a previous work on intensional logic [Montague, 1970], propositions are defined as sets of possible worlds and a context is identified with a set of propositions. In Human Computer Interactions, context is seen as a feature of interaction (a property of information, or of objects) and the proposed a model includes context and activity which are mutually constituent [Dourish, 2004].

## 3 The Context Theory

### 3.1 The Context Logic

While the approaches described in section 2.2 cover a wide spectrum of AI modelling, they show some difficulty to manage the intensional aspect (some of them present modality as a solution). One important aspect already underlined in [Giunchiglia, 1992], is the locality principle. This principle which states that different portions of knowledge are defined, is taken as a basis in the present work. Then, observing that physical objects are a central element of process activity and building on recent works [Boldini, 2000; Ranta, 2004; Villadsen, 2004], we propose a context model based on Intuitionistic Type Theory (ITT) for engineering applications. We consider the *context-of* the application, that is, the *context-of* the global goal (since the application is characterized by an objective). The analysis of contextual reasoning requires sound and expressive formalisms. Widely used in Natural Language processing [Boldini, 2000; Ranta, 2004; Cooper, 2005] and in Programming Languages [Bove and Capretta, 2001; Paulson, 1989; Coquand and Coquand, 1999], ITT [Martin-Lof, 1982] has been proven to be appropriate to support the linguistic nature of physical situations. We extend these works with the description of contexts. However, contexts are not situations but they are related to them and since contexts describe knowledge extracted from situations, they have a natural expressivity with types. Moreover, ITT provides an intensional type theory for representing most of the features characterizing contexts in an AI perspective. In the framework of physical processes, we introduce the Physical Context (PC) described by a record in which the fields detail the physical items (i.e., objects and their properties) of that context. Record types and record tokens (i.e., instances of types) are also introduced to separate the specification of potential contexts (through types) with their implementation (through tokens). A further attraction is that this subdivision also supports ontological engineering for the specification of applications during the design step.

The theory of types has been extended with Dependent Record Types (DRT) [Betarte, 2000; Kopylov, 2003] which formalizes a proof of a basic property of groups. Their ability to provide a simple structure that can be reused to specify different kinds of structured semantic objects is very attractive.

**Definition 1** *A dependent record type is a sequence of fields in which labels $l_i$ correspond to certain types $T_i$, that is, each successive field can depend on the values of the preceding fields:*

$$< l_1 : T_1, l_2 : T_2(l_1) \ldots, l_n : T_n(l_1 \ldots l_{n-1}) > \quad (1)$$

*where the type $T_i$ may depend on the preceding labels $l_1, ..., l_{i-1}$.*

A similar definition holds for record tokens where a sequence of values is such that a value $v_i$ can depend on the values of the preceding fields $l_1, ..., l_{i-1}$:

$$< l_1 = v_1, ..., l_n = v_n > \quad (2)$$

The empty sequence <> is a record type and the type $T_i$ is a family of types over the record type $l_1 : T_1, ..., l_{i-1} : T_{i-1}$. Assuming that Γ is a valid context[3], the record type formation rules are, provided that $l$ is not already declared in $R$ :

$$\frac{}{\Gamma \vdash <>: record - type} \qquad \frac{R : record - type}{\Gamma \vdash R \sqsubseteq <>}$$

$$\frac{\Gamma \vdash R : record - type \;\; \Gamma \vdash T : record - type \rightarrow type}{\Gamma \vdash < R, l : T >: record - type} \tag{3}$$

We also assume that the general rules of construction for $Set$ and $Prop$ types are valid and that primitive syntactic constructions (i.e., equality, functional application and lambda abstraction) hold (for more details see [Martin-Lof, 1982]). An important aspect of DRT is that sub-typing is allowed, for example a DRT with additional fields not mentioned in the type is still of that type. We can use the notion of record-type to offer the most rudimentary notion of PC, namely that it is a record which carries information about the semantic of any physical variable. In such a way, PC-types can range from the context of a single variable involved in a physical equation to the context of a real life situation. While types describe potential properties of the real world, tokens are dedicated to the run-time process. From basic ground types, e.g., $Location$, $Phys\_quantity$ (physical quantity), ..., complex PC-types can be composed with more than one variable. Propositions are treated as individuals that can be arguments of predicates. Let us consider the pressure measurement via a pressure sensor (e.g., a Pitot tube) in a small volume referred as $Vol_1$ of a water channel. Ground types generate the respective PC-types and tokens:

| | |
|---|---|
| $x : Location$ | $x = Vol_1$ |
| $q_1 : Liquid(x)$ | $q_1 = p_1$ |
| $y : Phys\_quantity$ | $y = Pressure$ |
| $q_2 : has\_PhysicalQuantity(x, y)$ | $q_2 = p_2$ |
| ... | ... |
| * PC-type * | * PC-token * |

in which $p_1$ and $p_2$ are the respective proofs of $Liquid(Vol_1)$ and $has\_PhysicalQuantity(Vol_1, Pressure)$. Since the Curry-Howard isomorphism identifies proofs with programs, it can be used to prove a specification, that is to say, to select which definitions are needed for the specification to work properly.

| | |
|---|---|
| $x : Vehicle$ | $x = truck$ |
| $y : RegistrationNumber$ | $y = 2678KX69$ |
| $l_1 : GPSLongitude$ | $l_1 = 12.0987$ |
| $l_2 : GPSLatitude$ | $l_2 = 67.2365$ |
| $t_e : evTime$ | $t_e = 2/11/06.11 : 33$ |
| $t_m : maxTime$ | $t_m = 2/11/06.12 : 00$ |
| $q_1 : has\_identification(x, y)$ | $q_1 = p_1$ |
| $q_2 : has\_Location(x, l_1, l_2)$ | $q_2 = p_2$ |
| $t : evTime < maxTime$ | $t = 2/11/06.11 : 33 < 2/11/06.12 : 00$ |

[3]A valid context in type theory is a sequence $x_1 : T_1, \ldots x_n : T_n$ such that there is a judgment having it as left side of a sequent.

In this second example, the context token states that the registration number of all vehicles present at this place before $2/11/06.12 : 00$ will be registered. The extension of a physical context type needs to define some basic rules of construction. The first rule asserts that a concatenation of PC types is again a PC type. The second rule states that any PC-type is a sub-type of the empty sequence (since a PC type with no labels doesn't impose any constraints on its objects). The two remaining rules specify that any concatenation of PC types is a subtype of each one of its component PC type since the former contains more specific information.

$$\frac{\Gamma \vdash c_1 : PC - type \;\; \Gamma \vdash c_2 : PC - type}{\Gamma \vdash c_1 \oplus c_2 : PC - type}$$

$$\frac{\Gamma \vdash c : PC - type}{\Gamma \vdash c \sqsubseteq <>}$$

$$\frac{\Gamma \vdash c_1 \oplus c_2 : PC - type}{\Gamma \vdash c_1 \oplus c_2 \sqsubseteq c_1} \qquad \frac{\Gamma \vdash c_1 \oplus c_2 : PC - type}{\Gamma \vdash c_1 \oplus c_2 \sqsubseteq c_2}$$

The extension of a physical context type $c$ to a context type $c'$ corresponds to the process of getting more information. These rules are extensible to any number of context types. Since in type theory, the analogue of a proposition is the judgement we can conclude that the judgement in $c$ is lifted to the judgement in context $c'$.

### 3.2 Discussion

The logical framework must provide an adequate coverage through the following dimensions:

- partial logic. The logic must be partial to account for expressions which simply lack a value in some contexts. The partiality is obviously an inherent property of DRT since truth values are attached to the presence (or absence) of an element of a set.
- dynamic aspect. The dynamic aspect is revealed at run-time through the context tokens stating what is true within a context at this time.
- non-monotonicity. Roughly speaking, monotonicity indicates that learning a new information cannot reduce the set of known information. During context extension, if the assumptions that hold in the added PC contradict judgments of the initial PC, non-monotonicity occurs. As a consequence, a lifted judgment formally derives the absurd type (⊥). We face a belief revision process since beliefs have to be changed to accommodate the new belief otherwise inconsistency can occur. An efficient way to cope with non-monotonic situations as suggested in [Boldini, 2000], consists either in declaring the extended PC as impossible, in other words to discard the extension, or alternatively, to revise the belief leading to a contradiction in the initial context. Anyway, one PC has to be declared as impossible context.

### 3.3 The Context Ontology

Context modelling abstractions are required to support common understanding and to facilitate communications between agents in distributed environments. A recent approach [Costa *et al.*, 2006] has proposed to characterize the concept of context within an ontological framework. The authors introduce some concepts such as the "containment context" which associates multiple entities within a context, the quality requirement in which entities are bearers of qualities (intrinsic moment), the formal relations between individuals and material relations in which entities share a common property.

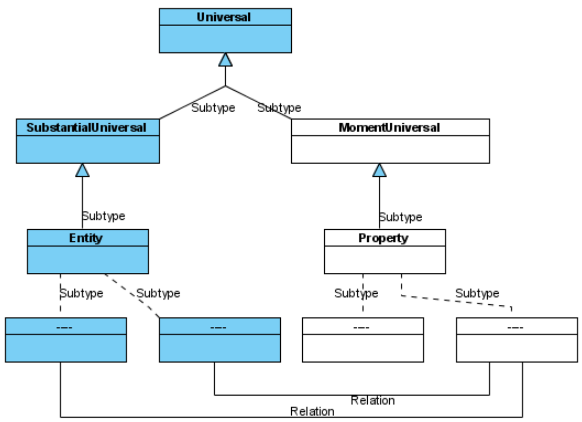



Figure 1: The Core Ontology

Despite its interesting definitions of entities and contexts, this conceptual model of context is not sufficient to cover all aspects of context reasoning since the underlying logic is not explicit. Moreover, the *context* concept described as a *moment universal* doesn't formalize the notion of *concept-of*. The main idea is to substitute the *context* concept with the more atomic concept of *property*. As a result we obtain the Core Ontology described in figure 1. The kind of possessive, allows to semantically relate the entity noun to a property noun in a specific "*has_<property>*" relation. Through the fundamental categories of *Entity* and *Property*, all the basic components of the physical world are defined. Above these assumptions, the containment context is nothing else than a context extension (sub-context-of) shifted in the logic, the intrinsic properties and the material relations by "*has_<property>*" while formal relations are directly expressed by propositions in the logic. Classical subsumption relations and material relations between entities and their properties explicitly conceptualize in a simple way a given application. The resulting upper level ontology does'nt formalize the context, but rather its elementary components, i.e., the properties. However, the ontological part is not sufficient by itself and must be extended with an appropriate logic. In such a way, a Context Theory can be formulated with a dual model (see Fig. 2). The first component (Conceptual part) built in the design step defines the ontology of the application and follows the construction rules of the upper level ontology. The second component (logical part) addresses the logical aspect relating the concepts defined in the previous layer through the generation of PC types including entities and propositions. It is only in this part (also built during the design step) that the context appears in a dynamic way.

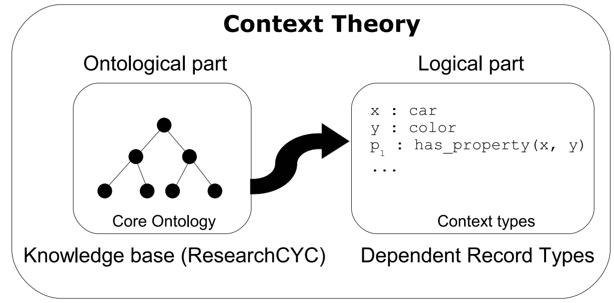



Figure 2: The Context Theory

From an extended perspective, the Context Theory presents capabilities either for more applied processes or for a more conceptual approach. A practical extension of the model appears through to the Information Flow (IF) paradigm of [Barwise and Seligman, 1997] which relates to the works cited by the end of 2.1. The conceptual part together with an executive module correspond to the classification of the IF model while the logical layer is nothing else than the Logic of the IF model. The IF model serves as a basis for distributed reasoning and has already been applied to ontology alignment. As a consequence, it extends easily the implementation of the present model in distributed environment under a categorical framework. Furthermore, a more higher level relates to the Standard Upper Ontology (SUO) with the lattice of Theories (LOT) [Kent, 2004]. It advocates that the present work also offers some facilities for an integration within this highly conceptual model.

### 3.4 The GLM Problem

The example, originally proposed by McCarthy, concerns an air travelling domain in which a traveller (named John) must plan a Glasgow-Moscow flight. He knows that there are two successive flights, one from Glasgow to London and another from London to Moscow. The objective is to formalize the reasoning process in which the traveller may be in Moskow after two flights unless he loses the ticket in London. The context model must cover the lost of a ticket in London. The first step is to define the goal to achieve, that is the proposition $g_1 = move\_to(John, Moskow)$. This goal can be divided in two subgoals, i.e. $move\_to(John, London)$ and $move\_to(John, Moskow)$. In this simple example, these two goal (tokens) stem from a single goal type, i.e., $move\_to(e_1, e_4)$ in which $e_1$ is of type $Person$ and $e_4$ of type $DestinationTown$. Therefore, a context type $c_1$ can be easily constructed from the domain ontology. Entities which range from $e_1$ to $e_4$ can be either intangible entities (e.g., $e_3$ and $e_4$) or spatial entities (e.g. $e_1$ and $e_2$). With $e_1 = John$, $e_2 = t0015$, $e_3 = BA515$ and $e_4 = Moskow$, an assignment like $p_3 = w$ corresponds to the typing judgement $w : has\_Destination(BA515, Moskow)$. As discussed elsewhere [Turner and Stevenson, 1991], achievement of goals within a process is context-dependent. As a consequence, the notion of *context-of* is formalized with a relation between the context and the goal type [4]. It is the present situation which makes true (or not) the relevant context of a given goal.

$$
Typ(c) \quad
\begin{array}{lcl}
e_1 & : & Person \\
e_2 & : & Ticket \\
p_1 & : & own(e_1, e_2) \\
e_3 & : & Flight \\
p_2 & : & has\_Flight(e_2, e_3) \\
e_4 & : & DestinationTown \\
p_3 & : & has\_Destination(e_3, e_4) \\
\ldots & : & \ldots
\end{array}
$$

The context propositions derived from the ontology can describe intrinsic properties as well as relational properties which relate two (or more) entities through their property. If the context is valid, after the two flights the traveller will be at Moskow but if he loses its ticket at London, the plan will fail since $p_1$ is false and then, the context $c_1$ is no longer valid. The difficulty with this problem is that the number of possible obstacles to the plan such as the loss of ticket, crew on strike, ... can be huge for high-level goals and it is not possible to take all of them into account. But if we consider elementary goals, it becomes possible to attach the possible obstacle(s) with them. The consequence is that a mechanism of goal composition with their related contexts must be investigated.

Figure 3 shows the conceptual model for an agent which implements the GLM example. The Common Lisp constraint packages SCREAMER and SCREAMER+ provide the support for combining constraints with typical Lisp data structures (the relation with the ResearchCyc ontology is not shown here).

## 4 Conclusion

We have explored the context theory as a framework for representing contextual information and inter-contextual information flow. Structuring facilities are conceptually grounded within the core ontology according to conceptual theories of philosophy and cognitive sciences. Checking the existence of entities with their properties (i.e. propositions) at run-time would make this approach suitable for context-aware applications. This paper has given some account of key features of DRT such as dependent types, propositions as types and types as objects. Instead of defining ontological relations between entities and contexts, the DRT introduces only two sorts (i.e., entities and prop) and expresses all other relations through context extension. The major interest in the DRT comes from their ability to express context dependencies in a dynamic way through context extension and their scalability through recursion. A prototype is currently under development with *researchCYC* to build context types (design step) and a *Lisp*_like engine allowing dynamic typing at runtime. We will also explore the distributed extension of this approach through IF-based formalisms. The dynamic integration relies on channel theory (IF-based) in which context theories can be logically integrated in a sound way.

[4]Notice that more than a single goal can be related within the context type.

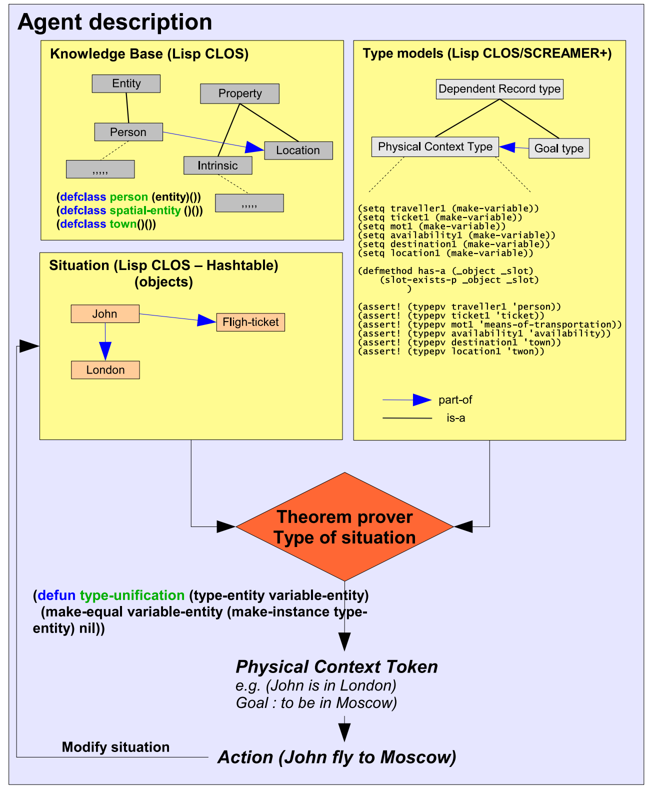



Figure 3: Context extraction from current situation in the GLM problem

## References


[Balabko and Wegmann, 2003] Pavel Balabko and Alain Wegmann. Context based reasoning in business process models. In *Procs. of the IEEE Int. Conf. on Information Reuse and Integration (IRI 2003)*, pages 120–128, 2003.

[Barwise and Seligman, 1997] Jon Barwise and Jerry Seligman. *Information Flow*, volume 44 of *Cambridge tracts in Theoretical Computer Science*. Cambridge University Press, 1997.

[Betarte, 2000] Gustavo Betarte. Type checking dependent (record) types and subtyping. *Journal of Functional and Logic Programming*, 10(2):137–166, 2000.

[Boldini, 2000] Pascal Boldini. Formalizing context in intuitionistic type theory. *Fundamenta Informaticae*, 42(2):1–23, 2000.

[Bove and Capretta, 2001] Ana Bove and Venanzio Capretta. Nested general recursion and partiality in type theory. In R.J. Boulton and P.B. Jackson, editors, *Procs. of the 14th Int. Conf. on Theorem Proving in Higher Order Logics*, number 2152 in LNCS, pages 121–135. Springer, 2001.

[Buvac *et al.*, 1995] Sasa Buvac, Vanja Buvac, and Ian A. Mason. Metamathematics of contexts. *Fundamentae Informaticae*, 23(3):412–419, 1995.

[Buvac, 1996] Sasa Buvac. Quantificational logic of context. In *Procs. of the 13th National Conference on Artificial Intelligence*, volume 1, pages 600–606, 1996.

[Chen *et al.*, 2003] Harry Chen, Tim Finin, and Anupam Joshi. Using owl in a pervasive computing broker. In *Procs. of Workshop on Ontologies in Open Agent Systems (AAMAS'03)*, 2003.

[Cooper, 2005] Robin Cooper. Records and record types in semantic theory. *J. Log. Comput.*, 15(2):99–112, 2005.

[Coquand and Coquand, 1999] Catarina Coquand and Thierry Coquand. Structured type theory. In *Workshop on Logical Frameworks and Meta-languages*, 1999.

[Costa *et al.*, 2006] Patricia Dockhorn Costa, Joao Paulo A. Almeida, Luis F. Pires, Giancarlo Guizzardi, and Marten van Sinderen. Towards conceptual foundations for context-aware applications. In *Procs. of the AAAI'06 Workshop on Modeling and Retrieval of Context*, pages 54–58. AAAI Press, 2006.

[Dey and Abowd, 2000] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Procs. of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness*, 2000.

[Dourish, 2004] Paul Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1):19–30, 2004.

[Ghidini and Giunchiglia, 2001] Chiara Ghidini and Fausto Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[Giunchiglia, 1992] Fausto Giunchiglia. Contextual reasoning. Technical Report 9211-20, Istituto per la Ricerca Scientifica e Technologica, 1992.

[Guarino, 1998] Nicolas Guarino. Formal ontology and information systems. In *Procs. of the 1st International Conference on Formal Ontologies in Information Systems (FOIS'98)*, pages 3–15, 1998.

[Kent, 2004] Robert E. Kent. The information flow framework: A descriptive category metatheory. In *Procs. of the International Category Theory Conference (CT04)*, 2004.

[Kopylov, 2003] Alexei Kopylov. Dependent intersection: A new way of defining records in type theory. In *Procs. of the 18th Annual IEEE Symposium on Logic in Computer Science*, pages 86–95, 2003.

[Martin-Lof, 1982] Per Martin-Lof. Constructive mathematics and computer programming. In *Procs. of the 6th International Congress for Logic, Methodology and Philosophy of Science*, pages 153–175, 1982.

[McCarthy, 1993] John McCarthy. Notes on formalizing context. In *Procs. of the 13th Int. Joint Conf. on Artificial Intelligence*, pages 555–560, 1993.

[Montague, 1970] Richard Montague. Pragmatics and intensional logic. *Synthèse*, 22:68–94, 1970.

[Mügge *et al.*, 2006] Holger Mügge, Tobias Rho, Daniel Speicher, Julia Kuck, and Armin B. Cremers. Towards an infrastructure for context-sensitive intelligence. In *SAKS Workshop on Context-sensitive Intelligence*, 2006.

[Paulson, 1989] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, 1989.

[Peleg *et al.*, 2003] Mor Peleg, Samson Tu, Jonathan Bury, Paolo Ciccarese, John Fox, Robert A. Greenes, Richard Hall, Peter D. Johnson, Neill Jones, Anand Kumar, Silvia Miksch, Silvana Quaglini, Andreas Seyfang, Edward H. Shortliffe, and Mario Stefanelli. Comparing computer interpretable guideline models: A case-study approach. *Journal of the American Medical Informatics Association*, 10(1):52–68, 2003.

[Ranta, 2004] Aarne Ranta. Grammatical framework: A type-theoretical grammar formalism. *Journal of Functional Programming*, 14(2):145–189, 2004.

[Strang and Linnhoff-Popien, 2004] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *Sixth International Conference on Ubiquitous Computing (UbiComp2004)*, pages 34–41, 2004.

[Strang *et al.*, 2003] Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. Cool: A context ontology language to enable contextual interoperability. In *Procs. of DAIS 2003*, number 2893 in LNCS, pages 236–247, 2003.

[Thomason, 1998] Richmond H. Thomason. Representing and reasoning with context. In *Procs. of the International Conference on Artificial Intelligence and Symbolic Computation*, number 1476 in LNCS, pages 29–41, 1998.

[Thomason, 1999] Richmond H. Thomason. Type theoretic foundations for context, part 1: Contexts as complex type-theoretic objects. In *CONTEXT*, number 1688 in LNCS, pages 351–360. Springer, 1999.

[Turner and Stevenson, 1991] Roy M. Turner and Robert A.G. Stevenson. Orca: An adaptive, context-sensitive reasoner for controlling auvs. In *Procs. of the 7th Int. Symp. on Unmanned Untethered Submersible Technology (UUST '91)*, pages 423–432, 1991.

[Uschold and Grüninger, 1996] Mike Uschold and Michael Grüninger. Ontologies: Principles, methods, and applications. *The Knowledge Eng. Review*, 11(2):93–155, 1996.

[Villadsen, 2004] Jørgen Villadsen. Multi-dimensional type theory: Rules, categories and combinators for syntax and semantics. In *Int. Workshop on Constraint Solving and Language Processing*, pages 160–165, 2004.

[Wang *et al.*, 2004] Xiao H. Wang, Tao Gu, Da Q. Zhang, and Hung K. Pung. Ontology based context modeling and reasoning using owl. In *Procs. of the 2nd IEEE Conference on Pervasive Computing and Communications (PerCom2004)*, pages 18–22, 2004.

# Context-aware probabilistic reasoning for proactive healthcare

**Rohit Joshi, Li Guoliang, Tze Yun Leong**
School of Computing, National University of Singapore,
3, Science Drive 2, Singapore 117543
{rjoshi, ligl, leongty}.@comp.nus.edu.sg

## Abstract

Decreasing costs and sizes of sensors, and advances in artificial intelligence motivate novel use of technologies in providing a better quality of health care to the patients. The notion of "context" is increasingly becoming critical in capturing diverse situations for providing patient-specific proactive health care. However, the inherent uncertainty in the sensed context makes the domain challenging. Bayesian networks are a popular framework for dealing with uncertain information. Representations and inferences to exploit contextual dependencies and context uncertainty are needed to effectively support analytical tasks in diverse situations for a particular proactive service. In this paper, we show that context-aware probabilistic reasoning can facilitate the successful integration of these situation scenarios together into one single framework for efficient reasoning. Furthermore, our representation of such contextual dependencies allows compactness, context-specific personalization of models and determination of contextually irrelevant information for possibly achieving performance improvement under resource constraints.

## I. Introduction

With rising health care crisis such as the shortage of caretaking professionals and the increasing episodes of the chronic diseases and cognitive disabilities, it is becoming a challenge to provide personalized care at hospitals. Decreasing costs and sizes of sensors, equipments and advances in artificial intelligence motivate novel use of technologies in enhancing quality of health care and facilitating personalized care outside the hospital domain. The vision of proactive healthcare [Intille, 2004] is to provide more healthcares outside the traditional clinical settings in the patient's home and in a proactive manner rather than reactive manner. Applications may include monitoring patient's daily activities, drug intake, early detection of chronic disease and reacting to emergencies.

Bayesian networks (BNs) are a popular framework for reasoning with the uncertain sensor information [Korpopaa *et al.*, 2003; Philipose *et al.,* 2004]. BNs can be applied to infer a patient state at home given the sensor readings and patient information. However, to provide quality healthcares at home, several patient situations may need to be inferred. We may need several small BNs for inferring each patient condition. For example, we may have one BN to estimate the number of persons in the house, one to predict the patient's current activity, one to sense the symptoms the patient is currently having and yet another BN to forecast the outcomes of a disease.

Context-aware reasoning may be particularly useful for supporting analytical tasks in diverse situation for a proactive service. For example, a proactive service for an elderly patient may require adapting to a particular situation and inferring different patient health conditions depending upon patient's situation, say a person being alone, with a caretaker, or with many people around. The sensed situation, such as "the person alone", may involve inherent uncertainty. Therefore, to capture the uncertain information in different situations, we can either use multiple BNs, one each for a particular situation or construct a large BN from all these possible situations. However, reasoning with multiple BNs may be too computationally expensive under resource constraints and reasoning with one large BN may be inefficient and inflexible. Context-awareness may facilitate compact representation of these situations in one single framework while providing appropriate model adaptation depending upon the context of the situation. The context is used to characterize the reasoning dependencies that exist in the model in different situations. Such context-sensitive knowledge leads to contextual dependencies that exist only for a specific value of a context variable. The probabilistic independence that results gives rise to finer-grain factorization of the joint probability distributions and has been termed as Context-specific independence (CSI) [Boutilier *et al.*, 1996].

In this paper, we present a potential application of context-aware reasoning for proactive healthcares and explore the advantages of context-sensitive network (CSN) framework [Joshi and Leong., 2006] to reason with such context-sensitive knowledge under situational uncertainty. We use the context as a probabilistic tag to represent context-aware situations. We show the representational advantage of the CSN framework in compaction and model adaptation, the flexibility in personalizing the model in a specific context and the easiness in determining the contextually irrelevant information. We demonstrate the potential usefulness of the framework for context-aware reasoning by examining the impact of context-sensitive information on a real-life heart disease dataset.

## II. Preliminaries

**Bayesian Network:** A BN is an acyclic directed graph capturing the factorized structure resulting from conditional independence among the variables. Each variable *X* in a BN is associated with a set of conditional probability distributions (CPDs), encoded in a conditional probability table (CPT) of the form *P(X|**Pa**(X),* where ***Pa**(X)* represents the set of parent variables (factors) on which the target or consequent variable *X* is depended on.

**Context:** Context **C**, in our work, means a state or an assignment of one value to each variable in set **C** and characterizes the dependencies that exist in the model [Boutilier *et al.*, 1996]. Context can be any information used as a probabilistic tag to index the models, for example, patient profile, activity, environment, or symptoms. In Figure 1a, Age is one such context variable. If Age >60, CAD is independent of Gender. The context variable must be a parent variable to model the contextual dependencies of the child variable. A context variable can also depend probabilistically on another variable, i.e. the context variable does not necessarily has to be the root variable.

**Context-specific Independence (CSI):** Context-specific independence is conditional independence that exists in a specific value of the context variable. Conditional independence is simply context-specific independence that exists in all contexts. CSI further factorizes a CPT into a product of probability factors using the probabilistic independence property, which reduces the parent set in a specific context. Qualitative structure of a BN is not flexible to capture such context-specific factorization. Formally, CSI is defined as follows [Boutilier *et al.*, 1996]: *Let **C** be a set of variables. Let a context be denoted by **C=c** and **c** refers to set of values of variables in **C**. Let **X, Y, Z**, and **C** be four disjoint sets of variables. **X** and **Y** are independent given **Z** in Context **C**=**c** if P(**X**|**Z, Y, C**=**c**) = P(**X**|**Z**, **C**=**c**).*

## III. Problem Scenario

**Example 1:** *An 87-year old woman is being discharged from hospital after being admitted for an acute myocardial infarction (heart attack). The hospital course was uncomplicated and she is back to her baseline status. Her past medical history is indicative of osteoporosis and a fractured hip 2 years ago. Patient has 1 son who lives several hours away. Dr Smith, her physician, regularly sees such patients and will like to manage their conditions proactively using the emerging technologies for smart homes. His objective is to predict different patient health conditions such as the chances for coronary artery heart disease (CAD) and possible injury to the patient at home in a situation-specific manner. He has some situation-specific guidelines to encode when it is desirable to predict CAD and Injury. Furthermore, Dr Smith would like to predict CAD using the relevant factors depending upon patient's profile such as Age and Ethnicity. Figure 1a and 1b show two such situations for a Caucasian elderly person above 60 years of age.*

***Situation 1: Patient is alone:*** *If the sensors sense that the patient has a hand on chest (indicating a chest pain) and has fallen down, predict the outcome of CAD using a guideline BN in Figure 1a.*

***Situation 2: Care taker is at home.*** *Predicting a possible Injury is important if the patient falls. However, in the situation the care taker is present to handle the emergency, predicting CAD is not required. A Hand-on-Chest may just be a gesture. Figure 1b shows that variable Chest and Fall are not dependent on CAD in this specific context.*

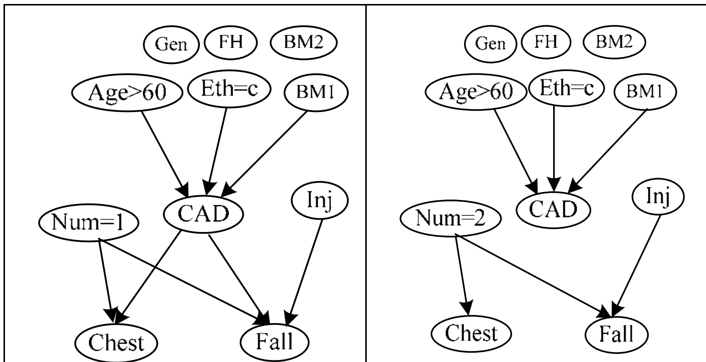



**Figure 1a**: Situation 1: specific context: (N=1, A>60, E=c)

**Figure 1b**: Situation 2 : specific context: (N=2, A>60, E=c)

**Legend:** Age (A), Ethnicity (Eth), Gen: Gender, FH: Family History, BM1, BM2: Biomarkers, Chest: Hand on Chest, Num: Number of people, **Context**: Age, Eth, Num

Here we assume that there is an inherent uncertainty in estimating the number of people in the house, and sensing the user profile and symptoms. There are three binary context variables in this example: Number of People, Age, and Ethnicity. We need $2^3$ (=8) Bayesian Networks to represent 8 possible situations to capture the complete uncertain knowledge. We would like to address the following questions:

a) *How can multiple context-specific reasoning networks be used together under context uncertainty?*
b) *Can a single reasoning model be flexible to use appropriate nodes on context switching and facilitate run-time model personalization?*

One way to encode multiple situations is to use the context as a deterministic tag using the techniques proposed in [Ngo *et al.*, 1997; Mahoney *et al.*, 1998]. Although these approaches solved the problem to a certain extent, they have the assumption that the sensed context has to be deterministic in an application and a root node in a BN. Representations capturing uncertain context tags are needed, as there can be an inherent uncertainty in many sensed contexts in our domain for example, "person being alone". One solution could be to use the Multinets approach [Geiger *et al*., 1996]. This approach requires one BN for each configuration of the context variables [Geiger et al] and use a complete context set as a root node. For example, eight BNs are required for three binary context variables. This may be too expensive under resource constraint because the number of BNs required can be exponential. Furthermore, if the evidence about the context is not given, it may be inefficient to reason with several BNs. For example, without the context information, a query on the probability of CAD given Fall in Figure 1 i.e. P(CAD|fall) will need all the BNs.

Alternatively, we could have encoded the uncertain information in all the possible situations in one large Bayesian network as shown in Figure 2, and possibly use a special data structure for the CPTs such as a tree CPT [ Boutilier *et al.*, 1996, Poole *et al*., 2003] to exploit CSI only for the inference. However, CSI is not represented graphically in the BN, so it is not easy to exploit situation-specific model simplification, resulting in a loss of context sensitivity. We would need to reason with the static dependency structure in all the situations. In Figure 2, we have purposely labeled the arcs to show that the dependencies exist only in that specific context. Contextual independence exists in this problem. For example, P(CAD|Age>60, Eth=c, Gen, FH, BM1, BM2) can be succinctly expressed as P(CAD| Age>60, Eth=c, BM1). CSN representation [Joshi and Leong, 2006] allows a way to exploit the contextual information in one single framework under situational uncertainty, and yet take the advantage of context-sensitivity under a specific context.

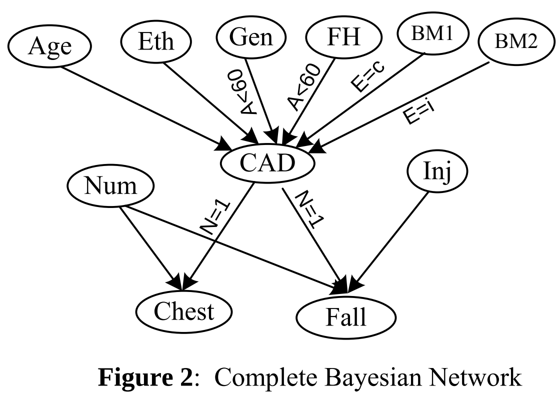



**Figure 2**: Complete Bayesian Network
Labels: A: Age, E: Eth, N: Num

## IV. Context-Sensitive Network

Representing the context-sensitive knowledge imposes an additional structure in factorization resulting from context-specific independence [Boutilier *et al.*, 1996]. Such context-sensitivity allows decomposition of some of the conditional probabilities, giving rise to a finer-grain factorization of the joint probability distributions. To represent the structure of further factorization from CSI, Joshi and Leong [Joshi and Leong, 2006] represented the CPT functions along with the variables on the graph itself, converting it into a bipartite representation.

Joshi and Leong proposed a new graphical representation, called *Context-Sensitive Networks (CSNs)* for representing and reasoning with contextual independence in terms of *Conditional Part Factors (CPFs)*. A CSN is a two-level bipartite graphical model representing the product of Conditional Part Factors (CPFs). The CPF is a new internal representation in CSNs that serves as the counterpart of a CPT in BNs, for distributions that exhibit contextual independence. A CPF is viewed as the partition of a full CPT which is valid in a specific context, denoted as $\Phi(X|\mathbf{Y})^{I(C=c)}$, where X is a variable, $\Phi$ is the conditional factor on X and **Y** is a set of parent variables of X in context **C**=**c** with the indicator function I. The CPT of a BN can be written as a product of disjoint sets of the CPFs. CSNs can be viewed as a special subclass of the recently proposed probabilistic graphical models - Directed Factor Graphs (DFGs) [Frey, 2003], which are in turn generalizations of BNs.

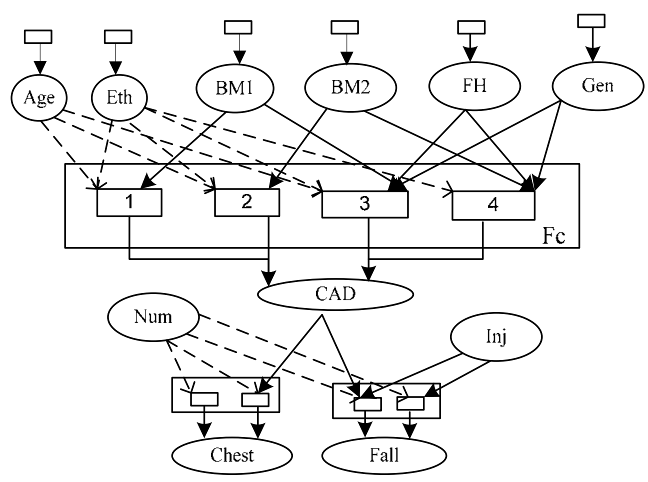



**Figure 3**: CSN equivalent to Figure 2

### Representing different situations in CSN

A CSN, as shown in Figure 3, consists of variable nodes (shown as ovals), context function nodes (small rectangles) and function nodes (big rectangles with small rectangles inside). A context function node represents a local CPF. Each CPF can be viewed as representing the CPT of a variable in a specific context. A function node can be viewed as a collection of the CPFs in all possible context values. The representation in Figure 3 is in fact equivalent to the Figure 2 with labels on the edges. In Figure 3, nodes 1-4 are the context function nodes with the CPFs representing four possible situations for the two context variables, Age and Ethnicity. We use dashed arrows in Figure 3 just to show the dependencies connecting the context variables and CPFs. Normal arrows show the dependencies in that context.

CSN allows compact and adaptive representation of the contextual information while allowing both general and context-sensitive information to be represented in one single CSN. In Figure 3, Age (A) and Ethnicity (E) are two binary context variables for CAD (C) and there are four relevant contexts: (A<60,E=c), (A<60,E=i), (A>60,E=c) and (A>60,E=i). There are four different CPFs for the variable CAD: $\Phi(CAD|BM1)^{I(A>60,E=c)}$, $\Phi(CAD|BM2)^{I(A>60,E=i)}$, $\Phi(CAD|Gen,FH,BM1)^{I(A<60,E=c)}$, $\Phi(CAD|Gen,FH,BM2)^{I(A<60,E=i)}$ The switching between these four CPFs depends on the values of the context variables. Variables Chest and Fall are associated with two CPFs each, covering all 8 possible situations in Example 1. In Figure 1a, the variable 'Fall' has 3 parents while in Figure 1b 'Fall' has just 2 parents. Representing the context-sensitive information also needs this flexibility in shifting the size of CPTs without the need of expensive manipulations.

## Context as probabilistic tag

The context variable is represented as a random variable with different possible states representing each situation. Multiple situations are compactly represented by tagging each specific situation with a particular value of the context variable. For example, the number of people, age and ethnicity are three context variables that tag all the eight possible situations in our example. Conditional Part Factors (CPF) represents the CPTs of a variable in a particular situation. Each CPF is associated with an indicator function with the context variables as arguments. The indicator function of each CPF has values 1 or 0 if the context is fully observed. If the context profile is incomplete, the CPFs satisfying that profile will have a value 1. For example, if only Age>60 is observed and Ethnicity is unknown, then the node 1 and 2 in Figure 3 will evaluate to 1 while node 3 and 4 will not be involved in further inference. If all the contexts are uncertain, then the full network with all the nodes is used for inference.

In addition, the context variable can depend probabilistically on another context and does not need to be a root node. For example, the probability of reading an alert may depend upon diverse device parameters in different situations based on the Activity context, which in turn can be dependent on the Location context. The context variable is just another random variable in the CSN framework, so the inference algorithm can automatically handle such scenarios.

## Context-specific model personalization

The CSN framework allows flexibility in model adaptation for the context-specific personalization. Figure 3 shows a generalized model where the model can reason with all the possible contexts. The evidence from any context variable can allow constructing more personalized models in a specific context. By context-specific personalization of the model, we mean that the same model representation should be flexible to change its dependency structure and allow execution of the different models based on the evidence about the context. Such personalized models can be much more effective in terms of both memory and speed as the graphical model is much simpler. CSN with its ability to work with the CPT partitions and context tags does context-specific personalization effectively without any expensive manipulations. Figure 4 shows a contest-specific personalized model as created in our framework for situation 1 in the example scenario. As the evidence about the context is given, only the indicator functions for node 1 for CAD, and a node each for Chest and Fall evaluate to 1. The model has removed dependency of gender, family history and some biomarkers as in this context they are contextually irrelevant variables for predicting CAD.

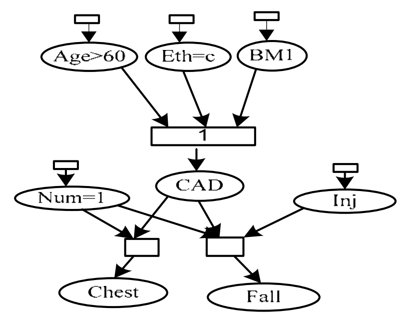



**Figure 4**: Context-specific personalized model

## Determining Contextually Irrelevant Information

Determining the contextually irrelevant variables is quite useful as it not only simplifies the reasoning model but also may decrease the cost of observations related to obtaining the irrelevant sensor values. For example, if the location of the person is known, say in a living room, only the information from sensors located around the living room would be contextually relevant to determine the current activity of the person, and the information from other sensors may be contextually irrelevant. CSN represents the CSI graphically, which allows an easy determination of the contextually irrelevant models for an observed context value The CPFs and respective dependencies that are not valid in a particular context can be removed from the graphical structure. For example, in Figure 3, nodes 2, 3 and 4 can be removed. Therefore, the nodes Gender, FH and BM2 will not influence CAD in this situation. If any of the nodes Gender, FH or BM2 had parent variables, they would also be contextually irrelevant to CAD.

### Supporting situation-specific Inference

Situation-specific inference requires an ability to reason with both the general and context-specific information. If all the context variables are fully observed, any off-the-shelf BN algorithm can be used on the personalized model. However, if the context is incomplete or uncertain, [Joshi and Leong, 2006] has proposed a new belief propagation algorithm that works naturally and efficiently over CSNs to support different types of inference by utilizing context-specific independence. CSN can effectively support situation-specific inference, preserving both the generality and the context-specific representations, and the afforded efficiency.

Here, we summarize the three operations defined for the message passing algorithm on the CSNs [Joshi and Leong, 2006]: a) *context-sensitive factor product*, b) *context node marginalization*, and c) *context marginalization*. The context-sensitive factor product multiplies the CPFs and incoming messages conditioned on the value of the context of a context function node. The context node marginalization marginalizes over all the variables that are not in the context of a context function node except the variable to which message is being passed to. Finally, the context marginalization sums over all the messages received to marginalize out the variables in a context. In the nutshell, each context function node receives messages from its neighbors, performs a context-sensitive factor product with its local CPF, and performs the *context node factor marginalization* for all the variables not in context. Then, the function node performs *context marginalization* on the messages received from all the context function nodes that are a part of it. For the uncertain context, the inference algorithm reasons with all the possible context situations and performs the context marginalization over all the messages received from the context function nodes. It will marginalize messages from nodes 1-4 in Figure 3. For observable context, only one particular CPF's indicator function will have value 1. Hence only 1 context function node will send the message to the function node. For example, in Figure 3, only node 1 sends the message to the variable C.

A major advantage of the CSN framework is that it not only supports a context-aware inference but also reduces the computational complexity in message passing. In general, the maximum factor width, the scope of the factor, can be much smaller in a CSN than in a BN. CSN utilizes the finer-grain factorization of the joint probability distributions, resulting in a fewer number of parameters as well as the number of calculations required for a message than a comparable BN. We are currently working on exploiting this for better inference algorithms.

## V. Case Study

Coronary artery disease (CAD) is the most common cause of sudden death in most developed countries. Chest pain is the chief complaint in CAD. However, CAD is often poorly diagnosed as the chest pain spans a wide clinical spectrum of diseases from the trivial to a life threatening. So, the patients with unstable angina (chest pain) can inadvertently be discharged from an emergency department and adverse outcomes in these patients represent a significant cause of death [Pope *et al.*, 2000]. Patient-specific proactive healthcare systems for forecasting the outcomes for a disease such as CAD based on the demographic and prior history information while incorporating the sensed environmental information can be extremely useful not only at homes but also in the emergency departments. For example, the proactive care systems can intelligently monitor vulnerable patients for the signs such as an increased blood pressure, glucose level, unstable heart activity, chest pain duration, continuous exposure to smoke, current patient state such as unconscious fall to predict the outcomes and send alerts to relevant caretakers in case of an emergency. We demonstrate the potential usefulness of the framework of context-aware reasoning for the proactive care by examining the impact on a simulated study over a real-life Coronary artery disease dataset.

We use a Coronary Artery disease (CAD) dataset of about 2950 patients from a local hospital. The dataset contains 43 variables, of which 11 describe patient's profile such as Gender, Race, Age and the other 32 specify gene biomarkers. The context variables are used to capture the situation-specific guidelines. We took user profile variables AGE and RACE as the context variables based on the expert knowledge and assumed context-specific independence in the distribution. Usually, males below 60 years of age are at a greater risk of having the CAD if they have a prior family history than the females in the same age group. In addition, the person's race determines what genetic biomarkers will be useful for the prediction. An important variable for predicting the outcomes in the reasoning network is CAD. We implemented the proposed framework in Matlab and tested it on a Pentium M 1.5Ghz with 384MB RAM.

We compared the compactness achieved by using "context" in this domain, and advantages of context-specific personalization of reasoning models using contextual dependencies in CSN. We constructed four different situations based on the context variables. The variable CAD has overall 9 binary parents, representing the 9 possible causes, each of which is valid only in a particular situation. These situations result in four CPFs for the variable CAD in the CSN. Each CPF is compact due to the context-specific independence. The CSN only requires 40 entries for CAD. If a large BN were deployed to capture the probabilistic context, the conditional probability table in BN for CAD would have required a total of $2^{10}$ i.e. 1024 entries for representing the same probability distribution. Furthermore, the personalized CSN model with observed Age and Race as in situation 1 of problem scenario required only about 10 variables in this situation instead of all 43 variables. Gender, family history and several of other gene variations were contextually irrelevant in this situation. Reasoning with 10 variables is much easier in small and resource constraint devices. This result is not surprising as it was expected that CSN representation would do run-time personalization and render many variables contextually irrelevant.

We also compared runtime with a comparable BN on the worst case scenario for reasoning over full network with no context observation. We conjectured that the comparison in the worst case is more interesting, because given a context observation, our representation will have a smaller dependency structure and will inference much faster than the comparable BN. We have only compared our proposed algorithm with the belief propagation on BN as the two algorithms are equivalent. Two different BNs were learnt: first using Bayesian Hill Climbing structural learning algorithm and second using domain knowledge and constraint elicitation from experts (Network 2). For testing speed and accuracy, we assumed contextual independence in variables with more than 3 parents and partitioned the CPT accordingly. Marginal of all the nodes were exactly similar within 0.01 tolerance limit. On Network 1, the inference on a BN took 3.5 seconds while on the CSN took 1.75 seconds. For Network 2, the BN took 3.2 seconds while CSN took only 1.89 seconds. The belief propagation on CSN has shown some promising results.

## VI. Conclusion

Context-aware reasoning can be particularly useful for the adaptive reasoning tasks for proactive and ambient intelligence applications. To provide quality proactive healthcares at home, several patient situations may need to be inferred. Context-sensitive network representation has shown promising advantages to integrate these multiple situations and exploit contextual dependencies and context uncertainty for effectively supporting analytical tasks in diverse situations. In this paper, we demonstrated the advantages of the CSN framework in representation compactness, model personalization and contextually irrelevant information determination in the healthcare domain. The concepts are equally applicable to other ambient intelligence applications too. A possible limitation in our work is the approximate algorithm, loopy belief propagation. However, the loopy belief propagation has been quite successful in many real-world applications because it is relatively simple to implement and usually converges to an exact value. Our ongoing and future works include evaluating the impact of CSN framework with the increased number of context, proposing better inference algorithms and prototyping a proactive health monitoring application in future.

## References


[Boutilier *et al.*, 1996] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller, "Context-specific independence in Bayesian Networks," *In Proc. of 12th Conf. UAI*, 1996.

[Geiger *et al.*, 1996] D. Geiger, and D. Heckerman, "Knowledge representation and inference in similarity networks and Bayesian multinets," *Artificial Intelligence*, vol. 82, pp. 45–74, 1996.

[Frey, 2003] B. J. Frey, "Extending factor graphs so as to unify directed and undirected graphical models," *In Proc. of 19th Conf. UAI*, 2003.

[Intille, 2004] S. S. Intille, "A new research challenge: persuasive technology to motivate healthy aging," *Transactions on Information Technology in Biomedicine*, vol. 8(3), pp. 235-237, 2004.

[Joshi and Leong, 2006] R. Joshi and T.Y.Leong, "Patient-specific Inference and Situation-dependent Classification using Context-sensitive Networks," *Proc. AMIA Symp*, 2006.

[Korpipää *et al.*, 2003] P. Korpipää, M. Koskinen, J. Peltola, S. M. Mäkelä, and T. Seppänen, "Bayesian approach to sensor-based context awareness," *Proc.Ubiquitous Computing,* vol. 07, pp.113-124, 2003.

[Kschischang *et al.*, 2001] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498-519, 2001.

[Mahoney *et al.*, 1998] Mahoney, S.M. and K.B. Laskey , "Constructing Situation-Specific Belief Networks," *Proc. of 14th Conf. UAI*, pp. 370-378, 1998

[Ngo *et al.*, 1997] L. Ngo and P. Haddawy, "Answering Queries from Context-Sensitive Probabilistic Knowledge Bases," *Theoretical Computer Science*, 1997.

[Philipose *et al.*,2004] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, D. Hahnel, "Inferring Activities from Interactions with Objects," *IEEE Pervasive Computing*, vol. 03, pp. 50-57, 2004

[Poole *et al.*, 2003] D. Poole and N. Zhang, "Exploiting contextual independence in probabilistic inference," *J. of Artificial Intelligence Research*, vol. 18, pp. 263-313, 2003.

[Pope *et al.,* 2000] J.H. Pope, T.P. Aufderheide, R. Ruthazer *et al.*, "Missed diagnosis of acute cardiac ischemia in the emergency department," *N Engl J Med*; vol 342, pp. 1163-1170, 2000.

# Enhanced Healthcare Provision Through Assisted Decision-Making in a Smart Home Environment

**JC. Augusto** and **P. McCullagh**
School of Computing and Mathematics
University of Ulster, Jordanstown
BT37 0QB, Northern Ireland (UK)
jc.augusto@ulster.ac.uk

**V. McClelland** and **J-A. Walkden**
Ulster Community Hospitals Trust
Newtownards, BT23 4AD,
Northern Ireland (UK)
julie-ann.walkden@ucht.n-i.nhs.uk

## Abstract

New technological developments are shaping alternative forms of healthcare provision in the home. We consider the problem of assisting elderly people and their carers during the night to reduce the occurrence, and potential side effects, of accidents suffered by vulnerable people.

The ambient assistance provided will assume minimal expectations on the technology people have to interact with. The technological solution considers the inclusion of night vision which provides features assessed by an algorithm based on the notions of causality and spatio-temporal reasoning. The system alerts, explains to, and advises carers, assessing situations, anticipating problems, and has potential to enhance home care provision.

**Keywords:** Smart Homes, Context Awareness, home safety, Intelligent handling of privacy.

## 1 Introduction

The population of the world is ageing. In 2000, there were 600 million people aged 60 and over; there will be 1.2 billion by 2025 and 2 billion by 2050 [WHO, 2006]. Medicine, as many other areas of our society, has benefited with the increase of the availability of information and additional use of technology that can innovate or extend the ways in which the person's health is managed. The demographic exponential growth of the elderly population and its concomitant requirements for care is likely to continue; making it practically impossible to perform extensive person to person follow ups and monitoring of patients' health. Elderly people are particularly affected by this lack of available human resources because the ageing process leads to a propensity to suffer chronic conditions. These conditions demand close monitoring and additionally the likelihood of being affected by hazards at home increases [DofH-UK, 2004]. One group of the ageing population which is particularly vulnerable to loss of independence are those affected by dementia. Some of the latest developments to increase independence of elderly people at home are related to the use of Smart Homes (see for example [Nugent and Augusto, 2006]).

Advances in sensor technology, wireless communication and the use of remote monitoring offer the opportunity to assist elderly people at home, as advised by The World Health Organisation [WHO, 2004], and they can maintain independence reducing the prospect of being cared for in a nursing home. There is also a growing tendency to decentralize healthcare provision across Europe moving away from secondary and tertiary care models and enabling primary and community care to take the lead ([TheHIP Group, 2005; RSPA, 2005]).

In this paper we consider the problem of assisting vulnerable elderly people and their carers during the night to reduce the occurrence and minimise potential side effects of home accidents. The ambient assistance provided will assume minimal expectations on the technology patients have to interact with. The solution includes night vision processing which is assessed by an algorithm based on the notions of causality and spatio-temporal reasoning. The system alerts, explains and advises carers by anticipating problems, assessing situations, and therefore has the potential of making the healthcare provision more efficient (both safer and more economic).

Section 2 explains how Ambient Intelligence, and in particular Smart Homes, can be used to improve a person's quality of life. We focus on the problem of enhancing the quality of nocturnal care which can benefit both patients and carers. Section 3 explains how a given context can be represented in our system, how the events occurring in that context will be monitored until a situation of interest is detected and how the system will consider those options to present them to the carer. Section 4 explains the practical relevance systems like the one presented here have for the real world. Finally section 5 explains why similar systems do not address properly the specific class of problems we address here: the improvement of nocturnal care provision for both elderly patients and their carers with no technological burden on the patient.

## 2 Technologically Assisted Healthcare

### 2.1 Ambient Intelligence and Smart Homes

Ambient intelligence (AmI), see for example [IST Advisory Group, 2003; Bailly *et al.*, 2005], is an Artificial Intelligence (AI)-based paradigm with increasing potential to make an impact in daily human life in the near future. The broad idea is to enrich a space (e.g., a house, a building, a bus station, a critical area in a hospital, etc.) with sensors so that through their use an automated or semi-automated decision-making system can provide benefits to those using that space.

One of the most well-known developments in the area of Ambient Intelligence is usually referred as a "Smart Home" where a house is equipped to bring advanced services to its users. Expected benefits can be: (a) increasing safety (for example by monitoring lifestyle patterns or the latest activities and providing assistance when a possibly harmful situation is developing), (b) comfort (for example by adjusting temperature automatically, etc.), (c) economy (for example controlling the use of lights), and other benefits according to the domain of application. Recent applications include the use of Smart Homes to provide a safe environment where people with special needs can have a better quality of life. For example in the case of people with dementia (the most frequent case being elderly people suffering from Alzheimer's disease) the system can be tailored to minimize risks and ensure appropriate care at critical times by monitoring activities, diagnosing interesting situations and advising the carer.

Research into Smart Homes is a multidisciplinary area in which software will inter-relate (a) sensors, (b) the architecture of a building, (c) a network for data transportation, (d) and different levels of users. Several artifacts and items in a house can be enriched with sensors to gather information about their use and in some cases even to act independently without human intervention. These modified elements form the perception and actuation levels of the intelligent system. Some examples of such devices are electrodomestics (e.g., cooker, tv, fridge, microwave) household items (e.g., taps, bathroom), house temperature handling items (e.g., air conditioning and radiators), and other accessories (e.g., bed, doorbell, phone and emergency pull cord switch).

### 2.2 Smart Homes Applied to Nocturnal Healthcare

Whilst there had been research conducted in the use of Smart Homes to enhance the quality of life for senile dementia patients, the focus has been on the most active period of the day [Nugent and Augusto, 2006]. Here we address an equally (or more) important period which has been neglected: the night time with its associated problems. Although there are numerous causes of concern associated to nocturnal care. Here we focus on the propensity to suffer falls [Shaw *et al.*, 2003]: a) they are a major cause of disability and loss of independence, causing fractures, head injuries, pressure sores, depression and social isolation, b) they are the main cause of accidental death in the elderly.

Processing images from video cameras can be used to assess whether the person is moving appropriately and performing activities of daily living or whether a fall or undue inactivity has been detected. This approach is innovative and in many cases (depending on the severity of the illness) may remove the need for body worn devices. Night vision and thermal image devices for movement and temperature monitoring can be fundamentally helpful (see for example an application on street surveillance at [Desurmont *et al.*, 2005]). The images obtained can have sufficient quality to facilitate segmentation and object identification, but be insufficient to provide a clear picture (providing a level of intimacy safety). With the use of feature extraction no human monitoring of these images will be required under normal circumstances in practice. It is recognised that image capture will raise issues of privacy, and will have ethical considerations, which may influence user acceptance. This poses significant ethical considerations. However statistics show that 90% of elderly people want to remain in their own home [Heaney, 2006], and often the citizen leads the professionals in evolution of the e-society.

The context of our research is the use of Smart Homes by a Health and Social Services Trust which is undertaking a major restructuring of its services by introducing Smart Homes technology to enhance the quality of the care provision delivered to the elderly population and reduce hospital and nursing/residential home admissions. A problem of concern for which there are no proposed solutions so far is the provision of intelligent nocturnal care which is beneficial to the patients and which is cost-effective. An additional aim in this context is improving the quality of life of the carers. Very often the carer will be a close relative who cohabits or lives in a house nearby. As the person being cared for deteriorates and becomes more dependent, the carer's quality of life deteriorates too. For example, they will suffer interrupted sleep and sleep deprivation [Grace *et al.*, 2000].

There are many aspects of daily life that can be considered and many sensors involved we focus here on a few sensors which allows: patient tracking and time measurement, bed occupancy detection, lights use and finally image processing as the technology to be used to diagnose day-night adjustment and falls. See [Nugent and Augusto, 2006] for example of sensors and their applications.

## 3 Assisting Decisions

### 3.1 Knowledge representation

Smart Homes allow the decentralization of services diversifying the healthcare provision at home. Smart Homes technology greatly depends on the possibility to locate and follow a person through spatio-temporal references and to relate those activities to expected patterns of behaviour. Here areas of AI such as complex events detection, spatio-temporal reasoning and causal reasoning are of fundamental importance (see for example [Augusto and Nugent, 2004; Augusto, 2005] for more details). Knowing which activities are being developed and being able to assess them from a specific perspective are important aspects identified with monitoring and diagnosis. To achieve this task, a causal theory of the system is needed to relate the important events contributing to a specific scenario and their effects. Once the system detects something interesting it provides fundamental information for the human carer to decide a course of action, if needed. Here explanations are very valuable to assist human decision making. The explanations of our system are based on set of causal rules which makes reference to the places and times of the activities developed leading to a particular scenario of interest within the Smart Home (e.g., a specific hazard).

The bibliography on causal reasoning is vast and due to space restrictions we do not cover it in detail in this paper. Instead of a general theory for causality we use a simple notation, which we will refer as $C$ and is borrowed from [Galton and Augusto, 2002], which will suffice to represent and reason about causally related states and events in a Smart Home. $C$ allows a simple specification of the main ingredients to represent causal relationships between elements of a domain. It distinguishes between *independent* ($\mathcal{S}_I$) and *dependent* ($\mathcal{S}_D$) states. The former are states whose value can change at any time, outside the control of the system and therefore it cannot be anticipated when they can change. They typically represent the environment and in our case they will represent the state of sensors being stimulated by the activities in the house. Dependent states have known state change dynamics and therefore they can be expressed in the heads of causal rules as an effect of the system reaching other states.

**Definition (syntax)** ([Galton and Augusto, 2002] page 3): There are two kinds of *causal rule*,

*Same-time rules*: $S_1 \sqcap S_2 \sqcap \cdots \sqcap S_n \rightsquigarrow S$
*Next-time rules*: $S_1 \sqcap S_2 \sqcap \cdots \sqcap S_n \rightsquigarrow \bigcirc S$

where each $S_i$ is an atomic state and $S \in \mathcal{S}_D$.

**Definition (semantics)**: lets assume a set of $\mathcal{S}$ of states $S_0, S_1, S_2, \ldots$ and a time $t$, $t \in Nat$, then by $S_t$ we represent the formulas which are true in the world at the state it is at time $t$:

$(\mathcal{S}, t) \models S$ iff $S \in S_t$
$(\mathcal{S}, t) \models -S$ iff $S \notin S_t$
$(\mathcal{S}, t) \models S_1 \sqcap S_2$ iff $S_1 \in S_t$ and $S_2 \in S_t$
$(\mathcal{S}, t) \models S' \rightsquigarrow S$ iff $(\mathcal{S}, t) \models -S'$ or $(\mathcal{S}, t) \models S$
$(\mathcal{S}, t) \models S' \rightsquigarrow \bigcirc S$ iff $(\mathcal{S}, t) \models -S'$ or $(\mathcal{S}, t+1) \models S$

Same-time rules are required to be *stratified*, i.e., they can be ordered in such a way that the states in a rule are either independent or dependent which have been in heads of rules defined at previous levels (see [Galton and Augusto, 2002] for the formal details).

The implementation uses ASCII symbols which are related to those of $C$: & represents logical 'and', # represents negation, `occurs(ingr(s), t1:t2)` represents the event of the system ingression to a state $s$ at the instant in between $t1$ and $t2$. A *State-State rule* `ssr(a => b)` represents `a` being true causes state `b` to be true and a *State-Event rule* `ser(a -> ingr(b))` represents `a` being true causes an ingression to state `b` being true. The extended-BNF definition below gives the syntax of the specification language. Given that symbols "[" and "]" are part of the object language we use "[" and "]" as meta-language symbols to indicate optional elements of a scenario specification. `{ E }*` is used to indicate zero or more instances of a particular element.

```
Spec ::= StatesDef {IndepStatesDef}*
         {HoldingStatesDef}*
         {OccuringEventsDef}*
         {SSRulesDef}* {SERulesDef}*
StatesDef ::= states([ListofStates]).
IndepStatesDef ::= is([#] State).
HoldingStatesDef ::= holdsAt([#] State,T).
OccuringEventsDef ::= occurs(ingr(State),T1:T2).
SSRulesDef ::=
        ssr(ConjOfStates => [#] State).
SERulesDef  ::=
        ssr(ConjOfStates -> ingr([#] State)).
ListofStates ::= State | State, ListofStates
ConjOfStates ::= [#] State1 & ConjOfStates
```

Some basic constraints apply, for example, (a) a specification has to be finite, (b) any `State` mentioned in either `IndepStatesDef, HoldingStatesDef, OccuringEvensDef, SSRulesDef` or `SERulesDef` has to be defined previously in `StatesDef`, (c) `State` not mentioned in `IndepStatesDef` is assumed to be dependent, (d) `T` is a Natural Number, and `T1` is equal to `T2`-1, (e) at least on `ssr` or `ser` is used, and (f) States not included in a `holdsAt([#] State, 0)` definition are assumed initially `false`. Definitions of `State` and `Event` are given by the domain.

*Scenario: automation of decision-making in Smart Homes.* Let's assume a Smart Home is being used to care for an adult with senile dementia. A system which aids these patients by caring for their safety, whilst allowing them to develop their daily activities as much as possible without intervention, sometimes will have the task to decide whether it is worth requesting the attention of the carer or not. Lets assume we have states: *inBed* representing the sensor detecting a person is in bed, *standing/moving/sitting/laying* representing the camera has captured an image which depicts the patient in one of those specific positions, *inbedTime* specifies the time window where the patient normally will go to bed, *mt20u* captures the passage of 20 units of time, *safe/undesirable* represent the belief that the person is in one of those situations, *carerVisits/carerCalls* represents that currently the carer has been required to take one of those actions, *carerGivesOK* is the input given by the carer through the system interface denoting s/he has checked or assisted the patient. We also consider a sequence of events and causal rules:

```
states([inBed, standing, moving, sitting,
  laying, inbedTime, mt20u, safe, undesirable,
  carerVisits,carerCalls,carerGivesOK]).
is(inBed).     is(standing).     is(moving).
is(sitting).   is(laying).       is(inbedTime).
is(mt20u).     is(carerGivesOK).
occurs(ingr(# inBed), 1:2).
occurs(ingr(sitting), 3:4).
occurs(ingr(inBed), 5:6).
occurs(ingr(sitting), 7:8).
occurs(ingr(mt20u), 28:29).
occurs(ingr(carerGivesOK), 31:32).
holdsAt(safe, 0). holdsAt(inbedTime, 0).
ssr(# inBed & standing => safe).
ssr(# inBed & moving => safe).
ssr(# inBed & sitting => # safe).
ssr(# inBed & laying => # safe).
ssr(inBed & # inbedTime => undesirable).
ssr(inBed & sitting & inbedTime => safe).
ssr(inBed & sitting & inbedTime & mt20u
                        => undesirable).
ssr(# safe => carerVisits).
ssr(undesirable => carerCalls).
ser(# safe & carerGivesOK -> ingr(safe)).
ser(undesirable & carerGivesOK ->
                    ingr(# undesirable)).
```

Notice that this scenario is assumed to be happening in the bedroom, which can be detected by the use of movement and/or RFID sensors. If the patient is detected as being located in another area of the house, e.g., the toilet or the living room, then other context specific sets of rules will be applied.

### 3.2 Monitoring and Detection

An algorithm has been provided in [Galton and Augusto, 2002] to explore the dynamics of a scenario specified using $C$ as it unfolds into the future from a pre-specified initial state. Whilst that possibility can be used to predict the causal effects being investigated the forward reasoning algorithm explores exhaustively the values of all state variables time after time. To increase efficiency we use a backward reasoning approach which only investigates that part of the specification which is strictly needed to gather feedback about a particular state. Another important difference is that here we focus on the explanation. Together with the prediction on the value of a state $S$ at time $t$ we offer an explanation of why that is the case. Given that this explanations will be based on causal rules we call them *causal explanations*. This is particularly valuable for carers to understand the context of a warning when an alarm is triggered.

Before any causal inference begins, it is necessary to collect together information about the causal relations amongst the states. These relations are represented by means of a set of trees, in which each node is labelled by one of the states. Each state appears as the label of the root node of one of the trees; within a tree, a node labelled by state $S$ has as its descendants one node for each state on which $S$ immediately depends (by one of the same-time rules, separate rules for the same state are handled in different trees). Thus the independent states always appear as leaves of the trees. We say a tree is *activated* at time $t$ if at least one of its independent states, say $I$, is known to hold at $t$; it is *fully activated* at $t$ if *all* its independent states hold at $t$.

Our backward-reasoning algorithm requires us to find the set of trees supporting a given state $S$. This is achieved in a recursive fashion: for each rule $B_1 \sqcap \cdots \sqcap B_n \leadsto S$ or $B_1 \sqcap \cdots \sqcap B_n \leadsto \bigcirc S$, we apply the process to find the trees supporting each of $B_1, \ldots, B_n$, and each combination of such trees is joined together to form a new tree. The base cases of the recursion are provided by the independent states, whose trees consist of single nodes. We shall denote by $\text{STT}_S$ ($\text{STT}_{-S}$) the set of trees supporting $S$ ($-S$). The general strategy of the algorithm can be informally described as follows. For a query $Holds(S,t)$, starting from $t$ and working backwards we try to find the closest time to $t$ at which a tree supporting either $S$ or $-S$ becomes fully activated by the facts provided in the initial specification of the problem. If the first fully activated tree found supports $S$ then the algorithm answers 'true' to $Holds(S,t)$; otherwise (the first fully activated tree found supports $-S$), it answer 'false'. If no tree supporting either $S$ or $-S$ is found then the algorithm reaches time 0, so the truth value of $S$ at $t$ is determined by its truth value at 0 by persistence (that is, $STT_S \cup STT_{-S} \neq \emptyset$ is always true). The tree or fact used to answer the query is the causal explanation for the happening of $S$ at $t$.

A more detailed explanation is given by the following algorithm. The algorithm can be better understood by following the example given after it. We use $[-]S$ as an abbreviation for '$S$ or $-S$'.

**Goal:** causal explanation for $holds([-]S, time)$
**Input**: a set of non-cyclic Same-Time Causal Rules, *STR*
a set of facts $F$ giving the known states
sets of independent and dependent states: $\mathcal{S}_I$, $\mathcal{S}_D$
a set of known events, $E$, describing state ingressions
**Output**: answer to a query $Holds(S, time)$, with an explanation for its [not] holding at $time$.

IF $holds([-]S, time) \in F$ or
$occurs(ingr([-]S), (time-1) : time) \in E$
THEN answer "*true*" ["*false*"] and give fact as explanation
ELSE
get the sets of trees $STT_S$, $STT_{-S}$
1) Obtain list $AcTimes$ of Activation Times
2) $t \leftarrow nextT(AcTimes)$
3) REPEAT
a) find a tree $T$ in $STT_S$, $STT_{-S}$ with main rule
(a1) $r : B_1 \sqcap \ldots \sqcap B_n \leadsto [-]S$
activated at *t* such that:
*) for each $B_i \in \mathcal{S}_I$: $holds(B_i, t)$
**) for each $B_d \in \mathcal{S}_D$: $holds(B_d, t)$, or
(a2) $r : B_1 \sqcap \ldots \sqcap B_n \leadsto \bigcirc[-]S$
activated at *t* such that:
*) for each $B_i \in \mathcal{S}_I$: $holds(B_i, t-1)$
**) for each $B_d \in \mathcal{S}_D$: $holds(B_d, t-1)$
b) IF $[-]S$ can be proved true from a tree
THEN answer "*true*" ["*false*"] and
use the tree as the explanation
ELSE $t \leftarrow nextT(AcTimes)$
UNTIL ($[-]S$ can be proved true from a tree)

In step (1) "Obtain list $AcTimes$ of Activation Times" means $AcTimes$ is the merged list of activation times for $STT_S$ and $STT_{-S}$. In step (2) "$t \leftarrow nextT(AcTimes)$" means $t$ is made $max(AcTimes)$ and $t$ is extracted from $AcTimes$. If $holds(S,T)$ with $0 \leq T < time$ then $S$ is assumed to hold at $time$ by persistency because should its truth value could have changed in between $T$ and $time$ then a tree to support $holds(\overline{S}, T')$, $T < T' < time$ should have been found at an earlier step.

**Lemma:** The algorithm always finishes.
*Proof:* the set of causal rules are required to be non-cyclic. See [Galton and Augusto, 2002] where they are stratified, which is a stricter notion of being cycle free. Given this hypothesis the algorithm will explore for every time $T$, $0 \leq T \leq time$, in decreasing order, whether $holds(S, time)$ or $holds(-S, time)$ can be proven. In the last case the algorithm will always stop at 0 with a trivial tree as an explanation consisting of the fact $holds([-]S, 0)$, i.e., $holds([-]S, T)$ is true because $holds([-]S, 0)$ and its truth value persists at least until $T$.

### 3.3 Automated Detection and Carer Assistance

In our Java/Prolog-based implementation a predicate `holds(State, GoalTime)` gathers all the meaningful trees to prove or disprove that a `State` is holding at a `GoalTime`. A specific procedure will also gather the alternative explanations for and against the position that `State` is holding at `GoalTime`.

For example, if we want to know whether the state `undesirable` has been reached at say time 30, and if so a description of the context, we can query the system with `holds(undesirable,30)?` and the answer and explanation will be produced:

```
[inBed & sitting & inbedTime & mt20u
                                => undesirable]
```

A fact tells us the context assumes it is `inbedTime` at 0 and then actions are recorded through the sensors that the patient goes `inBed` at 6, `sitting` at 8 and then more than 20 units of time elapses triggering a diagnosis of reaching a potentially `undesirable` situation, i.e., an indication the person may be unwell or fall asleep in an unhealthy position.

If later on we want to know whether the state `undesirable` has been reverted few minutes later then `holds(undesirable,34)?` will be answered negatively explaining that the care giver took care of the situation (an undesirable state was reached at 29 and persisted until 31, after that point the caregiver's intervention allows the situation to be back to normal):

```
[undesirable & carerGivesOK ->
                        ingr(# undesirable)]
```

Complex events can be detected using the language defined by Galton in [Galton, 2005]. Based on Galton's complex-event description language [Augusto and Nugent, 2004] presented an Active Database system with enhanced spatio-temporal reasoning capabilities. There it is explained how different combinations of events related to the spatio-temporal coordinates of a patient and the duration of her/his activities can be monitored and detected. In our system the complex-event detector detects those specific conditions which are important to detect recognize that the patient is in one of the important postures, i.e., *standing*, *laying* or *sitting*. At each time images are processed and the salient features are passed to the complex-event detector. This module will evaluate the current information together with the recent history and if a change in one of the identifiable postures of the patient is recognized then an event is inserted through the predicate `occurs`. Then the causal reasoner will have updated the state ingressions and the same-time and next-time rules can be applied to finish one cycle in the reasoning system.

## 4 Benefits of the System to Society

The problem of nocturnal care and in particular the detection of falls should not be underestimated. For example in one district of the UK, North Down and Ards in Northern Ireland, over 2600 people aged over 80 fall each year. This figure is much higher for people aged over 65 as 1 in 3 will fall per year. Many will eventually suffer a serious injury such as hip fracture which carries a six months mortality rate of 20%. 50% of survivors are unable to return to independent living. Over 95% of hip fractures are falls related. The cost of individual hip fracture averages 29.000 Euros (45% acute care, 50% social care and long term hospitalisation, 5% drugs and follow up) [NHS-EW]. It has been estimated that in the UK falls in the elderly cost 1.4billion Euros annually [Scuffhametal2003]. These figures are replicated in member states throughout Europe.

We have described a prototype which implements part of the logical core of the decision making system to be used for a Smart Homes system. Figure 1 summarizes the essential components in the monitoring and decision-making process. This prototype shows that it is possible to increase the quality of care provided during the night for dementia patients living at home by alerting carers of unsafe conditions during the night period. It also provides evidence that it is possible to build a solution which (a) relies on vision technology but not necessarily in an intrusive way as previously believed (b) helps to prevent or detect falls without the patient having to wear or interact directly with any technology (c) increases the cost effectiveness and quality of healthcare provision.

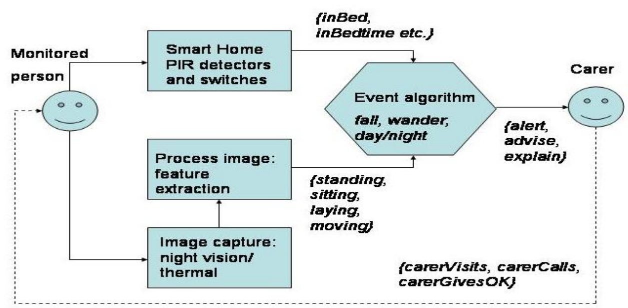



Figure 1: Monitoring and decision-making.

## 5 Comparison with Similar Approaches

Given the importance of addressing ways to provide home care for elderly people, researchers have started to explore technological solutions to enhance healthcare provision in a way which complements existing services.

*Autominder* [Pollack, 2005] uses a scheduler and temporal constraints to guide elderly people through their daily tasks. One difference with our proposal beyond the specific hazard we consider is that they assume direct interaction with the technology on behalf of the patient or, most likely, the carer. We provide a solution which does not rely on elderly people having to remember how to use technology and also reduces carer intervention to the essential, which is particularly important to enhance their quality of life during the night. Wearable sensors have been also used, see for example [Patterson *et al.*, 2005], to provide context-aware services to people on Smart Homes. In this work we explicitly want to depart from the assumption that the patient has to pro-actively interact with devices in order to obtain a benefit. Although we recognize wearable devices are beneficial in some cases and for some patients, we are focusing our efforts on exploring services that can be provided without depending of the assumption that a patient is forced to wear a sensor or use a device (e.g., a PDA). We think that for the case of elderly patients with dementia this assumption is unrealistic and does not take into account that patients may forgot how to use or even that they are supposed to use them.

Other approaches either use full image processing [Natale *et al.*, 2004] without caring for patients need for intimacy or go to the other extreme refusing to use any image processing at all and therefore loosing their benefits. Here we support the thesis that image processing can be used in a way that does not minimize patient's privacy and at the same time a simple outcome can be connected to an inferential unit capable to enhance the service with advantages for patients, carer and the National Health System.

## 6 Conclusions

We provide a description of the real-world problem of providing automated nocturnal assistance to elderly patients and their carers. The computational processes underlying a prototype that has been used to find a solution to the problem are explained. The system provided is based on spatio-temporal reasoning which is then used in connection with a causal reasoner to keep a conceptualization of how activities are developed by the patient and to detect when s/he may need assistance. The backward reasoning algorithm can be used in real-time to detect a problem and trigger an alarm with an associated explanation or to conduct retrospective analysis on the reasons for a previous alarm.

The practical implications of our work is that safety can be assisted by intelligent technology without imposing the burden of wearing or interacting with technology in order to obtain that assistance. This not only benefit people requiring support but also improves the quality of life of their carers and impacts in the overall efficiency of the health system. All these features will gain increasing value as our society ages stretching the current scarce resources.

## References


[Augusto and Nugent, 2004] J. C. Augusto and C. D. Nugent. The use of temporal reasoning and management of complex events in smart homes. In R. López de Mántaras and L. Saitta, editors, *Proceedings of European Conference on Artificial Intelligence*, pages 778–782. IOS Press, 2004. August, 22-27.

[Augusto, 2005] J. C. Augusto. Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33(1):1–24, 2005.

[Bailly *et al.*, 2005] G. Bailly, J. L. Crowley, and G. Privat, editors. *Proceedings of European Conference on Smart Objects and Ambient Intelligence (sOc-EUSAI'05)*, Grenoble, France, October 2005. ACM Press, NY, USA.

[Desurmont *et al.*, 2005] X. Desurmont, B. Lienard, J. Meessen, and J-F. Delaigle. Real-time optimizations for integrated smart network camera. In *Real-Time Imaging IX*, volume 5671 of *Proceedings of SPIE*, pages 85–92, San Jose, CA, USA, February 2005. N. Kehtarnavaz and P. A. Laplante.

[DofH-UK, 2004] DofH-UK. Improving chronic disease management. `www.dh.gov.uk`, March 2004.

[Galton and Augusto, 2002] A. Galton and JC. Augusto. Two approaches to event definition. In A. Hameurlain, R. Cicchetti, and R. Traunmller, editors, *Proceedings of 13th DEXA*, pages 547–556. LNCS Vol. 2453 - Springer Verlag, 2002.

[Galton, 2005] A. Galton. Eventualities. In M. Fisher, D. Gabbay, and L. Vila, editors, *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 25–58. Elsevier, 2005.

[Grace *et al.*, 2000] J. B. Grace, M. P. Walker, and I. G. McKeith. A comparison of sleep profiles with dementia with lewy body and Alzheimer's disease. *International Journal of Geriatric Psychiatry*, 15(11):1028–1033, 2000.

[Heaney, 2006] Marie Heaney. Developing a telecare strategy in Northern Ireland. In *Procedings of the Workshop on The role of e-Health and Asssistive Technologies in Healthcare*. Ulster Hospital, UK, March 2006.

[IST Advisory Group, 2003] IST Advisory Group. Ambient intelligence: from vision to reality, for participation in society & business, September 2003. URL: `ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf`.

[Natale *et al.*, 2004] F. De Natale, O. Mayora-Ibarra, and L. Prisciandaro. Interactive home assistant for supporting elderly citizens in emergency situations. In *Proceedings of Workshop on "AmI Technologies for WellBeing at Home"*. Johan Plomp and Paolo Tealdi (eds), November 8-10 2004. EUSAI2004, Eindhoven, The Netherlands.

[Nugent and Augusto, 2006] C. D. Nugent and J. C. Augusto, editors. *Proceedings of the 4th International Conference on Smart Homes and Health Telematic (ICOST2006)*, volume 19 of *Assistive Technology Research*, Belfast, UK, June 2006. IOS Press.

[Patterson *et al.*, 2005] D. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the IEEE Int. Symposium on Wearable Computers*, pages 44–51, Osaka, Japan, October 2005. IEEE Press.

[Pollack, 2005] M. E. Pollack. Intelligent Technology for an Aging Population: The Use of AI to Assist Elders with Cognitive Impairment. *AI Magazine*, 26(2):9–24, 2005.

[RSPA, 2005] RSPA. Northern Ireland home prevention strategy 2004-2009, August 2005. URL: `www.rospa.com/ni/info/hap\_strat\_04.pdf`.

[Shaw *et al.*, 2003] F. Shaw, J. Bond, D. Richardson, P. Dawsona, N. Steen, I. McKeith, and R. Kenny. Multifactorial intervention after a fall in older people with cognitive impairment and dementia presenting to the accident and emergency department: randomised controlled trial. *British Medical Journal*, pages 326–73, January 2003.

[TheHIP Group, 2005] TheHIP Group. The impact of e-health and assistive technologies on healthcare. `www.tehip.org.uk`, 2005.

[WHO, 2004] WHO. International plan of action on ageing: report on implementation, November 2004. URL: `www.who.int/gb/ebwha/pdf\_files/EB115/B115\_29-en.pdf`.

[WHO, 2006] WHO. The world is fast ageing - have we noticed?, April 2006. URL: `www.who.int/ageing/en/`.

# The detection and tracking of the body movements with blackboard architecture

**Christine Leignel[1], Christophe Dousson[2]**
[1]GET – ENST Bretagne, Département Image et Traitement de l'Information,
CS 83818, 29238 Brest France.
Email : christine.leignel@enst-bretagne.fr
[2]France Telecom R&D, 2 Avenue Pierre Marzin, 22304 Lannion Cedex.
Email : christophe.dousson@orange-ftgroup.com

## Abstract

In the context of Human Machine Interfaces (HMI), the visual recognition of the body and gestures allows natural and intuitive remotely controlled HMI, as well as the animation of an avatar. This paper describes an artificial intelligence system for detecting and tracking a person in real time in a natural scene, acquired using a single color camera. The scene analysis system uses a blackboard intelligence system made up of a hierarchical structure, with a strategy at the top which supervises tasks which in turn activate specialists using an opportunistic approach. The detection process is based on the search for the various parts of the body (head, arms, forearms, hands) in an image thanks to specific processing for each of them. Each of the levels possesses its own blackboard in order to organize the data into a hierarchy and to gradually increase the system's intelligence. The upper part of the body is detected but this work can be extended to whole body detection.

## 1 Introduction

In the context of developing human machine interfaces (videoconferencing, collaborative virtual environments, multimodal applications), it is useful to reproduce the user's gesture via a clone or avatar animation. The animation of a synthetic clone requires movement analysis in order to restore gestures. The body detection of a person in a scene, with a single camera, without constraints (no equipment on the person), is a challenging problem because of the variety of possible postures, the problem of one limb being hidden by another limb or by an object in the scene, and because of the lighting conditions. We attempt to detect and track a person by his motion and his body [Tamagawa et al., 2001; Chang and Huang, 2000]. The objective of the system presented is to describe the whole body. In this paper, we limit ourselves to the detection of the upper body of a person in an image via a blackboard artificial intelligence architecture, but the lower body can be detected in the same manner. The person is in front of the camera, they can stand up or sit down. The background is of no importance because it is subtracted in the first acquired image when the person comes into the field of the camera.

The model adopted is composed of sticks for the arms and forearms, squares for the hands, a rectangle for the chest, and an ellipse for the head. In order to obtain a complete body closest to reality, that is, so that the model corresponds as closely as possible to the person in the scene, our system is based on an architecture enabling the cooperation of various sources of knowledge from image processing to *a priori* information about relationships between the different parts of the human body.

The different limbs of the body (arms, forearms, hands, chest, etc.) are searched for in the image by using specific processing for each of them at the pixel level, our lowest source of knowledge. Sources of knowledge exchange information through blackboard architecture. They cooperate at the top level in the strategy by opportunistic supervision, necessary to arbitrate potential conflicts, and to trigger a given task (upper limb search task, hand search task, etc.) according to the results available in the blackboard. Each task activates one or more specialists.

Body part aggregations are performed at higher levels of knowledge, thanks to a hierarchical organization of the knowledge sources and blackboards (one blackboard for each level), like the ATOME blackboard [Laasri and Maître, 1989]. After the body is detected, the system can be used for tracking people in real time in an image sequence obtained with a single camera.

Section 2 presents related work. Section 3 describes the hierarchical artificial intelligence system. The hierarchical structure used is explained in section 4. In section 5 the rules for the creation and elimination of elements are presented, in section 6 scores of the upper limbs are explained, and in section 7 some results are shown. We conclude in section 8.

## 2 Related work

Statistical techniques, that allow the position of hands and head to be found, already exist [Wren et al. 1997; Bernier and Collobert, 2001]. But they do not give the arm and forearm position needed for the synthesis of an animated clone or avatar, and in this case, the animation must rely on inverse kinematics techniques which can be imprecise. [Agarwal and Triggs, 2006] discuss a bottom-up approach that uses local image features to estimate human upper body poses from single images in cluttered backgrounds and learn a set of bases that correspond to local features on the human body. This works in the presence of a natural background, without prior segmentation. But the depth ambiguities have not always been resolved.

On the other hand, blackboard systems already exist for speech recognition and computer vision. In voice recognition, the HEARSAY II blackboard [Lesser and Erman, 1988] assembles syllables from phonemes and thus proceeds up to the sentence. The VISIONS computer vision system [Hanson and Riseman, 1978] analyzes a scene thanks to the interpretation of the joint knowledge sources leading to the grouping of segments into objects and objects into scenes. In "Sahara", an acronym for Semi-automatic help for region analysis, the interpretation of airports is considered and objects of interest like runways, taxiways, buildings and shelters are searched for [Druyts et al., 1997]. Our system differs in its application and in the hierarchical structure of its blackboards, allowing them some intelligence, as will be seen below.

# 3 The hierarchical artificial intelligence system

## 3.1 The global architecture knowledge sources

The system is composed of three levels of sources of knowledge:

- A low level composed of specialists,
- An intermediate level, made up of tasks,
- A high level consisting of strategy.

The specialists are **knowledge modules**: they do not have knowledge about the object detected. The tasks and the strategy on the other hand are **decision modules**. The tasks assemble results from the specialists, and the strategy decides to activate a particular task.

The **specialists**, presented below in paragraph 4.1, operate on the pixels and suggest possible solutions for the localization of an element, hand, forearm, and arm. Each one constitutes a hypothesis of a distinct nature.

The **tasks** process results given by the specialists to formulate new hypotheses. For example, the upper limb (UL) task finds a hypothesis of the existence of a UL, which contains exactly one shoulder, one arm, one forearm and one hand.

The **strategy** groups the results of the different tasks to build the whole body (cf. figure 1), and the rules (cf. § 5) make it possible to keep only two upper limbs per person.

Specialists are not independent and communicate with each other via the blackboard of lowest level BB3 (cf. figure 2), which contains the results of the different specialists updated at each processing. Each level has its own blackboard, and arrows give the direction of flow of data traffic, and show exchanges with blackboards (cf. figure 2). Each blackboard has its own intelligence according to its level in the structure. For example, the blackboard of the UL task contains the upper limbs.

The strategy activates different tasks, which in turn activate one or more specialists, according to the partial results contained at the same time in the blackboard of its level. An opportunistic supervision structure arbitrates the potential conflicts between different tasks or specialists which otherwise can be activated simultaneously.

## 3.2 The global architecture of the blackboards

The low level blackboard contains hypotheses for hands, forearms, arms and shoulders. The task blackboard gives a semantic interpretation of the hypothesis produced by the specialists (cf. figure 1). For example, the hand is associated with the corresponding forearm, the forearm to the arm, etc. The strategy blackboard at the high level groups together the hypotheses in order to construct the upper body part (two ULs, one head and a chest). The three blackboards, organized into a hierarchy, communicate with each other via the tasks and the strategy (cf. figure 2).

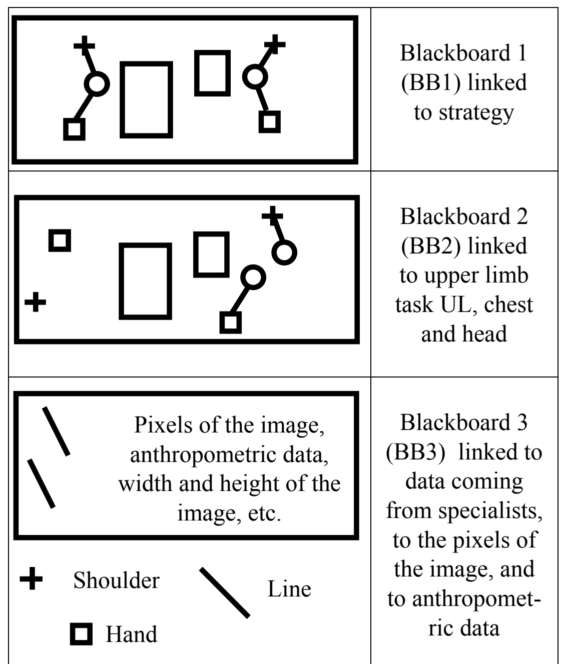



Figure 1. Hierarchical structure of the blackboard.

# 4 The hierarchical structure

We now describe the specialists, the tasks, and the strategy.

## 4.1 The specialists

Two kinds of specialists are considered, those of type 1 (foreground mask, skin-colored mask, intersection mask, contour detection) and those of type 2 (Hough transform). The results of the first are used by the second.

**Foreground and skin-colored masks**
After recording an initial image, the foreground mask is the subtraction between the background and the current image; the skin-colored mask shows skin-colored pixels in the YUV color space obtained thanks to a skin color table; the intersection mask contains the pixels that belong to both the foreground and skin-colored mask. These masks are used by the various specialists and are stored in the low level blackboard.

**Head detection specialist**
Under the hypothesis that the head is elliptic, it is tracked and located by an elliptical Hough transform [Leavers, 1993], and is fuzzy [Leignel et al., 2003], which is an innovative approach.
The major restriction for the head detection with an elliptical Hough transform is when the person is wearing glasses.

**Arm detection specialist**
Under the hypothesis that arms and forearms are modeled by line segments, the SHEN CASTAN detector [Shen and Castan, 1986] finds the contours in the image. By the line Hough transform, straight lines can be detected [Duda and Hart, 1972] in the principal directions image (contours obtained by SHEN CASTAN are summed by zones, given the image of strong gradients), and forms the hypothesis of arms and forearms.

**Detection of the hands**
Assuming that the histogram on the face to be similar to that of the back of the hands, and that the person does not have bare arms, we use the face histogram to detect hands. From the histogram of the face $h(u(i),v(i))$ on the initial frame, and only on the skin-colored pixels i, the image of probability is:

$$P(u,v)=\frac{h(u(i),v(i))}{\sum_{(u,v)} h(u(i),v(i))} \tag{1}$$

This probability for each pixel makes it possible to assign a value of confidence to each hand hypothesis defined in a rectangle around the extremity of the arm hypothesis in a shoulder-to-hand approach (cf. § 4.3).

**Detection of the shoulders**
Shoulders are located from the head via the anthropometric data [Winter, 1979], when the person is in front of the camera.

## 4.2 The tasks

The tasks use and structure the results generated by the specialists, stored in the low level blackboard. Three tasks are defined:

- The head task whose purpose is to locate the head,
- The chest task whose aim is to find the chest,
- The upper limb task whose objective is to make only one upper limb.

When limited to only one person, the task of detecting the chest uses a priori knowledge, called the long term memory in VISIONS, in opposition to the short term memory of the specialists. The chest is between two parallel vertical lines from the shoulders and under the head. This chest knowledge is important in the case where a UL is hidden. For example, if we know the position of the chest, we can make the hypothesis that the arm is behind the chest when hidden. By establishing the correspondence between the four elements (hand, forearm, arm, shoulder) constituting the UL, the UL task verifies the existence of a UL at each activation of a specialist. Two zones (shoulder and elbow) are associated with the two joints of the four distinct elements which make up the UL.

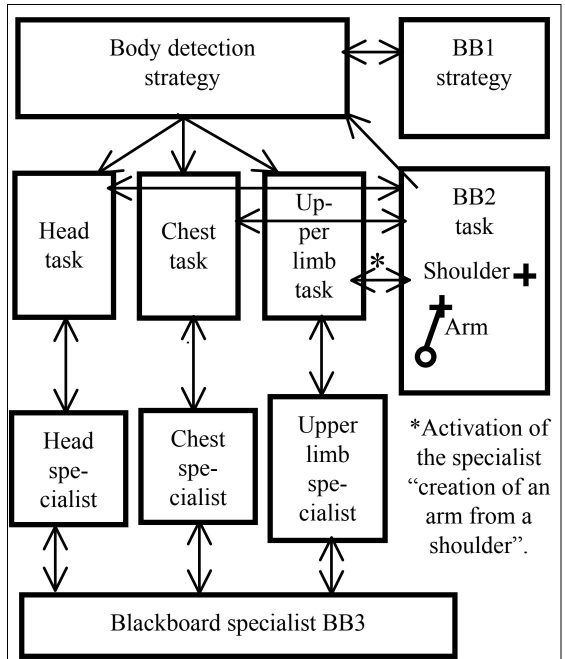



Figure 2. Hierarchical structure of the intelligent system: strategy, tasks and specialists and their relationship with blackboards (BB).

## 4.3 The opportunistic approach to the upper limb task

Three approaches are feasible: the shoulder-to-hand (arterial) approach, the hand-to-shoulder (venous) approach and the opportunistic approach.

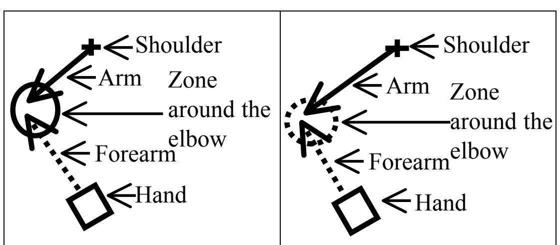
In a procedural shoulder-to-hand approach, from a zone around the shoulder, we examine the hypothesis of straight lines corresponding to arms by a Hough transform. The length of the arm is determined by the anthropometrical data and then, the extremity of the arm given by its length corresponds to the elbow. In a similar way, in a zone around the elbow, the different forearm hypotheses stem from the previous arm hypothesis by a Hough transform. The length of the forearm segment makes it possible to find its extremity.

The UL task thus allows the construction of different hypotheses of ULs (various elbows and associated with each, different forearms) to reach their extremities. Each of them is checked thanks to the hand-checking module by measurement, in a zone around the extremity of the arm, of a factor of confidence related to the histogram of skin color obtained on the face.

The procedural hand-to-shoulder approach goes from the hypotheses of the hands to the shoulders. The histogram obtained on the face guides the search for potential hands hypotheses. Each hypothesis defines a zone inside which the various hypotheses of forearms are obtained by a line Hough transform on the image of gradients, and with the same technique we go from forearm to arm with various hypotheses for arms and thus for shoulders. It is then necessary to check the consistency of the shoulder position in relation to the head position. The problem of the shoulder-to-hand and hand-to-shoulder approaches is that there is no universal approach; it is impossible to know beforehand which approach to use.

In the opportunistic approach, all the hypotheses are recursively investigated (shoulder, arm, forearm, hand), including both the shoulder-to-hand and the hand-to-shoulder approaches. Each task activates specialists in an opportunistic way, i.e. according to the current results in the low level blackboard, until no more rules are applied. Let us consider a situation to illustrate the opportunistic approach. From the shoulders (we first consider the shoulders because the head is located with certainty), suppose that strong hypotheses for arm detection are found on which hypotheses for forearms can be based. Zones at the end of the arms characterize elbows. Let us suppose that all forearm hypotheses have a too low score (cf. § 6). The decision module strategy chooses in this case to discard these forearm hypotheses. Then from the hypotheses of hands (assuming that they are more likely than the discarded forearms), the decision module strategy generates forearms hypotheses. It is then necessary to check if the extremity of the forearms falls into the zone around the elbow previously defined from the arm and thus if the upper limb is complete (cf. figure 3(a)).

In the same way, the opportunistic approach can be described in the complementary manner (beginning from the hand and then switching to the shoulder). In this case the elbow is the zone defined at the end of the forearm (cf. figure 3(b)).



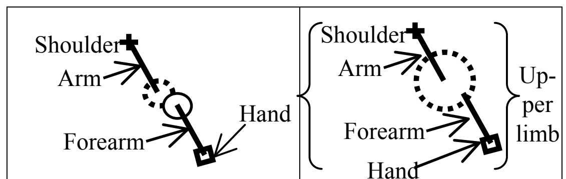

a. arm defined elbow | b. forearm defined elbow

Figure 3: Opportunistic approach.

### 4.4 The strategy

The strategy elaborates one hypothesis of the upper part of the body assuming two upper limbs linked by two shoulders fixed starting from the head, and a chest.
There are two possible strategies for our system:

- By an opportunistic approach: all the specialist creation and elimination rules are activated as long as the condition of their activation is checked. The number of upper limbs right and left are counted and a score is given for each of them in order to select the best one on each side (right/left);
- By a decision module located in the strategy: once head, shoulders and hands have been located, useful specialists are selected alternately in a procedural way, by one of the two following forms, until there is at least one UL on each side (left/right) :
  - Generation of new elements, where elimination rules of elements are applied, UL searched for, and rules for elimination of ULs are activated,
  - Loosening constraints about sizes of elbow and shoulder zone (cf. figure 4) leads perhaps to new ULs, ULs are searched for again, and rules for elimination of ULs are activated.




a. Before constraint | b. After constraint

Figure 4. Completing an upper limb after loosening elbow zone.

## 5 The rules

We need rules for creation and elimination of elements and U.Ls to keep only one hypothesis per U.L per side.

### 5.1 The rules of elements creation

The rules are in the form condition->action. An action is made up of specialist activation. The condition searches in the low level blackboard for the element necessary to activate an action. If the condition is not verified, the action is not activated. Rules and associated conditions are as follows (cf. tables 1 and 2):

| Element | Shoulder | Elbow |
|---|---|---|
| Condition ↓ | Search for a shoulder | Search for a forearm |
| Action | Creation of arms from a shoulder | Creation of arms |

Table 1: Condition/Action for shoulder and elbow.

| Element | Venous hand<br>hand → shoulder | Arterial hand<br>shoulder → hand |
|---|---|---|
| Condition | Search for a hand | Search for a forearm |
| ↓<br>Action | Creation of fore-arms from the hand | Verification of the availability of a hand |

Table 2: Condition/Action for the venous and arterial hand.

## 5.2 The rules of elements and upper limb elimination

Among the element rules, we can differentiate logical and anthropometric rules. Some of them are described below:

- If arm 1 comes from a forearm, itself is the result of arm 2, arm 1 is discarded (logical rule),
- If a forearm or an arm is in front of the face, the forearm or the arm is discarded (logical rule),
- Elimination of an arm if it is close and parallel to another arm (logical rule), seen in figures 5 and 6,
- Elimination of an arm in front of the chest (anthropometric rule), and so on.

Some rules for the elimination of a upper limb are:

- Elimination of an upper limb whose angle between arm and forearm is within 0°-35°, but preservation of the constitutive elements,
- Elimination of a hand which is on a forearm, in the case of a single upper limb, and so on.

# 6 The scores

We need to give a score the upper limbs found, in order to keep only the best score per side.

$$ScoreC.S.M = ScoreSh + ScoreArm + ScoreFore + ScoreH + ScoreElbow$$

where:

- $ScoreSh$ the shoulder score, $ScoreSh = function(D_{AS})$ ($D_{AS}$ is the distance between the arm and the shoulder),
- $ScoreArm / ScoreFore$ the arm/forearm score given by the normalized sum of strong gradients on the arm/forearm segment,
- $ScoreH$ the hand score provided by the correlation between hand and face,
- $ScoreElbow$ the elbow score, $ScoreElbow = function(D_{AF})$ ($D_{AF}$ is the distance between the arm and the forearm).

# 7 The results

We show the interest of an opportunistic approach on two images, in the case of four hypotheses of arms and forearms. White crosses symbolize shoulders. Segments on the image of strong gradients are hypotheses of arms and forearms. Squares represent hands. Upper limbs are in white on the person, and in black in the strong gradients image. On the right, each figure (10, 12, 14, and 16) shows results for the elements and the ULs in the strong gradients image.

In figure 9, the shoulder-to-hand approach in image 2 brings to light a right UL, and a left UL. But this approach does not lead in image 3 to a right and left UL, so another approach is necessary.

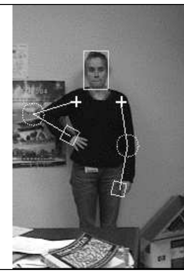


Figure 9 Image 2. Upper limbs.

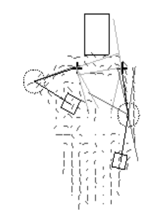


Figure 10 Image 2. Elements and ULs on the image of strong gradients.

Procedural arterial (shoulder-to-hand) approach in image 2.

In figure 11, the opportunistic approach in image 2 reveals a left UL, and two right Uls, only one of which corresponds to reality. In figure 13, the hand-to-shoulder approach in image 3 leads to the generation of a left UL, and a right UL, but the shoulder-to-hand approach does not give the same results. In figure 15, the opportunistic approach in image 3 shows a left UL, and four right Uls, only one of which corresponds to reality.

The interest of the opportunistic approach lies in the fact that it is not possible to know beforehand which of the two approaches (shoulder-to-hand or hand-to-shoulder) leads to the creation of ULs. The opportunistic approach makes it possible to find upper limbs, including the case where one of the two procedural approaches forbids it.

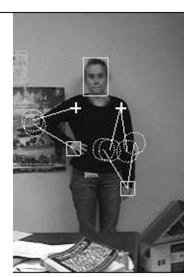


Figure 11 Image 2. Upper limbs.

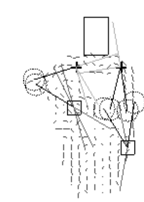


Figure 12 Image 2. Elements and ULs.

Opportunistic approach in image 2.

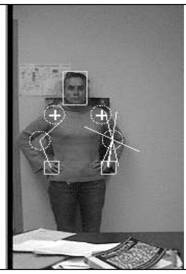


Figure 13 Image 3. Upper limbs.

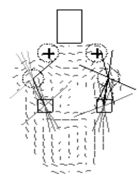


Figure 14 Image 3. Elements and ULs.

Procedural venous (hand-to-shoulder) approach in image 3.

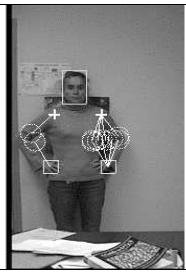


Figure 15 Image 3. Upper limbs.

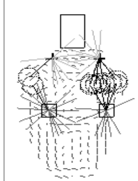


Figure 16 Image 3. Elements and ULs.

Opportunistic approach in image 3.

## 8 Conclusion

This system for detecting a person in the field of a camera is an innovative approach by its blackboard type artificial intelligence aspect. Blackboards allow the collaboration of various sources of knowledge in an opportunistic or in a supervision way. Low level blackboard hypotheses are explored recursively to detect forearms, arms, hands, and chest. Each level has its own blackboard with its hierarchical organization intelligence. The aim of the UL task is the creation of an upper limb. The chest task uses *a priori* knowledge. The strategy groups together two upper limbs, the chest and the head to constitute the upper body. It is possible to extend the system to the whole body, considering the lower body elements by similarity. Moreover, the hierarchical organization of the blackboard structure allows the tracking of persons to become generic to more than one person.

Cases of hidden limbs could be dealt with by tracking, to remove ambiguity on the current frame by taking into account the past frames in the case of numerous possible hypotheses or of various incomplete hypotheses.

## References


[Agarwal and Triggs, 2006] Ankur Agarwal and Bill Triggs. A Local Basis Representation for Estimating Human Pose from Cluttered Images. In *Proceedings of the 7th Asian Conference on Computer Vision*, Hyderabad, 2006. Springer-Verlag. Lecture Notes in *Computer Science*, I:50-59, 2006.

[Bernier and Collobert, 2001] Olivier Bernier and Daniel Collobert. Head and Hands 3D Tracking in Real Time by the EM Algorithm. In *IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, 2001.

[Chang and Huang, 2000] I.-C Chang and C.-L Huang. The model-based human body motion analysis system, *Image and Vision Computing*, 18(14):1067-1083, November 2000.

[Druyts et al., 1997] Pascal Druyts, Wim Mees, Dirk Borghys, Christiaan Perneel, Marc Acheroy, and J.L Valero. Sahara: Semi-automatic help for region analysis. In *Proceedings of the Joint workshop of ISPRS Working Groups I/1, I/3 and IV/4: Sensors and Mapping from Space*, 17:267-274, Hannover, GE, September 1997.

[Duda and Hart, 1972] Richard O. Duda and Peter E. Hart. Using the Hough transforms to detect lines and curves in images. *Comm. Of the ACM*, 15(1):11-15, 1972.

[Hanson and Riseman, 1978] Allen R. Hanson and Edward M. Riseman. VISIONS: A Computer System for interpreting Scenes. *Computer Vision Systems*, 1978.

[Laasri and Maître, 1989] Henri Laasri and Henri Maître. Flexibility and efficiency in Blackboard Systems: Studies and Achievements in ATOME. *Blackboard Architectures and Applications*, pages 309-322, 1989.

[Leavers, 1993] V.F. Leavers. Survey, which Hough transform? *Computer Vision, Graphics, and Image Processing*, 1993.

[Leignel et al., 2003] Christine Leignel, Olivier Bernier, Daniel Collobert, and Renaud Seguier. Transformée de Hough elliptique floue rapide. In *CORESA'2003*, pages 189-192, Lyon, France, 2003.

[Lesser and Erman, 1988] Victor R. Lesser and Lee D. Erman. A Retrospective View of the HEARSAY II Architecture. *Blackboard Systems*, pages 87-111, 1988.

[Shen and Castan, 1986] J. Shen and S. Castan. An optimal linear operator for edge detection. In *IVC*, 1986.

[Tamagawa et al., 2001] Ken Tamagawa, Toshio Yamada, Tetsuro Ogi, and Michitaka Hirose. Developing a 2.5-D Video Avatar. In *IEEE Signal Processing Magazine*, 18(3):35-42, May 2001.

[Winter, 1979] David A. Winter. *Control of Human Movement*. Biomechanics and Motor, John Wiley and Sons, 1979.

[Wren et al., 1997] Christopher R. Wren, Ali Azarbayajani, Trevor Darell, and Alex P. Pentland. PFINDER : Real-Time Tracking of the Human Body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780-785, 1997.

# Dynamic Macroprogramming of Wireless Sensor Networks with Mobile Agents

**Reza Razavi**
University of Luxembourg
FSTC, LUXEMBOURG
razavi@acm.org

**Kirill Mechitov, Gul Agha**
University of Illinois at Urbana-Champaign, IL, USA.
{mechitov, agha}@cs.uiuc.edu

**Jean-François Perrot**
Université Pierre et Marie Curie
LIP6, Paris, FRANCE
jean-francois.perrot@lip6.fr

## Abstract

Wireless Sensor Networks (WSNs) are a key enabling technology for Ambient Intelligence. Macroprogramming has been proposed as a technique for facilitating programming WSNs, but current solutions do not provide the combination of dynamicity and query specification that would be useful to domain experts. We have implemented the first query engine which provides both these features. Our system leverages AI methods such as multiagent systems and sophisticated meta-level knowledge representation techniques to keep track of the domain knowledge and to enable adaptation in query processing. Such adaptations include dynamic representations, transformations, optimizations and deployment strategies translating queries into a system of automatically generated mobile actors in a WSN.

**Keywords**: Sensor Networks, Macroprogramming, Mobile Agents, Actor Systems.

## 1 Motivation and Problem Statement

The infrastructure for *Ambient Intelligence* (AmI) [IST, 2003] will include a massive deployment in everyday environments of *Wireless Sensor Networks* (WSNs) consisting of autonomous, spatially distributed, tiny, low-powered computers, endowed with communication, sensing and actuating capabilities [Whitehouse *et al.*, 2006; Luo *et al.*, 2006]. To realize this vision, WSNs must provide an omnipresent interactive environment. Our goal is to support the exploitation of an AmI infrastructure by ordinary end-users, not embedded systems programmers.

*Macroprogramming* has been proposed as a technique for facilitating programming WSNs. Macroprogramming enables the specification of a given distributed computation as a single global specification that abstracts away low-level distribution. The programming environment first automatically compiles this high-level specification into the relatively complex low-level operations that are implemented by each sensor node, and then deploys and executes these operations [Mainland *et al.*, 2004].

As explained in [Razavi *et al.*, 2006a], we are interested in situations where both the users' requirements and the WSN environment may be dynamic. This goal matches the diversity of functionalities that end-users need from the ambient infrastructure, further amplified by the unpredictability of the phenomena being monitored and the potential changes in the ambient computing infrastructure. We develop a high-level language which supports WSN macroprograms called *uQueries.* A uQuery is represented and executed using *uQuery Engines.* Thus the key challenge is to support the cost-effective and stepwise development of *uQuery Engines*. In particular, this requires enabling uQuery specifications by multiple concurrent and uncoordinated end-users. It also requires deploying and executing such specifications in a parallel and resource-efficient manner which ensures interoperability with other uQuery engines.

In this paper, we focus primarily on representation (Section 3) and execution of uQueries (Section 4). uQueries result in a system of dynamically created meta-actors which generate actors that are concurrently deployed on a WSN. The meta-actors coordinate to control the execution of these actors on a mobile agent system. Due to limited space, we do not address our implementation of the end-user Web interface for uQueries.

As a motivating example, consider a scenario proposed by Microsoft Research [Woo *et al.*, 2006]: a parking garage has been wired with break beam sensors and security cameras. Two ordinary end-users, namely Liz and Pablo, working independently, desire to use the ambient system for their own goals. Liz is a site manager of the garage building and is interested in collecting vehicle arrival time data. Pablo is a security officer in the building who wants to issue tickets to speeding drivers. In the remainder of this paper, we will use this example to illustrate our system.

## 2 Meta-level uQuery Engine Architecture

A key challenge is to dynamically transform high-level specifications of the users' queries into low-level executable code for WSNs. We need a truly dynamic deployment of independent code segments, which may interact and migrate, running on distributed sensor nodes. This is facilitated by a meta-level architecture for uQuery Engine.

### 2.1 Knowledge Level Support

The *knowledge level* keeps track of the domain knowledge and controls query processing (representing, transforming, optimizing and deploying). Specifically, it tracks two types of metadata: *Static metadata* consisting of uQuery representations and the business ontology, and *Dynamic metadata* consisting of the contextual information (including computed domain objects) obtained from uQuery execution.

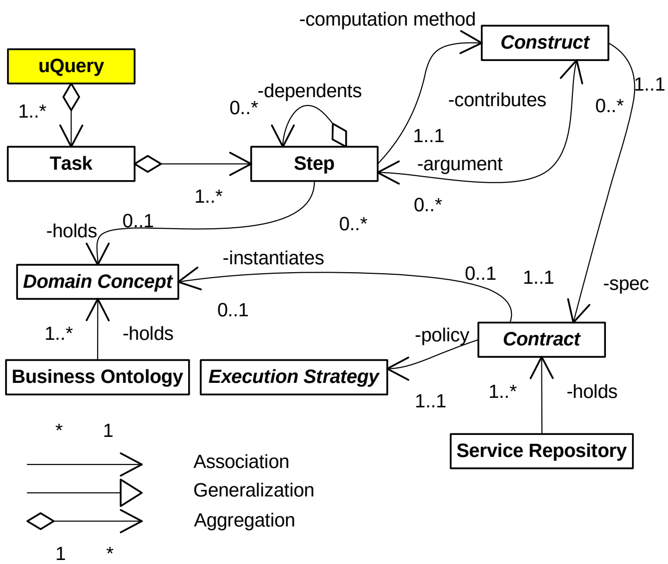



**Figure 1: uQuery representation meta-model**

In [Razavi *et al.*, 2006b] we describe *Dart*, the first representation meta-model that satisfies requirements of dynamicity, end-users accessibility, as well as extendibility and reusability by programmers. Figure 1 illustrates Dart using the UML notation. A *uQuery* is represented as an aggregation of *tasks*, where each task is a set of interrelated steps. When a *step* is executed, it produces a domain object. A *construct* specifies the computation method for that object. A step bridges a *construct* with the concept of that object. The most common type of construct is the *primitive construct*, which reifies a function call. Each construct refers to (or *instantiates*) a *contract* which holds meta-data about the construct. In the case of *primitive constructs*, a contract incorporates the type specification, arguments, name and a list of properties required for its execution. The *service repository* for a uQuery Engine holds these contracts. The *business ontology* holds the domain concepts, together with their relationships and constraints. In order to execute a construct, we need to make platform-specific decisions, which are delegated to the *execution strategy*.

We assume that both the business ontology and the service repository are given. We further assume a comprehensive service repository which enables all interesting uQueries to be explicitly represented as a composition of its elements.

### 2.2 Operational Level Support

At the operational level, we require that fine-grained mobile agent applications can be executed on resource-limited, real-time distributed systems. The mobile agent platform is responsible for:

- Deploying and executing actor code dynamically generated at runtime,
- Dynamically discovering and providing access to all sensors in the WSN, and
- Implementing the elements of the service repository.

```
1. ((lambda (migrate)   ; main function
2.   (seq
3.     (migrate         ; migrate to destination
4.       200            ; destination id
5.       111)           ; meta-actor id
6.     (par (extmsg     ; send result back to source
7.       111            ; meta-actor id
8.       (migrate
9.         100          ; source id
10.        (prim        ; execute primitive
11.          1          ; primitive index in library
12.          nil))      ; list of arguments (empty)
13.    ))))
14.  (lambda (adrs val)   ; migrate function
15.    (callcc (lambda (cc) (send (list adrs cc
16.      (list quote val)))))))
```

**Figure 2: Filled mobile agent code template.**

To the best of our knowledge, the only existing system which satisfies our operational level requirements is *ActorNet* [Kwon *et al.*, 2006]. ActorNet agents are based on the actor model of computation [Agha, 1986]. The ActorNet runtime consists of an interpreter running on each sensor node in the WSN along with several supporting services. The runtime enables the actors to execute, communicate, migrate and access sensors and actuators.

The uQuery execution procedure fills a template for an ActorNet agent (see Figure 2). The template contains four areas to be filled by the code generator:

- Meta-actor id on lines 5 and 7,
- Application logic on lines 10-12,
- List of arguments on line 12, and
- Execution location and uQuery Engine server address on lines 4 and 9, respectively.

In Section 4.2, we explain how this template is used by the knowledge level when executing uQueries.

## 3 uQuery Representation

We now describe the method for representing uQueries in our system. The execution procedure will be explained in the Section 4.

Consider the parking garage example from Section 1. Pablo wants to make a uQuery to take a photo from a specific camera in a sensor network. We assume a business ontology with one concept, *Photo*, and a service repository with one contract, *Capture Camera*, which returns a Photo object, and a library including an implementation of the above service (*e.g.* in nesC for the Mica2 sensor platform). The representation of this simple uQuery comprises a single task with a single `Capture Camera` step, which specifies a camera sensor id as a property.

More realistic uQueries, however, require a higher degree of concurrency, control (iterative and conditional), and nesting of tasks. For instance, consider Liz's query that asks for a vehicle arrival time histogram for a period of two weeks. The additional ontologies required to specify steps for this uQuery are shown in Figure 3. We denote the return value of tasks by an upward arrow, and the "contributes" relation of Figure 1 by a dashed arrow. The entire box represents a uQuery. The boxes 2, 3 and 4 represents tasks. The rest of the boxes represent steps. Due to shortage of space, we skip the details of the ontology and services required.

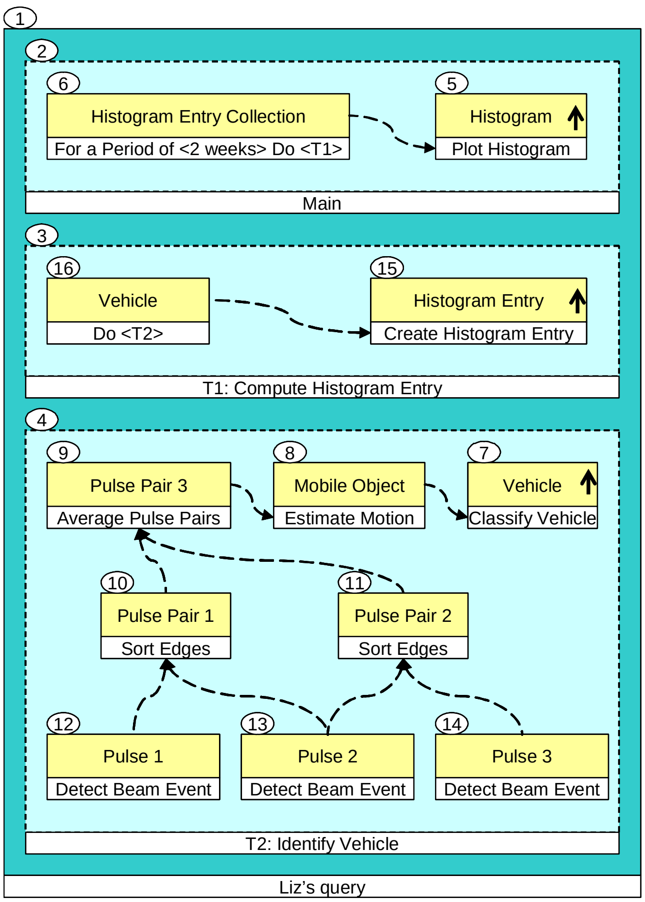



**Figure 3: Representation of Liz's query by three tasks (Numbers in ovals are used here for annotation, and are not a part of the query representation).**

The corresponding uQuery, called `Liz's query` (1),[1] is represented as three interrelated hierarchical tasks: a main task (2), and two *subordinate* tasks, called `T1: Compute Histogram Entry` (3) and `T2: Identify Vehicle` (4). The `main` task has two steps: `Plot Histogram` (5), which returns a Histogram object to the user, and the control construct `For a Period Do` (6) that is in charge of executing `T1` for a duration of two weeks. `T1` is then subordinate to step (6), which controls its execution. It in turn comprises two steps. The first step instantiates `Create Histogram Entry` (15), which returns a Histogram Entry object (to the main task). This requires as argument a Vehicle object, provided by the second step of `T1`, `Do <T2>` (16), which nests `T2` as computation method. `T2` comprises several steps (7-14), whose representation is analogous to that of the `Capture Camera` step described above. The task `T2` uses inputs from multiple break beam sensors to detect moving objects and classify them as vehicles.

[1] In this paragraph, numbers between parentheses, *e.g.*, (1), refer to the numbered rectangles in the Figure 3.

To summarize, this uQuery representation contains occurrences of the following kinds of steps:

- *Primitive calls*: designed as steps that call a primitive as their computation method,
- *Control structures*: designed as steps that control the execution of other tasks, such as the `For a Period Do` step that controls `T1`,
- *Nested tasks*: designed as steps that use a task as their computation method, *e.g.*, the `Do` step that uses `T2`, simplify the specification of complex uQueries by hierarchically modularizing them.

The syntax of Dart is *recursive*, *i.e.*, steps may hierarchically point to tasks. It is also extensible, *i.e.*, *tasks* and *steps* may be used to extend constructs. Arbitrary control constructs can be implemented by uQuery Engine programmers using these properties, thus extending the abstract notion of a construct in Dart. We provide by default an abstraction called *Control Construct*, which has an instance variable called *closures*, which keeps track of a collection of subordinate tasks whose number and semantics depend on the desired functionality, based on the meta-data held by its contract. In the case of the `For a Period Do` control structure, which iteratively executes a given task for a duration specified as a property, the `closures` collection holds one task, here a pointer to `T1`. Its `propertyValues` instance variable, inherited from its superclass, holds the duration value, *i.e.*, 2 weeks in our example.

The representation of nested tasks is close to that of control structures. The `closure` collection contains only one task, which is executed unconditionally upon each activation of the step.

## 4 Concurrent Execution of uQueries

We now explain how the above representations are concurrently executed by our system. We separate the *actor* at the operational level, which is a thread and associated code actually executed as a mobile agent, from the *meta-actor* at the knowledge level, which is a thread that controls the generation of code and execution of the actor, based on the meta-data contained in the query representation. Once deployed, the agent system interprets the actor code and starts its execution. When an actor sends a message, such as the return value, to the uQuery Engine, the messaging interface signals the meta-actor waiting on messages from this actor and delivers the message.

### 4.1 uQueries and Tasks

Execution of uQueries, tasks and steps is controlled by corresponding meta-actors. Meta-actors are also implemented as actors, executing concurrently in separate threads, communicating and synchronizing through asynchronous message passing. The API of meta-actors comprises `start()`, `body()`, and `stop()` methods. `body()` specifies the core function of the meta-actor, in fours steps:

- Checking preconditions for executability,
- Initializing data structures,
- Executing the behavior,
- Sending an asynchronous message to the parent to notify the execution result.

The behavior execution step is specific to each kind of meta-actor. *uQuery meta-actors* start with launching a meta-actor for their main task. The *main task meta-actor* then continues the execution by launching *step meta-actors*. A step meta-actor can only execute in two cases: (1) it is associated to a *root step*, *i.e.*, a step whose construct requires no arguments, or (2) all its effective arguments are available. The execution strategy associated with each step is in charge of guiding this process.

uQuery and task meta-actors are designed as *Composite Meta-actors*, which provides functionality for launching a set of child meta-actors, here respectively tasks and steps, and controlling their execution. If the construct of a step points hierarchically to tasks and subordinate tasks, then, the corresponding step meta-actor creates also hierarchically *subordinate task meta-actors*. The above execution algorithm is applied recursively to the latter.

For example, a uQuery meta-actor is created and launched for executing Liz's query. This meta-actor launches a task meta-actor for the main task, which in turn launches a step meta-actor for its unique root step, *i.e.*, `For a Period Do`. The other step of the main task is not executable since it needs an argument. The `For a Period Do` meta-actor starts a timer to control the number of iterations. Each iteration consists of launching a subordinate task meta-actor for executing the same task, `T1`. The execution result of these meta-actors is collected and constitutes the result of the execution of the `For a Period Do` step. At this point, the main task meta-actor passes this collection to the `Plot Histogram` meta-actor and resumes it.

In the case of `T1`, there is also a unique root step, which is `Do <T2>`. The execution of the associated `T1` meta-actor consists of creating a subordinate `T2` task meta-actor. The `T1` meta-actor sleeps then on a semaphore, waiting for `T2` meta-actor to return a Vehicle or signals an error.

### 4.2 Low-level Primitives

By low-level primitives we mean services implemented by the library available at the operational level (ActorNet in our case). Such services provide access to platform-specific sensing and data processing functionality. Executing steps that instantiate primitives requires dynamically generating and deploying actor code to the WSN.

The actor template (see Figure 2) is filled by the meta-actors as the steps of the graph data structure for each task are traversed. Each step meta-actor which has its preconditions satisfied and has been signaled, directly or indirectly, by a returning step of the main task, can generate the code for its associated actor, in parallel with other meta-actors.

The resulting actors, in text form, are deployed to the WSN by the meta-actors, through the *Actor Deployment Interface* (ADI). The ADI is a multithreaded server providing socket connections over the Internet for concurrently and remotely deploying and executing actors. The id of its generating meta-actor is included in the actor code, enabling the actor to communicate with its associated meta-actor through the *Actor Messaging Interface.* The meta-actor can sleep until it receives the actor's output object through the messaging interface, after the actor finishes its computation. Arguments and return values are marshaled and unmarshaled using the information kept by the Execution Strategy. In case a primitive cannot successfully complete its computation, a failure is signaled to the relevant meta-actor, which may trigger garbage collection for the task and its associated actors and meta-actors

For example, the `T2` meta-actor concurrently generates and deploys on the ActorNet platform mobile actor code for the three `Detect Beam Event` steps shown is Figure 3. As Beam Event objects are received, the `Extract Edge` meta-actors may be signaled concurrently, and so on.

### 4.3 Complex uQueries

Now let us consider a query which requires distributed coordination between different sensors. Such a query cannot be executed as a single mobile agent. For example, Pablo wants to take a photo with a camera *only* when a nearby break beam sensor is triggered. The task that represents this query has two steps: detecting a Pulse object and then taking a Photo.

For executing such queries in a coordinated manner, we use the *argument* relation in our meta-model. Whenever an argument is encountered in the query specification, the knowledge level automatically creates a dependency relation between the meta-actors for these two steps. Meta-actors with dependencies check whether they have all the arguments necessary to generate and execute their associated actors. In our example, the meta-actor for capturing a Photo will be made dependent on the `Detect Beam Event` meta-actor. It will be deployed and executed only when a Pulse object is returned by the latter.

We can now consider a scenario where uncoordinated concurrent queries are entered into the system by two different users, Liz and Pablo. Pablo is interested in taking photos of Vehicles when they speed through break beam sensors. We assume that Liz's query remains the same as in the previous section. Each query is represented independently, *i.e.*, there is no structural relationship between query representations. In this case, the execution for each query is triggered concurrently at the knowledge level, according to the algorithm described above. This execution procedure can be subject to optimization, as discussed below.

## 5 Optimization

The uQuery Engine design presents several opportunities for optimization. First, in the case of concurrent queries, data computed in one query can be reused to satisfy requirements of another. This mechanism is based on exploiting static and dynamic metadata (see Section 2.1).

For example, a Vehicle object produced by one of the queries described above contains dynamic metadata such as a timestamp and the detecting beam sensor's id. When processing a query with a step requiring a Vehicle object as argument, *e.g.*, `Compute Histogram Entry` in task `T1` above, the already-computed Vehicle may be substituted instead of executing the `Identify Vehicle` subtask, assuming the dynamic metadata of the object matches the constraints within the static metadata of the query. Such matching algorithms can be implemented using an inference engine, such as *NéOpus* [Pachet and Perrot, 1994], which provides rule-based reasoning in object-oriented applications. The benefit of this process is to save significant communication, sensing and computation resources in the WSN, where these resources are scarce, since redundant actors are not generated, deployed and executed at the operational level.

Secondly, our query representation allows computing a *computation history* for every business object by looking recursively at the constructs and objects used in its computation. The computation history allows access to the metadata associated with each element of the history. As a result, more complex matching algorithms can be implemented. For example, a query can compare the conditions under which the available business object has been computed, and whether they match the requirements of the current query. Incidentally, computation histories are also useful for *auditability*, that is identifying precisely all computation steps and their contexts for a given event in the system. The same feature allows satisfying *non-repudiability*, *i.e.*, by computing who has initiated which action.

As another optimization opportunity, we consider the *functional specification* of queries, where the user only specifies the desired functionality without explicitly listing all implementation details, such as the identifiers or location coordinates of sensors. The query processing engine can infer the requirements for these resources from the static metadata associated with the query and the contracts of the primitives it invokes. We can then fill in specific sensors and other parameters matching these criteria. In order to find suitable sensing resources meeting these constraints, we extend ActorNet with the *Network Directory Service* (NDS), which performs network monitoring and provides an up-to-date listing of sensor and computing resources available in the network. The NDS is searchable by sensor type, location and other attributes, listed as name-value pairs. Thus the query engine can find appropriate sensors for a query by looking at attributes such as location and orientation. This simplifies the query creation process for the user and provides opportunities for optimizing query execution, by enabling the uQuery Engine to pick the best resources (*e.g.*, closest, least congested, or having the most remaining battery power) satisfying the query specification.

Finally, at the operational level, the mobility of actors allows for load balancing and resource-aware task allocation, even as new actors are added. In the parking garage example, we can choose, based on the current communication traffic and available computing power, whether to move the raw break beam sensor data to a PC for processing, or to move the vehicle detection code, as a system of mobile agents, to the sensor where the data is generated. Also, by exposing available resources (sensors, actuators, data stores, *etc.*) to the knowledge level, compatible resources may be substituted for one another at runtime to simplify scheduling or reduce congestion.

## 6 Related Work

A survey of solutions currently proposed in the literature reveals a variety of approaches to macroprogramming WSNs: a spreadsheet approach [Woo *et al*., 2006], *EnviroSuite* [Luo *et al*., 2006], a market-based approach [Mainland *et al*., 2004], and *Semantic Streams* [Whitehouse *et al.*, 2005]. Although many of these approaches are quite powerful, none of them provide the language abstractions required for dynamic macroprogramming by end-users as outlined above.

For example, the spreadsheet approach uses an Excel spreadsheet to represent the layout of nodes and insert their functionality in the spreadsheet; queries are resolved by a logic program that generates a composition of services, where a service is a .Net component. The approach satisfies the requirement of usability by non-programmers. However, it is not sufficiently general: it enforces a particular naming grid-based scheme and does not allow for the definition of arbitrary groups of nodes and operations over such groups.

EnviroSuite proposes environmentally immersive programming, an object-based programming model in which individual objects represent physical elements in the external environment. In both EnviroSuite and ActorNet, actors or objects must be created explicitly by programmers to provide a service. Behavioral specifications are not in terms of groups of actors. Protocols to support operations over groups of objects and protocols to implement such specifications may not be re-used.

Traditionally, WSN application development involved fairly static programming languages, operating systems and reprogramming services, for efficiency reasons. For example in TinyOS, the sensor network application components are written in nesC and compiled together with the operating system code and middleware services into a single application image, which can be uploaded to the sensor nodes using the Deluge protocol [Hui and Culler, 2004] prior to program execution. This approach proves successful in achieving its stated goal of highly efficient utilization of sparse computing resources. Unfortunately, it is ill-suited for an open system comprising a dynamic set of diverse, transient tasks that is the expected workload in ambient systems. If we take Deluge as the deployment method for our target systems (where queries from multiple users, all specified at runtime, need to be transformed into executable code, uploaded on the deployed sensor network and executed), this results in unnecessarily transmitting a large volume of code which has not changed (OS components, routing protocol, *etc.*) along with the newly-generated application code.

## 7 Conclusion

Ambient Intelligence technologies will enable novel applications and new work practices in many fields. AmI will provide for the integration of real-time data into business processes, enabling real-time decision making and business process definition and modification. Such dynamicity will facilitate responding to situations more efficiently, with a higher degree of quality and end-user satisfaction.

In this paper, we explained how dynamic macroprogramming by end-users can be achieved for ambient systems such as WSNs. The two-level approach to architecting the query engine allows separating query representation and reasoning concerns from those of their effective execution on diverse runtime platforms through model-to-code transformation. The knowledge level comprises the representation of a domain ontology and a service repository, together with the uQuery composition machinery, and implements the rules that govern uQuery transformation, coordination and optimization. Our system allows for effective end-user query specification and automated execution. The query representation meta-model, designed with extendibility and reusability as primary considerations, allows uQuery engine programmers to add specific constructs via classical object-oriented techniques. Business logic primitives are separated from the core of the mobile agent system, facilitating addition of new domain-specific primitives to the system.

Representations of queries are transformed to platform-specific code for ActorNet, dynamically deployed and executed. Using ActorNet as the query execution environment provides dynamicity of macroprogramming, while enabling load balancing and other optimizations to take place.

The presented meta-level architecture for implementing uQuery Engines is prototyped by reusing and extending our previous work on ActorNet and its implementation on Mica2 motes [Kwon et al., 2006], and Dart implemented as an object-oriented framework in different dialects of Smalltalk [Razavi et al., 2006]. This prototype is called *Ambiance Platform* and uses a new implementation of Dart in *Squeak* (http://www.squeak.org/). We further use the *Seaside* framework (www.seaside.st) for the dynamic Web-enabled uQueries programming interface of Ambiance.

A number of important issues remain to be explored. These include dynamic and seamless integration of sensor and actuator nodes, sensitivity to privacy concerns and trustworthiness, and analyzing coherence and integrity of uQueries. For example, the computation history discussed in Section 5 allows security enforcement through dynamic decision making about whether to execute the current step or not. Furthermore, significant optimization may be possible, for instance by integrating learning mechanisms into query specification, transformation and execution.

## Acknowledgments

This work is partially funded by the University of Luxembourg, in the framework of the *Åmbiance* project (R1F105K04) and NSF under grant CNS 05-09321, by ONR under DoD MURI award N0014-02-1-0715 to UIUC. The authors would also like to acknowledge the valuable collaboration of F. Arbab, A. Cardon, N. Bouraqadi, P. Bouvry, Ch. Dony, V. Ginot, R. Kumar, R. Johnson, M. Malvetti, S. Nadjm-Tehrani, D. Riehle, S. Sundresh, and J. Yoder.

## References

[Agha, 1986] Agha, G.: “Actors: a Model of Concurrent Computation in Distributed Systems”. MIT Press, 1986.

[Hui and Culler, 2004] Hui, J.W. and Culler, D.: “The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale.” In SenSys’04, pp. 81-94, 2004.

[IST, 2003] IST Advisory Group. Ambient intelligence: from vision to reality, September 2003.

[Kwon *et al*., 2006] Kwon, Y., Sundresh, S., Mechitov, K., Agha, G.: “ActorNet: An Actor Platform for Wireless Sensor Networks.” In AAMAS, 2006.

[Luo *et al*., 2006] Luo, L., Abdelzaher, F., He, T., Stankovic, J.A.: “EnviroSuite: An Environmentally Immersive Programming Framework for Sensor Networks.” ACM Transaction on Embedded Computing Systems, 2006.

[Mainland *et al*., 2004] Mainland, G., Kang, L., Lahaie, S., Parkes, D.C., and Welsh, M.: “Using Virtual Markets to Program Global Behavior in Sensor Networks.” 11th ACM SIGOPS European Workshop, Belgium, 2004.

[Pachet and Perrot, 1994] Pachet F, Perrot JF.: “Rule firing with metarules.” In SEKE’94. pp. 322–29, 1994.

[Razavi et al., 2006a] Razavi, R., Mechitov, K., Sundresh, S., Agha, G., Perrot, J.-F.: Ambiance: Adaptive Object Model-based Platform for Macroprogramming Sensor Networks. Poster session extended abstract. OOPSLA 2006 Companion. Portland, Oregon, USA (2006 ).

[Razavi *et al*., 2006b] Razavi, R., Perrot, J.-F., Johnson, R.: Dart: A Meta-Level Object-Oriented Framework for Task-Specific, Artifact-Driven Behavior Modeling. In Proceedings of DSM’06, Portland, Oregon, USA. ISBN 951-39-2631-1, pp 43-55, 2006.

[Whitehouse *et al*., 2006] Whitehouse, K., Zhao, F. and Liu, J.: “Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data.” In EWSN’06, Switzerland, pp. 5-20, 2006.

[Woo *et al*., 2006] Woo, A., Seth, S., Olson, T., Liu J., and Zhao, F.: “A Spreadsheet Approach to Programming and Managing Sensor Networks.” In IPSN SPOTS, 2006.

# XPod: A Human Activity Aware Learning Mobile Music Player

**Sandor Dornbush, Jesse English, Tim Oates, Zary Segall, Anumpam Joshi**
University of Maryland, Baltimore County
{sandor1, english1, oates, zary, joshi} @ umbc.edu

## Abstract

The XPod system, presented in this paper, aims to integrate awareness of human activity and musical preferences to produce an adaptive system that plays the contextually correct music. The XPod project introduces a "smart" music player that learns its user's preferences and activity, and tailors its music selections accordingly. We are using a BodyMedia device that has been shown to accurately measure a user's physiological state. The device is able to monitor a number of variables to determine its user's levels of activity, motion and physical state so that it may predict what music is appropriate at that point. The XPod user trains the player to understand what music is preferred and under what conditions. After training, the XPod, using various machine-learning techniques, is able to predict the desirability of a song, given the user's physical state.


Figure 1: Proposed XPod Form Factor

## 1 Introduction

In this paper we study the problem of creating a music player that plays contextually correct music by using a system of physiological sensors to monitor a user's state. We propose using machine learning algorithms to estimate a user's preference of songs in various situations. Several machine learning algorithms were used to model the complex relationships between an individual's musical preferences and his or her activities. We studied several machine learning systems on a modified version of the existing mobile MP3 player, XPod. This player will attempt to select the song best suited to the current activity of the user. XPod was previously reported on in [Dornbush *et al.*, 2005], wherein the system was designed to use a database and a neural network to suggest music to users. In this paper we expand on that work, and approach many different machine-learning algorithms, with varied results.

This paper is layed out in the following format: in Section 2, we will review similar work in this field. In Section 3, we will discuss the motivation for creating such a device. In Section 4, we will speak about the data collected and how it was used to produce learning algorithms. We will then discuss the results of five experiments in Section 5, and end with a discussion on future research in this area in Section 6.

## 2 Related Work

Context aware systems can greatly improve a users experience with computer systems. For example several systems relate user activity to mobile phones [Bylund and Segall, 2004; Siewiorek *et al.*, 2003]. This paper proposes an extension to the mobile MP3 player, XPod, which is able to automate the process of selecting the song best suited to the current activity of the user. That system used a collection of machine learning techniques to play contextually correct music. The array of user state information was converted to one of three states (active, passive and resting) using a decision tree. The state was used as a key into the database of state dependant ratings. That was combined with the results from a neural network that estimated the users preference of a song. While this showed the potential of an adaptive context aware music player it had significant limitations.

The concept of a music player that is aware of the user's activity has made it into the mainstream market with industry leaders Nike and Apple teaming up to deliver an iPod that wirelessly communicates with a sensor in a Nike running shoe[Nike and Apple, 2006]. That system has had limited market success so far, but has shown that there is consumer interest. The primary use case for this system is to record and analyze a runners athletic performance. That system is built to facilitate the user's athletic training, not to allow the user to train the system. We uniquely address the problem of a context aware music player that learns a users preferences.

Other researchers have studied the relationship between a users activity and the music selection played for them. In [Park *et al.*, 2006] the authors explore the idea of using Bayesian networks to predict the proper music for the current situation. The authors used a fairly small set of attributes to describe the context. The authors only used attributes from the large area context such as weather, user gender, time and season.

Sonic City[Gaye *et al.*, 2003; Gaye and Holmquist, 2004] developed a wearable jacket that created music based on the sensed light, noise and movement. In this experiment the user and the environment together create music.

[Elliott and Tomlinson, 2006] developed a system that correlates the song played to a users pace. This system used machine learning to determine which song to play, matching the song's beats per minute(BPM) to the users pace. While this naive approach provides a level of context aware music it is not sufficient. For one this does not address the question of what music to play when a user is not active or when participating in an activity such as bicycling where the concept of pace is not well defined. Secondly in our experiments BPM is not sufficient to characterize a piece of music. We found some pieces of music with high BPM that our user found suitable to low activity level, and pieces of music with low BPM that were suitable for high activity level.

## 3 Motivation

The XPod concept is based on the idea of automating much of the interaction between the music player and its user. The XPod project introduces a "smart" music player that learns its user's musical preferences for different activities, and tailors its music selections accordingly. The device is able to monitor a number of variables to determine its user's levels of activity, motion and physical states at the current moment and predict what music would be appropriate. The XPod user trains the player to predict the preference of music under various conditions. After an initial training period, XPod is able to use its internal algorithms to make an educated selection of the song that would best fit its user's emotion and activity.

Before playing a song the internal algorithm is used to predict the users rating of that song in their current state. That prediction is used to weigh the chance that the current song will be played. A song with a low expected rating may be skipped in the current state. Every song has a chance of being played at any time. This is done so that the XPod explores the feature space and does not get stuck playing a few songs. For example if a song is rated zero once, that should not be interpreted rating that song as zero in all contexts. In a different state that song might be rated a four. Our goal would be to model the level of knowledge for every song; then use that

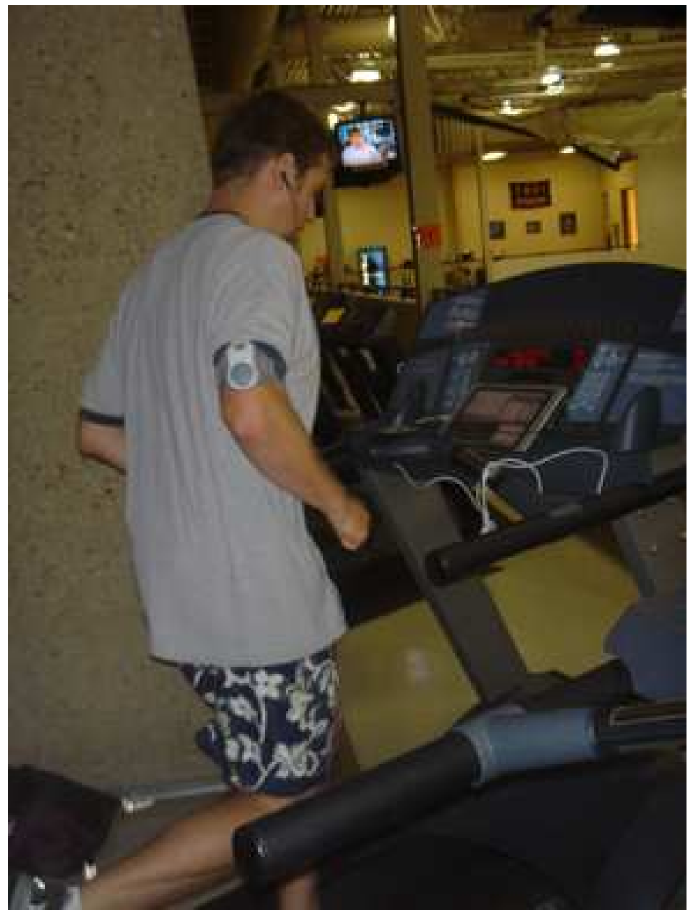


Figure 2: An author collecting training data.

| | Name | Type |
|---|---|---|
| Input | Galvanic Skin Response | Real value |
| | Mean Acceleration Longitudinal | Real value |
| | Std. Dev. Acceleration Longitudinal | Real value |
| | Mean Acceleration Transversal | Real value |
| | Std. Dev. Acceleration Transversal | Real value |
| | Skin Temperature | Real value |
| | Heat Flow | Real value |
| | Heat Flow Cover | Real value |
| | Transversal Cadence | Integer |
| | Longitudinal Cadence | Integer |
| | Time of Day | Integer |
| | Day of Week | Integer |
| | Song Genre | Symbolic |
| | Song Artist | Symbolic |
| | Song Album | Symbolic |
| | Song Title | Symbolic |
| | Beats Per Minute | Integer |
| Output | User's Action | Integer{0-4} |

Table 1: Input and output fields for XPod classifiers.

| Genre | Artists | Tracks |
|---|---|---|
| Alternative Rock | Red Hot Chili Peppers | 17 |
| Blues | Louie Armstrong | 20 |
| Electronic | M.I.A. | 12 |
| Funk | Mofofunka | 15 |
| Hip-Hop | Black Eyed Peas | 14 |
| | Digable Planets | 11 |
| | Kanye West | 20 |
| Jazz | Art Blakey | 8 |
| | John Coltrane | 13 |
| | Miles Davis | 10 |
| Reggae | Bob Marley | 10 |
| Rock | Beatles | 11 |
| | Phish | 16 |
| | Rolling Stones | 10 |
| | Grateful Dead | 8 |
| | Jimi Hendrix | 6 |
| | Santana | 9 |
| Ska | Tokyo Ska Paradise Orchestra | 27 |

Table 2: Music library.

model to weight the trust we have in the rating. At this point we used a fixed discount of the predicted rating.

We propose a form factor illustrated in Figure 1 where the device is mounted on an armband matching the current widespread usage patterns of MP3 players. In addition a device on the arm can capture an accurate view of how a user is moving his or her body.

## 4 XPod Dataset

The XPod system is comprised of a standard MP3 playing device and a human body sensor. The device tracks and stores a record of song meta-data as a song is played, including artist, album, genre, title, and beats-per-minute. In addition, the system records the time of day, a user's rating (from 0 to 4 stars), and a full range of physical responses from the user's body. The full set of attributes recorded are detailed in Table 1. These measurements include skin temperature, heat flow, two dimensions of acceleration, cadence, and galvanic skin response, which is a measure of how much sweat is on the user's skin. Each of the symbolic attributes, artist, album, title, and genre are all expanded into a large number of binary attributes. This was done so that the symbolic attributes could be accurately handled by the numerical algorithms, such as SVMs and neural networks. For this reason the total number of attributes in our experiments, including the user state, is 289. Typically each instance is a sparse array with most attributes set to 0 or false.

To gather the information about the user's physical state, a BodyMedia [BodyMedia, 2006] device was used. This device straps on to the arm, and broadcasts its readings wirelessly to a nearby system, which records the data for use by the XPod. The BodyMedia device is capable of monitoring a user's physiological and emotional state[Nasoz *et al.*, 2003]. In this paper we focus on the physiological state; however this system should be able to adapt musical preference to the emotional state.

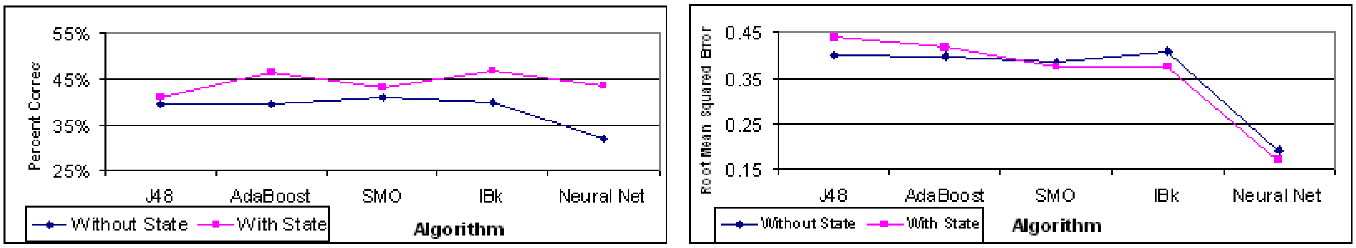
## 5 Machine Learning Algorithms

To test the XPod, we trained several independent learning algorithms on our test data. To construct our dataset, we gathered 239 different mp3 song files. Each song was analyzed to find the beats per minute. A researcher collected training information using a prototype system. The prototype shown in Figure 2 involved a tablet computer and a BodyMedia device.

A researcher on the XPod team proceeded to record training instances in a variety of physical situations (exercise, mild activity, rest, etc.). A training instance, or data point, includes a value for each field in Table 1. 565 training instances were recorded. For each instance the XPod player would rate a song and play that song. If the rating matched the researcher's preference he took no action. If the rating did not match the preference the researcher gave the song a rating from 0 to 4, reflecting how appropriate the researcher felt the song was at that time. A rating of 0 would result in the music player skipping the remainder of the song. Each classification algorithm was trained on some or all of the training instances. That training was used to predict how a user would rate a song in the future.

It is our goal to show that a music player can choose the contextually correct music if it uses information about a user's physiological state. To prove this theory we created two sets of machine learning systems, those trained with user state information, and those without user state information. "State" refers to the array of information gained from the BodyMedia device, as well as any other outside information, such as date and time. Due to our small sample size, activity can be inferred from the date and time. For this reason those variables where included in state information. We will consider our experiment successful if the system is more accurate when it has access to the state information than when it does not have access to state information.

We used 10 fold cross-validation to measure the accu-




(a) % Accuracy of various learning algorithms.

(b) RMSE of various learning algorithms.

Figure 3: Performance of learning algorithms

racy of the machine learning algorithms. We also experimented with Leave-One-Out-Cross-Validation(LOOCV). We found very similar results between the two methods. Since LOOCV is much more expensive we have reported the results of 10 fold cross-validation. We used classifiers from the open source Weka library[Witten and Frank, 2005] and neural networks from the open source Joone library[Marrone and Team, 2006].

In the following experiments each training instance could be classified into one of five classes. The expected performance of a random algorithm would be 20%. All of the algorithms performed significantly better than random.

### 5.1 Decision Trees

The first classifier used was the decision tree algorithm (J48)[Quinlan, 1993]. When learning without state, the decision tree was able to properly classify the training data 39.47% of the time. However, when using state, the decision tree was able to properly classify the training data 41.06% of the time. The accuracy of decision trees was not the best in the survey, however they do show a slight (2%) advantage of using state information in the learning algorithm. The poor generalization of J48 can be seen in the fact that the system not using state information had less error than the system that did use state information.

### 5.2 AdaBoost

The J48 classifier improved significantly when it was boosted with AdaBoost (AdaBoostM1)[Freund and Schapire, 1996]. This classifier was correct against the training data 39.47% without state, and 46.55% with state. AdaBoost suffered from the same generalization performance inversion as the original J48 classifier. This showed that AdaBoost is somewhat effective at increasing the performance of the J48 algorithm.

### 5.3 Support Vector Machine (SVM)

The third classifier we experimented with was support vector machines (SMO)[Platt, 1998; Keerthi *et al.*, 2001]. SVM's generalized well and had a little improvement when using state (43.19%) over not using state (40.89%). In this case the SVM was almost able to divide the dataset into the researcher's preference based solely on the musical data. When adding in state, the dimension space changed minimally, adjusting enough to shuffle a few incorrectly classified instances to the proper area.

The small difference between the SVM trained with state information and without state information, (2%) is likely a result of the relatively large feature space and small training set. The expressiveness of the second order kernel allows the SVM to identify the user's preference without the state information.

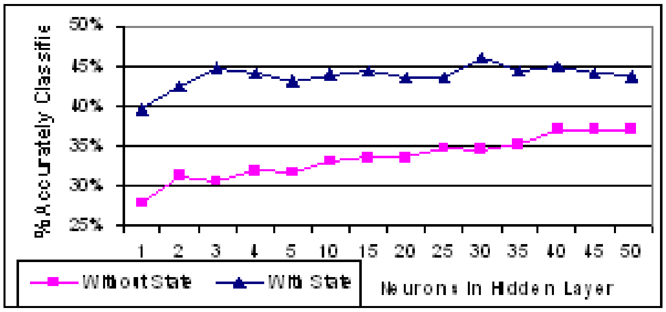
### 5.4 K-Nearest Neighbors(KNN)

We had surprisingly positive results from the lazy classifier: k-nearest neighbors (IBK)[Aha and Kibler, 1991]. We allowed Weka to choose the optimal number of neighbors. The best number of neighbors was found to be 9. Results showed a 7% increase in accuracy when using state (46.72%) over not using state (39.82%). More importantly KNN had a low root mean squared error (RMSE) (0.3753).

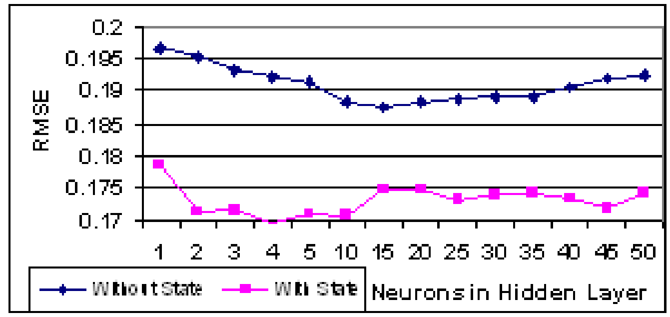
### 5.5 Neural Networks

We had very promising results from a neural network trained on this data-set. We created a three layer network with 288, or 276 inputs depending on whether state information was used. A small hidden layer and a single neuron output was used. We experimented with a variety of different size hidden layers from 1 to 50. The results of these experiments are shown in 4(a) and 4(b) We found very similar results with a small number of hidden nodes, 3, as when we used a large number of hidden nodes, 50. As the size of the hidden layers grows, the accuracy of the network using state information does not increase much. However the accuracy of the network not using state information does increase. We believe that the more complex networks are better at memorizing erroneous information to accurately rate the songs.

We had difficulties with over training. The network would find the best validation error in the first 100 training epochs. We used early stopping to keep the best network on the validation data. We are investigating ways to avoid this problem. We were able to achieve respectable performance with a network given state information. That network correctly classified instances 43.54% much better than the 31.87% accuracy without state information. This is not the best percent accuracy, however it did get the best results in terms of RMSE (0.17). The neural network had a fraction of the RMSE of the other methods.




(a) % Accurately identified using different size hidden layers.




(b) RMSE using different size hidden layers.

Figure 4: Performance of different size networks.

## 6 Conclusion and Future Work

Our goal is to show that a music player trained with a user's physical activity and preference can choose the contextually correct music. All of the systems evaluated performe significantly better than random(20% accuracy). As shown in Figure 3(a), given state information every system chooses the exactly correct label more often than the same algorithm without state information. For each testing instance there are five possible categories. Figure 3(b) shows that the tree based algorithms tend to generalize poorly, evidenced by the fact that systems using state information had greater RMSE than the systems not using state information. The other algorithms were able to generalize well and achieved high accuracy and low RMSE.

We believe that if we collected still more training instances the difference between the performance of the statefull and the stateless system would grow. Presumably if we collected enough training instances we would find a pair of instances that are identical on all non-state attributes but have different ratings. Then any classifier without state information would have to give both instances the same label. However only one could possibly be correct. A classifier given the same instance including the state information has a chance to classify both instances correctly. Therefore if we collected more training instances the difference between stateful and stateless systems should increase.

Although the lazy classifiers tended to perform well, in practice this may not be the case. Specifically, a portable music device is not likely to have high processing power. Given an active user of such a device, listening for multiple hours a day over the course of one or two years, the device would search an instance space of over tens of thousands of data points. Performing a calculation like this may be more than inefficient: it could be wholly impractical.

Support vector machines may be well suited to the task as they can begin to classify new instances having very little training data to build on. From the end-user's perspective, this is a desirable feature, as the user would need to spend very little time setting up the system, and more time enjoying the benefits. Further, SVMs are capable of classifying in a very high dimensional space while only performing calculations in a much smaller number of dimensions. However it is not clear if SVMs could be created on a constrained device. Perhaps the SVMs would be created on a unconstrained device such as a PC. Then the trained SVM would then be transferred to the portable device. Even constrained devices can evaluate a SVM.

Decision trees would likely be the most computational feasible classifier, as they can be converted into a rule set, which can be evaluated very rapidly. As we've shown in this application, decision trees perform better with boosting.

Our view is that the neural network is the most promising result. Although it did not get the highest exact accuracy it tended to get very close to the right answer, reflected in the small RMSE. Since the result was used to influence pseudo-random choice of music it is actually more important to be close than to be exactly accurate. For example if the correct rating should be 4 but the system responds with 3, that will result in a low RMSE, but will not count towards the percent correct. A rating of 0 would result in the same percent accurate, however a much higher RMSE. Since a song rated 0 would be skipped, but a song rated three or four would likely be played, a close answer is almost as good as the correct answer. For that reason we feel that the low RMSE of the neural network indicates that it would be the most useful algorithm. Many embedded devices such as mobile phones already employ neural networks, therefore it should be possible to use neural networks in mobile music playing devices.

In future work we will investigate other meta-data that could be associated with the music. We have used relatively simple music analyzing software to find the beats per minute, however it is possible to find much more by analyzing the music[Logan and Salomon, 2001]. It would also be interesting to investigate human generated meta data in community systems such as the Pandora Project[Westergren, 2006] or Audoscrobbler[Audioscrobbler, 2006]. Any new meta-data regarding songs could be included as additional inputs into the machine learning algorithms. We will investigate augmenting the training instances already collected with additional meta-data. Our goal will be to see if there is a significant increase in performance given new information.

We will investigate prototyping this system in a physical device. While the BodyMedia device provides many different attributes a satisfactory system could likely be built with a selection of those attributes. An inspection of the decision tree built by J48 shows 20 decisions based on acceleration, almost four times more than the sum of all decisions based on other state variables.

Nokia research has generously granted our research group two 5500 Sport phones[Nokia, 2006]. These phones have accelerometers to measure a users activity level and the capability to play MP3's. We expect to implement the XPod system on this device in the near future.

We have shown the relative advantages of different machine learning systems at choosing the contextually correct music. People have shown an interest in this type of system. However more works need to be done to further refine this system.

## 7 Acknowledge

We would like to thank Chad Eby for the renditions of the proposed XPod form factor and Nokia for the donation of the 5500 Sport phones.

## References

[Aha and Kibler, 1991] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[Audioscrobbler, 2006] Audioscrobbler. http://www.audioscrobbler.net/, 2006.

[BodyMedia, 2006] BodyMedia. http://www.bodymedia.com/, 2006.

[Bylund and Segall, 2004] M. Bylund and Z. Segall. Towards seamless mobility with personal servers. *INFO Journal*, May 2004.

[Dornbush *et al.*, 2005] Sandor Dornbush, Kevin Fisher, Kyle McKay, Alex Prikhodko, and Zary Segall. XPod a human activity and emotion aware mobile music player. In *Proceedings of the International Conference on Mobile Technology, Applications and Systems*, November 2005.

[Elliott and Tomlinson, 2006] Greg T. Elliott and Bill Tomlinson. Personalsoundtrack: context-aware playlists that adapt to user pace. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 736–741, New York, NY, USA, 2006. ACM Press.

[Freund and Schapire, 1996] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. pages 148–156, San Francisco, 1996. Morgan Kaufmann.

[Gaye and Holmquist, 2004] Lalya Gaye and Lars Erik Holmquist. In duet with everyday urban settings: a user study of sonic city. In *NIME '04: Proceedings of the 2004 conference on New interfaces for musical expression*, pages 161–164, Singapore, Singapore, 2004. National University of Singapore.

[Gaye *et al.*, 2003] Lalya Gaye, Ramia Mazé, and Lars Erik Holmquist. Sonic city: the urban environment as a musical interface. In *NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression*, pages 109–115, Singapore, Singapore, 2003. National University of Singapore.

[Keerthi *et al.*, 2001] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, pages 637–649, 2001.

[Logan and Salomon, 2001] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *IEEE International Conference on Multimedia and Expo, ICME*, 2001.

[Marrone and Team, 2006] Paolo Marrone and Joone Team. Joone, 2006. http://www.jooneworld.com/.

[Nasoz *et al.*, 2003] Fatma Nasoz, Kaye Alvarez, Christine L. Lisetti, and Neal Finkelstein. Emotion recognition from physiological signals for presence technologies. *International Journal of Cognition, Technology and Work - Special Issue on Presence*, 6, 2003.

[Nike and Apple, 2006] Nike and Apple. http://www.apple.com/ipod/nike/, 2006.

[Nokia, 2006] Nokia. 5500 sport phone, 2006.

[Park *et al.*, 2006] H. S. Park, J.O. Yoo, and S. B. Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International Conference on Fuzzy Sytems and Knowledge Discovery (FSKD'06)*, pages 970–979, 2006.

[Platt, 1998] J. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[Quinlan, 1993] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Siewiorek *et al.*, 2003] Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Neema Moraveji, Kathryn Reiger, and Jeremy Shaffer. Sensay: A context-aware mobile phone. In *ISWC*, 2003.

[Westergren, 2006] Tim Westergren. Music genome project, 2006. http://www.pandora.com/.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*, volume 2nd Edition. Morgan Kaufmann, San Francisco, 2005.

# An Active Classification System for Context Representation and Acquisition

**Fulvio Mastrogiovanni, Antonio Sgorbissa and Renato Zaccaria**
Laboratorium DIST, University of Genoa, Italy
{fulvio.mastrogiovanni, antonio.sgorbissa, renato.zaccaria}@unige.it

## Abstract

This paper presents a distributed knowledge representation and active data fusion system suitable for use in Ambient Intelligence applications. The architecture is based on the idea of an ecosystem composed by interacting artificial entities. They collaborate to perform an intelligent multi-sensor data fusion according to the guidance of an active classification layer. The approach has been thoroughly tested in simulation, and part of the architecture has been exploited in many applications.

## 1 Introduction

The Ambient Intelligence (AmI) paradigm is aimed at designing distributed systems which are able to build a context model on the basis of sensor data collected by networks of devices, to reason about such a model and then to interact with users by providing them with context-based services through actuators.

During the past few years, several approaches have been proposed to support the idea of a *Smart Space*, i.e., a functional interface to services designed to improve the users' quality of life. A Smart Space requires efficient methods to manage the information flow exchanged by such services. This complexity arises from the mutual role played by different disciplines, encompassing topics which range from health care monitoring [Barger *et al.*, 2005] to mobile robotics [Broxvall *et al.*, 2006] , from intelligent surveillance applications [Kong *et al.*, 2005][Chen *et al.*, 2003] to knowledge representation [Augusto and Nugent, 2005]. Moreover, the related growth of the networks connecting sensors and actuators to cognitive subsystems requires the aggregation of possibly ambiguous information. As a consequence, reliable data fusion techniques must be used.

The goal of data fusion is to *maximize* the useful information content acquired by heterogeneous sources in order to infer relevant situations and events of the observed environment. In order to be reliable, data fusion does not operate in isolation: on the contrary, it must be supported by *a priori* knowledge guiding data interpretation, and it must cooperate with a system providing data acquisition, filtering, etc.

Several data fusion models have been designed and successfully used in different applications reported in literature. In particular, the extended JDL (Joint Directors of Laboratories) model [White, 1998] hypothesizes that information acquisition can be modeled as a five layers process. With respect to this framework, present AmI systems mainly address data fusion at the *numerical* level, usually exploiting sound and well-defined techniques like bayesian inference (such as the well know Kalman Filter [Dhaher and MacKesy, 2004]), parameter estimation [Kong *et al.*, 2005], Fuzzy Logic [Chen *et al.*, 2003] [Hagras *et al.*, 2003] or other methods, disregarding data fusion at the *symbolic* level.

In this paper we propose a multi-agent cognitive architecture able to actively select the sources of information needed to further classify current situations, according to some predefined model. In Section 2 we introduce the main concepts related to our architectural approach. In particular, we focus on its distributed data fusion capabilities in the context of AmI applications. In Section 3 we focus on the representation at the symbolic level, detailing how this layer guides the low level knowledge acquisition process. Next, actual implementation and experimental results are presented and discussed. Conclusion follows.

## 2 A Distributed Approach to Data Fusion

In AmI, the original definition of an ecosystem [Odum, 1959] can be extended to validate the intuition that a Smart Space is an ecosystem whose artificial entities interact for the fulfillment of their goals: among them, it is possible to identify also the "well being" of the users. The presented architecture is designed to support numerical and symbolic data fusion. Data fusion is considered a decentralized process managed by several cognitive-oriented entities. Therefore, the architecture of the data fusion process is mapped onto the ecosystem' structure.

A Smart Space can be modeled as a collection of agents $\{\alpha\}$, performing different tasks. Agents can be classified into two disjoint sets: whilst device-related $\{\alpha^d\}$ agents manage sensors for data acquisition or actuators, cognitive $\{\alpha^c\}$ agents perform data interpretation and apply control rules to guide the system behavior.

In the context of an AmI system, $\{\alpha^d\}$ agents deal with hardware devices, thus providing cognitive agents with data. Moreover, they can also perform preliminary data analysis and react to emergency situations by simple data inspection (e.g., checking if a numerical value is over a given threshold). $\{\alpha^c\}$ agents apply control rules to incoming data in order to issue low level references back to $\{\alpha^d\}$ agents. They can be organized in networks performing tasks related to data filtering, feature extraction, symbolic inferences, knowledge representation, data fusion and planning.

Despite the fact that an exact formalization is outside the scope of the paper, each agent $\alpha_j$ can be formally defined as a 4-element vector, $\alpha_j = \langle \theta, \gamma, \iota, \omega \rangle$, where $\theta$ specifies the agent capabilities, $\gamma$ is a set of goals that $\alpha_j$ can contribute to achieve, and $\iota$ and $\omega$ are operators returning, respectively, the set of data needed by $\alpha_j$ to perform its tasks and the set of data produced by $\alpha_j$. Agents can aggregate in more complex *niches*, characterized by a common goal $g$ to fulfill. We define a goal $g$ as the result of $n$ cooperating agents. In particular, $g = \omega(A_g)$, where $A_g = \{\alpha_j : j = 1, ..., n\}$ is the niche of the involved agents, and $\omega$ is the previously introduced operator returning the *relevant* output data of $A_g$.

With respect to a higher level of complexity, niches themselves can be modeled as agents and, as such, they are possibly part of larger niches. These considerations allow us to use indifferently the generic term "agent" in the following discussion, and to adopt a recursive definition of agent, $\alpha_k = A_g = \{\alpha_j : j = 1, ..., k-1, k+1, ..., n\}$.

It is worth noting that the given definitions do not assume any specific restriction about agent *situatedness* or inter-agent communication. It seems reasonable to assume that $\{\alpha^d\}$ agents should be considered *embedded* in the devices they are dealing with, while no assumption is made about embodiment of $\{\alpha^c\}$ agents. Nonetheless, if agents are distributed in a network according to specific needs, communication should be guaranteed. Thus, we assume that, for each pair $(i, j)$ of communicating agents, it does exist a communication *channel* $c_{i,j}$, (formally, $\forall \alpha_i, \alpha_j \exists c_{i,j}(\alpha_i, \alpha_j)$), where $\alpha_i$ and $\alpha_j$ are cooperating agents such that (a subset of) $\iota(\alpha_i)$ is provided by (a subset of) $\omega(\alpha_j)$ or *viceversa*.

## 3 Modeling Data Fusion as an Ecosystem

The JDL extended model is a five layers architecture specifically designed to deal with heterogeneous information provided by multiple sources. In the following, the JDL model is used as a framework to discuss the data fusion capabilities distributed throughout the proposed architecture.

### 3.1 Level 0: Sub-object Assessment

Level 0 involves acquiring, filtering and processing raw data in order to extract feature based information. In our work, Level 0 capabilities are achieved through a tight coupling among device and cognitive agents. Usually, a suitable niche can be designed according to the goals of Level 1, i.e., arranging incoming data to fit predefined formats. Thus, raw data coming from $\{\alpha^d\}$ are usually preprocessed and then sent to the proper cognitive agent for feature extraction.

Consider an user tracking system based on camera images. Level 0 could involve the fulfillment of the goal $g_{bb}$, to obtain bounding boxes from image data. This task can be achieved by instantiating the following niche, or agent, $\alpha_{bbe} = A_{bbe} = \{\alpha^d_{cd}, \alpha^c_{be}\}$, where *bbe* stands for "bounding boxes extractor", $\alpha^d_{cd}$ is a camera device and $\alpha^c_{be}$ is an agent implementing a bounding box extraction algorithm.

### 3.2 Level 1: Object Assessment

The main goal of Level 1 is to arrange data acquired by Level 0 in models relating information to symbols, i.e., the symbol grounding [Harnad, 1990]. Still unsolved in its general formulation, the symbol grounding problem arises whenever a symbolic component is added to an artificial system.

Numerical data fusion and data association techniques do not deserve further investigation here. It suffices to note that, with respect to the introduced tracking problem, at this level we could add to the Level 0 niche a suited collection of agents (e.g., a Kalman Filter agent $\alpha^c_{kf}$ and a motion model agent $\alpha^c_{mm}$) to provide, e.g., the estimated position of the target.

Our architecture deals with symbolic knowledge representation and data fusion by introducing a new agent $\alpha^c_{kb}$ managing a knowledge base represented by a Description Logic (DL). DLs consist of a *Terminology Box* (TBox), representing concepts, descriptions and relationships (roles), and an *Assertional Box* (ABox), describing the actual scenario using the ontology described by the TBox. In our case, we focus on the problem of active monitoring of aging and disabled people: in particular, we want to design a pervasive system supervising their activities and reacting to emergency situations whenever they arise.

$\alpha^c_{kb}$ allows symbolic data fusion by using two *layers*: a representation of the Smart Space and its related *information space*, i.e. "objects" and devices of the physical world, along with their associated data, and the data fusion structure, responsible for the creation of meaningful concept instances from base symbols corresponding to sensor data. In order to update the ABox in real time, we assume the availability of a comprehensive niche of agents providing $\alpha^c_{kb}$ with heterogeneous data. Describing in detail all the concepts introduced in the TBox is impossible within the scope of this paper. Therefore, we discuss only the concepts used in later paragraphs.

**Modeling the Smart Space**

The TBox maintains a topological representation of the environment, which is divided in places (using the `Place` concept, e.g. `KitchenPlace`), and then in areas. Further specifications of `Area`, e.g. `ToiletteArea` or `StoveArea`, are defined.

`Entity` serves as a base concept for `Object`, `User` and `Robot`. Each `Area` contains `Object`s, which are classified according to their type (e.g., `Shelf`, `Table`, etc.). Each `Object` is characterized by a `Position` within the environment (metric or symbolic, e.g., a given `Area`) and other information according to the `Object` itself: e.g., a `Shelf` keeps track of the contained objects using the role `contains`. A `Device` is an `Object` with a current `State`, whose instances are updated regularly by device agents. Devices are classified in `SensorDevice`, characterized by an *influence area*, (i.e., their scope in the environment modeled as a collection of areas: in $\mathcal{AL}$ syntax, $\forall$`scope.Area`), and in `ActuatorDevice`. For each physical device type (e.g., cameras, windows, doors, etc.), a corresponding *child* of `Device` is introduced, e.g. `CameraDevice` or `WindowDevice`.

User and Robot model users monitored by the system and (mobile) robots interacting with the Smart Space. Each Entity can perform Actions: modeled Actions include UserActions (e.g., ToBeSomewhere or ReadSomething), RobotActions (e.g., GoToPlace or AvoidObstacle) as well as ObjectActions, like FallDown, etc.

In our architecture, each device is managed by a specific agent. An Agent is defined using an $\mathcal{AL}$ definition corresponding to the one introduced in Section 2, using roles linking to the concepts Capability, Goal, and Data. Next, a DeviceAgent is modeled as a child of Agent with an additional role specifying its controlled Device. For each Device concept (e.g., CameraDevice), a corresponding DeviceAgent is introduced (e.g., CameraDeviceAgent), characterized by some Data. As an example, consider a bounding box feature, as provided by $\alpha_{bbe}$ (see Section 3.1): it is modeled as a child concept of Data, characterized by specific information related to the feature itself through the use of roles.

In this layer, the ABox contains instances of the concepts Device, Entity, Area, Data, etc. Sensory data are mapped to instances of Data, thus updating specific roles of Device instances. Therefore, they are not really given a semantic meaning. For these reasons, this layer does not suffer from the symbol grounding problem, because association between sensor data and symbols is designed *a priori*.

**Symbolic Data Fusion**

Symbolic data fusion is achieved through subsumption, which is the main reasoning scheme provided by DLs. Given two concepts, namely $\mathcal{C}_1$ and $\mathcal{C}_2$, we say that $\mathcal{C}_1$ is subsumed by $\mathcal{C}_2$ (and we write $\mathcal{C}_1 \sqsubseteq \mathcal{C}_2$) if $\mathcal{C}_2$ is more general than or equivalent to $\mathcal{C}_1$. Specifically, subsumption operates on concept descriptions. In the following, given a concept $\mathcal{C}$, we will denote its description $\mathcal{D}$ by defining an operator $\delta$ such that $\mathcal{D} = \delta\mathcal{C}$, and its instances $\mathcal{I}$ by an operator $\xi$ such that $\mathcal{I} = \xi\mathcal{C}$.

Consider again the problem of user position tracking. Suppose we are only interested in determining the areas visited by the user. At our disposal we have three sensors: one camera and two PIR (Passive Infra Red) detectors. A niche is arranged as follows. Three device agents, $\alpha^d_{cam}$, $\alpha^d_{p1}$ and $\alpha^d_{p2}$, are present. $\alpha^d_{cam}$ provides a cognitive agent $\alpha^c_{bbe}$ with raw images. $\alpha^c_{bbe}$ extracts bounding boxes from the images which are then passed to another agent $\alpha^c_{blob}$, able to extract color blobs from the bounding boxes. These data are used to associate a specific bounding box with an user, whose dress colors are supposed to be known. The details are not really relevant: in AmI applications, it could be possible as well to use intelligent wearable technology and RFID tags to perform this association. The symbol grounding problem which should arise is therefore simplified by the redundancy of the sensor network. On the contrary, $\alpha^d_{p1}$ and $\alpha^d_{p2}$ are not able to infer user identity, but they can provide boolean information about the presence of someone in a specific area (according to the sensor range and position).

In the ABox, cameraDev is a CameraDevice, while pirDev1 and pirDev2 are instances of PIRDevice. The niche of cognitive agents composed by $\alpha^c_{bbe}$ and $\alpha^c_{blob}$ is abstracted by functionality in getUserIdAgent, controlling cameraDev. getUserIdAgent provides data about user identity, i.e., instances of the CameraData concept. pirDev1 and pirDev2 are managed by pirAgent1 and pirAgent2, respectively. Each User is characterized by a role id specifying its identity, such that $\forall$id.CameraData, and by a Position i.e., a collection of areas.

The data fusion process is managed by checking the subsumption between the scope role of each Device (see Algorithm 1). The CameraData concept is used to identify the instance of the User whose position is to be computed. Moreover, the user position is initialized to the description of the cameraDev.scope role. Assuming that the camera is able to observe three areas, namely area1, area2 and area3 in the ABox, the description d is inizialited to area1 $\sqcap$ area2 $\sqcap$ area3. Then, for each PIRDevice such that something was detected, the role scope is checked. Again, assuming that dp1 $=$ $\delta$pirDev1.scope $=$ area2 $\sqcap$ area3 and dp2 $=$ $\delta$pirDev2.scope $=$ area2, we have d $\sqsubseteq$ dp1 $\sqsubseteq$ dp2. As a consequence, dp2 $=$ area2 is the new user position.

**Algorithm 1** Compute User Area

**Require:** id $= \xi$CameraData; p1, p2 $= \xi$PIRData
**Ensure:** user area
**for all** u such that u $= \xi$User **do**
  **if** id $\sqsubseteq \delta$u.id **then**
    d $= \delta$cameraDev.scope
    **for all** p such that p $= \xi$PIRDevice **do**
      **if** d $\sqsubseteq \delta$p.scope **then**
        d $= \delta$p.scope
      **end if**
    **end for**
  **end if**
  $\delta$u.pos $=$ d
**end for**

In other words, symbolic data fusion processes are carryed out by adding cognitive agents cooperating with $\alpha^c_{kb}$, which implement the required algorithms. For example, Algorithm 1 is implemented by a cognitive agent $\alpha^c_{cua}$. Moreover, because for each cognitive agent there is a symbolic counterpart within the knowledge base, each data fusion process can be thought of as an *epistemic* operator $\mathcal{K}$ operating on concepts described by the input and output data of the corresponding $\alpha^c$ when a particular configuration of sensor data is updated.

### 3.3 Level 2: Situation Assessment

In this Level relationships among Level 1 objects are established, on the basis of the ontology and the scenario defined in Level 1. As previously pointed out, we are interested in tracking (sequence of) situations in order to detect and to react to arising anomalies. Therefore, we provide the TBox with another layer modeling the situations we want to monitor. Because we are using a DL-based language, we implicitly obtain a hierarchical representation.

**Modeling the Situations**

In this work, situations are considered predicative structures stating facts about the current state of the environment. Therefore, the Situation concept is intruduced to establish a mapping between an Entity and some Action or event. The concept is detailed using intermediate base concepts (e.g., BeingSomewhere or ReadingSomething), related to the childs of Action (e.g., ToBeSomewhere or ReadSomething) and to other concepts, depending on their nature: e.g., BeingSomewhere is connected to Place through the role inPlace and to Area through the role inArea. While Situations are aimed at capturing the current events occurring within the environment, Actions represent actual instances of actions fired by an Entity: Situations are then *meta*representations of Actions. The Situation concept can be further specialized: e.g., BeingSomewhere subsumes child concepts like BeingNearToilette, BeingInBed, BeingNearStove etc., characterized by role inArea restricted to be, respectively, ToiletteArea, BedArea, StoveArea etc.

As long as new Data become available at the symbolic level, new instances of Situation are created within the ABox. Despite its simplicity, for each intermediate Situation branch, this tree-like layer proves to be effective in simulation experiments. It is characterized by a number of advantages compared to other more complex models: (i) it is easily manageable and extensible: new Situation branches can be added by creating new concepts, given that the system is able to distinguish among different situations through (a combination of) sensor data, i.e., through an opportunely defined niche of agents; (ii) its creation can be automated: actual work is focused on creating decision trees from data sets obtained in simulation whose nodes are concepts conveniently defined; (iii) using the subsumption, a system exploiting the tree for managing an *active* monitoring system can be easily implemented.

Consider the following example, explaining how the data fusion process can be carried out at this Level. Assume that we are tracking the activities of $\texttt{u} = \xi\texttt{User}$ inside the kitchen, and that the system inferred that u is inside KitchenPlace, using data coming from a camera. Therefore, $\texttt{a1} = \xi\texttt{ToBeInsideTheKitchen} \sqsubseteq \texttt{ToBeSomewhere}$ is instantiated. This fires the creation of $\texttt{s1} = \xi\texttt{BeingInsideTheKitchen} \sqsubseteq \texttt{BeingSomewhere}$. Along the Situation tree, s1 could be further detailed with the knowledge of the actual Area inside which u is located. The different possibilities are known in advance since each Place is divided in Areas. To achieve this goal, the system could query the status of the the PIR sensors located all over the kitchen, using Algorithm 1, implemented by a cognitive agent $\alpha^c_{cua}$. Assume again that StoveArea is the output produced by $\alpha^c_{cua}$. In other words, once the Place is known, an operator $\mathcal{K}_{cua}$ can be fired, producing new Data symbols, and thus ultimately instantiating a new $\texttt{s2} = \xi\texttt{BeingNearStove}$: because $\texttt{s2} \sqsubseteq \texttt{s1}$, s1 is replaced by s2 as the actual situation for this branch of the Situation tree.

Basically, for each epistemic operator $\mathcal{K}$ relative to an user u it is possible to derive a new description to be added to its situation s. Obviously, this description (which we define as $\delta\mathcal{K}$) is operator dependent. This implies that, for each branching concept of our Situation tree, we must implement an epistemic operator $\mathcal{K}$ providing the necessary $\delta\mathcal{K}$ to further detail the classification, i.e., a corresponding $\alpha^c_{\mathcal{K}}$ is to be istantiated, conveying the required information to perform data fusion.

**Algorithm 2** Classify Situations

**Require:** $\mathcal{K}$
**Ensure:** classification or alarm

```
s = ξSituation
if K is fired then
    δs = δs ⊓ δK
    if δs ⊑⊥ then
        unexpected classification: fire an alarm
    end if
end if
```

The overall process can be modeled at the *meta*level using a new epistemic operator $\mathcal{K}_c$, where $c$ stands for "classification", implementing Algoritm 2. If $\delta\texttt{s}$ is inconsistent, an alarm can be fired. In this way, whenever a Situation has child concepts, the system can fire the proper epistemic operator to produce the description able to push further the classification.

For each Entity, the complete current state can thus be inferred by simply considering the superimposition of the most recent instances of the Situation concept relating the Entity itself to each intermediate base concept introduced so far. Composing Situations originated from different tree branches, it is possible to build more complex representations of the environment. For example, assume the availability of another epistemic operator, $\mathcal{K}_{ss}$, corresponding to a niche of agents communicating information about the detection of smoke by a specific sensor located over the stove. This operator provides the system with a description $\delta\mathcal{K}_{ss}$ producing, e.g., $\texttt{s3} = \xi\texttt{SmokeNearStove}$. The combined Situation $\texttt{s} = \delta\texttt{BeingNearStove} \sqcap \delta\texttt{SmokeNearStove}$ can thus be inferred, suggesting that maybe u is cooking.

**Detecting Sequences of Situations**

As an example of how to use the symbolic data fusion mechanism presented so far, we focus in this Section on the problem of detecting classes or sequences of situations. This task can be accomplished by adding new concepts to the knowledge base and new corresponding operators. Sequences of situations are modeled using the SitSeq concept, which is a collection of situations managed by a role madeOf such that $\forall\texttt{madeOf.Situation}$. A particular SitSeq is MonitoredSitSeq, specifying the interesting Situation to be monitored through the role interesting.

When $\mathcal{K}_c$ is fired, the corresponding Situation instance is added to the madeOf role. In order to detect sequences, several approaches are equally legitimate. For example, the frequency at which situations occur is an important indicator of periodic user behaviors. Consider the problem of monitoring user health status by checking its visits to the toilette. This can be achieved by instantiating a new concept, ToiletteSeq, characterized by $\forall\texttt{interesting.NearToilette}$. Each time the madeOf role is updated, its definition is compared to $\delta\texttt{interesting}$.

By applying a new operator $\mathcal{K}_a$ implementing an alarm condition (based, e.g., on the frequency of the occurrences of the interesting `Situation` in $\delta$`madeOf`), it is thus possible to fire specific alarm behaviors. For example, if the user is living in a private apartment within an assisted-living facility, a remote operator in the nursery can be notified about the user problems.

### 3.4 Level 3: Impact Assessment

The Level 3 of the JDL model is responsible for predicting the effects of the system behavior over the entities (users, robots or devices) modeled by the system itself. Each `Device` is augmented by a description of its behavior modeled as a state machine. The purpose of this representation is twofold: (i) modeling each device as a *fluent*, i.e., an entity whose state changes in time, in order to reason at the temporal level; (ii) determining expected state values given the actual state and inputs, thus being able to detect device faults and react accordingly. This design choice can be extended to involve a planning system, embedded in the representation and data fusion system, able to predict the sequence of actions (and, above all, their estimated effects on the environment and users) necessary for the fulfillment of a given goal. The system can be implemented in practice by defining a new operator $\mathcal{K}_p$ to perform the planning process. Our approach is very similar to the one described in [Mastrogiovanni *et al.*, 2004], thus not deserving further details.

### 3.5 Level 4: Process Refinement

Level 4 deals with topics ranging from learning to input refinement or inter level information flow management. As previously discussed in Section 3.3, learning is the focus of actual work, so it is not longer discussed here. Input refinement is managed by our architecture as follows. A simple mechanism, introduced in Section 3.4, is used to infer possible device faults. If it is the case, or if a particular device is needed within a particular area (e.g. a gas sensor to detect gas leaks), the device network can be physically reconfigured through the use of mobile robots. In our perspective, mobile robots are *mobile* extensions to the Smart Space. They are provided with local device or cognitive agents communicating with other distributed niches. Given the high level of system integration, cognitive agents running on fixed workstations can cooperate with device agents on board of mobile robots. As a consequence, the operator $\mathcal{K}_p$ can be used to instantiate a new `Problem` whose corresponding solution will be used to move the robot toward the specified `Area`, thus implementing the reconfiguration.

Inter level information flow management deals with techniques able to guide data acquisition. In our architecture, this can be accomplished by a new operator, $\mathcal{K}_{ac}$, performing an *active* classification procedure over the `Situation` concepts. Recall the `Situation` hierarchy introduced in Section 3.3. For each branching level, we assume the availability of a corresponding epistemic operator $\mathcal{K}$ able to provide a description $\delta\mathcal{K}$ useful for the classification. Whenever a `Situation` concept has a non null set of child concepts, and no information is available from the corresponding operator, the system can decide by purpose to query the specified cognitive agent or to instantiate an alternate method to obtain the same information, i.e., by requiring a network reconfiguration through the use of mobile robots.

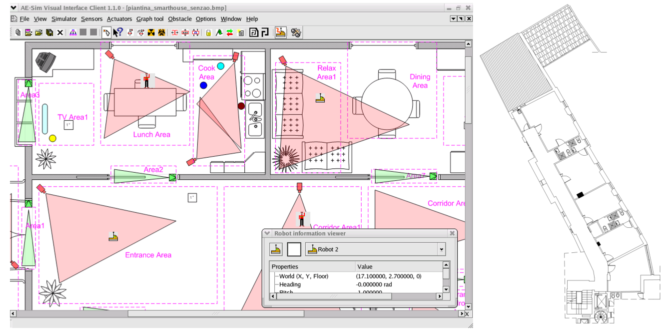



Figure 1: (Left) The simulation environment; (Right) Map of an on going experimental set-up.

## 4 Implementation and Experimental Results

In our actual implementation, $\{\alpha^d\}$ agents exploit the Echelon LonWorks Fieldbus for distributed control and programming. $\{\alpha^c\}$ agents are implemented using ETHNOS, a multi agent framework developed for distributed Robotics and AmI applications [Piaggio *et al.*, 2000]. The knowledge base is developed embedding in ETHNOS Classic, a Description Logic which guarantees sound and complete subsumption inferences. Planning capabilities required by Level 3 are achieved using a PDDL compatible planner.

The cognitive agents of our framework have been tested in a simulation environment built using the architecture itself [Capezio *et al.*, 2006]. It consists of a niche composed by agents implementing simulated devices and user behaviors, visual interfaces (see Fig.1 on the left), etc. Simulated experimental results are carried out by adding specific agents implementing instances of patterns of user activities, and providing sensors with data sets recorded through real sensors (particular data sets have been developed to simulate fire, gas leaks, etc.).

**Algorithm 3** Base Pattern

**Require:** $A = \{a_1, ..., a_n\}$; $E = \{e_1, ..., e_m\}$; $E_{pdf}$
**Ensure:** A specific user behavior
**for all** $i$ such that $i = 1, ..., n$ **do**
  perform action $a_i$
  choose $j$ according to $E_{pdf}$
  fire the event $e_j$
**end for**

Each base pattern (see Algorithm 3) is a sequence of parameterized actions, e.g., movements, interactions with the environment, etc. Examples of base patterns are $A_{answer-phone-call}$ = {answer, talk, hang-up} or $A_{lunch}$ = {goto-kitchen, goto-stove, cook, wait(1), goto-table, eat}. It is worth noting that not all the actions are treated as discrete events: e.g., user movements (i.e., goto-*someplace*) correspond to trajectories in the Cartesian space. Moreover, during each iteration in Algorithm 3, an event $e_j$ is chosen to possibly introduce a perturbation in the current action. Events can range from waiting a certain amount of time to interacting with appliances, from receiving phone calls to completely changing the current pattern with a new one. Events are selected according to a non-uniform outcome probability distribution $E_{pdf}$.

In all the experiments, $\alpha^c_{cua}$ and $\alpha^c_{bbe}$ are able to track the user with cameras, PIRs, lasers and other sensors, maintaining also multiple hypotheses through subsumption. Another agent, $\alpha^c_{fa}$, is able to fire alarms whenever the user remains in the `BedArea` for more than a specified time, i.e., the corresponding situation `BeingInBed` lasts for too long. Moreover, the `ToiletteSeq` sequence of situations is detected multiple times, thus firing proper alarms.

Specific experiments are aimed at testing the active classification system. For example, during the execution of the pattern $A_{lunch}$, after the user has moved to the `StoveArea`, the event `AnswerPhoneCall` is selected. This implies that the current pattern is replaced by $A_{answer-phone-calls}$. After some time, $\alpha^c_{kb}$ is not updated with the expected information. Thus, a specific query to $\alpha^c_{ss}$ is made. If the smoke sensor is responding, the system reminds the user about his previous cooking action. On the contrary, if no smoke is detected, it is inferred that the user was doing something else (the user could be asked about it, being questions posed by the system a suitable way to obtain information). After some time, if a `Cook` action is performed, the epistemic operator $\mathcal{K}_{ss}$ operates on the knowledge base in order to infer that the new user situation is `Cooking`; if not, a new classification is made or an alarm is fired.

The system is actually being tested in a simplified scenario comprising a couple of rooms (a bedroom and a kitchen) which has been fournished in our department. In particular, the system using incoming data from cameras, PIRs, smoke sensors etc. to infer the actions performed inside the rooms. Moreover, the overall system is going to be tested at Istituto Figlie di N.S. della Misericordia, Savona, Italy, an assisted-living facility for elderly and disabled (see Fig.1 on the right).

## 5 Conclusion

In this paper we presented a hybrid knowledge representation and data fusion system developed for integrated AmI applications. Despite its simple architecture, it is able to manage heterogeneous information at different levels, "closing the loop" between sensors and actuators. Based on the concept of an ecosystem of artificial entities, the system architecture has been thoroughly tested in simulation, and part of it has been tested in many real applications. The system is compliant with all the Levels introduced by the JDL extended model, thus being able to process numerical as well as symbolic data. While actual work is focused on expanding the data fusion capabilities through learning, i.e., adding new branches to the `Situation` based structure, future work will involve an in depth investigation about the integration between all the subsystems in a real, complex, experimental set-up.

## References


[Augusto and Nugent, 2005] J. C. Augusto and C. D. Nugent. A new architecture for smart homes based on ADB and temporal reasoning. In *Proc. of 3rd Int. Conf. on Smart Homes and Health Telematics (ICOST)*, Canada, July 2005.

[Barger *et al.*, 2005] T. S. Barger, D. E. Brown, and M. Alwan. Health-status monitoring through analysis of behavioral patterns. *IEEE Trans. on Systems, Man, and Cybernetics – Part A*, 35(1), January 2005.

[Broxvall *et al.*, 2006] M. Broxvall, M. Gritti, A. Saffiotti, B.S. Seo, and Y.J. Cho. PEIS ecology: Integrating robots into smart environments. In *Proc. of the 2006 Int. Conf. on Rob. and Autom. (ICRA)*, Orlando, FL, May 2006.

[Capezio *et al.*, 2006] F. Capezio, D. Femia, F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria. AE-Sim 2.0: a distributed simulator for intelligent spaces and robots. In *Proc. of the Fourth Int. Conf. on Smart Homes and Health Telematics (ICOST06)*, Northern Ireland, UK, June 2006.

[Chen *et al.*, 2003] Shaoua Chen, Hong Bao, Xianyun Zeng, and Yimin Yang. A fire detecting method based on multi-sensor data fusion. In *Proc. of the 2003 Int. Conf. on System, Man and Cyb. (SMC)*, Washington, October 2003.

[Dhaher and MacKesy, 2004] A. H. G. Al Dhaher and D. MacKesy. Multi-sensor data fusion architecture. In *Proc. of the 3rd IEEE Int. Work. On Haptic, Audio and Visual Environments and their Applications (HAVE)*, Ottawa, Canada, October 2004.

[Hagras *et al.*, 2003] H. Hagras, V. Callaghan, M. Colley, and C. Graham. A hierarchical fuzzy-genetic multi-agent architecture for intelligent buildings online learning, adaptation and control. *Information Sciences – Informatics and Computer Science*, 150(1–2), March 2003.

[Harnad, 1990] S. Harnad. The symbol grounding problem. *Physica D*, 42:335 – 346, 1990.

[Kong *et al.*, 2005] Fan Tian Kong, You-Ping Chen, Jing-Ming Xie, and Zu-De Zhou. Distributed temperature control system based on multi-sensor data fusion. In *Proc. of the 2005 Int. Conf. on Machine Learning and Cybernetics*, China, August 2005.

[Mastrogiovanni *et al.*, 2004] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria. A system for hierarchical planning in service mobile robotics. In *Proc. of the Eighth Conf. of Int. Auton. Systems (IAS-8)*, The Netherlands, March 2004.

[Odum, 1959] E. P. Odum. *Fundamentals of Ecology*. W. B. Sanders, United States, 1959.

[Piaggio *et al.*, 2000] M. Piaggio, A. Sgorbissa, and R. Zaccaria. Pre-emptive versus non pre-emptive real time scheduling in intelligent mobile robotics. *Journal of Exp. and Theoretical Artificial Intelligence*, 12(2), 2000.

[White, 1998] F. E. White. Managing data fusion systems in joint and coalition warfare. In *Proc. of the 1998 Int. Conf. On Data Fusion (EuroFusion98)*, Great Malvern, UK, October 1998.

# Context-awareness and user feedback for an adaptive reminding system

**Nadine Richard and Seiji Yamada**
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo-to 101-8430, Japan
nadine@nii.ac.jp, seiji@nii.ac.jp

## Abstract

Busy people would benefit from an adaptive tool for reminding them what they have to do. We are currently developing an adaptive, emotional and expressive agent, which learns when and how to notify users about self-assigned tasks and events. In this paper, we focus on two issues for our learning system: the selection of the input data set, and the design of an appropriate mechanism for getting user feedback without being too intrusive. We describe in particular the inputs that encapsulate the current context of the agent: relative temporal distances, an history of reminders and categories, and the context of both the user and the device on which the agent is running.

## 1 Introduction

When designing an interactive learning system, two major issues are:

- selecting an appropriate set of input values; in the case of an ambient system, this should include information about the current context of the application
- choosing a feedback mechanism, to let the user reward or punish adaptive components after they respectively lead to correct or incorrect actions

We are developing an adaptive, emotional and expressive assistant for reminding tasks and events, called TAMACOACH. Our interface agent adapts to the organizational skills and preferences of a user, by learning **when and how** to present notifications, instead of requiring the user to set explicit alarms. Because users do not interact very often with their calendar and todo managers, the design of the feedback mechanism is particularly important. At the same time, the reminding system should be aware of the context before notifying the user.

### 1.1 Reminders and User Feedback

The content of a TAMACOACH reminder (*e.g.* starting date or location) is extracted from ICALENDAR files manipulated through the user's favourite calendaring application. The form of a reminder (*e.g.* pop-up window or e-mail) depends mostly on the current status of the user and on the capabilities of the device on which TAMACOACH is running (desktop computer, laptop or PDA). A notification is accompanied by a set of user replies (*e.g.* accept or postpone), which are necessary to get explicit feedback about the usefulness of the system.

### 1.2 Necessary Input Data

TAMACOACH learns from both explicit and implicit interactions with its user, respectively through the user interface and through the successive modifications of the calendar data (*via* an external calendar application). On the one hand, the triggering of a reminder depends on:

- the relative temporal distance to the event starting date or to the task due date
- various attributes that help to classify user habits and preferences, *e.g.* contact information, priority or categories

We can extract or abstract those necessary inputs from the data stored in ICALENDAR files. On the other hand, the form of the reminder mostly depends on the current context:

- the user status: availability, on-going activity, physical location, and mood
- the execution environment of the reminding system

We gather information about the user status through the agent GUI, and information about the agent execution context through the operating system of the device.

### 1.3 Organization of the Paper

In order to situate the purpose and the specificities of our reminding system, we present some related work about personal and expressive assistants in the following section. In particular, we explain why such an agent should be both emotional and expressive. Then, in section 3, we outline the learning system of TAMACOACH, and introduce the mechanism that gets feedback from the user. In section 4, we describe what we believe should constitute the context-aware inputs of an adaptive reminding system: relative temporal distances, historical data about categories and previous reminders, and both user and device context. Finally, we conclude and present future work.

## 2 An Adaptive, Emotional and Expressive Reminding System

Recent research projects on personal assistants mainly focus on recommenders and virtual secretaries. Such cognitive assistants learn and reason about tasks, user behaviour and their own behaviour, in order to justify their actions, answer questions and give advice. In some cases, those assistants can be social agents, which create long-term relationships with their users. Nevertheless, we believe that a non-cognitive agent, with limited learning capabilities and basic expressivity, should be sufficient for building a simple, adaptive reminding system, while providing a less-intrusive feedback.

### 2.1 Personal Time Managers

The management of personal and professional calendars and todo lists is difficult: it requires remembering on-going activities, and continuously rearranging priorities. As pointed out by P. Berry *et al.* [Berry *et al.*, 2006], human time management has an intensely personal nature. People are usually reluctant to delegate this specific task to others, and all the more to a software. They also have different preferences and practices regarding how they schedule their time. Moreover, people tend to use various media to keep track of things they intend to do, and often do not record all the tasks and events they should remember of [Bellotti *et al.*, 2004].

Todo managers are used to collect, maintain and organize lists of self-assigned tasks, with different natures, durations, regularities and frequencies. Electronic calendars and todo managers are different but closely related applications, often part of PIM (Personal Information Management) software suites. Within such tools, users can set alarms for the most important events, such as meetings or deadlines.

Many desktop or PDA applications, like SUNBIRD or EVOLUTION, propose traditional ways of organizing todo-lists and calendars, but do not take into account the multiple sources of task-oriented information. TASKVISTA [Bellotti *et al.*, 2004] is a light-weight task list manager, which reduces the cognitive overload of information and events coming from different sources (*e.g.* e-mail, todo manager and Web browsing). Developed by the same research team, TASKMASTER [Bellotti *et al.*, 2003] enables users to keep track of threads of activity and discussions, manage deadlines and reminders, and markup tasks directly within a single e-mail application.

However, those applications do not adjust to the specific organizational habits of their users. On the other hand, cognitive assistants like PTIME [Berry *et al.*, 2005] learn about user scheduling habits and preferences, mostly in order to autonomously negociate meetings.

### 2.2 Expressive Assistants

As advocated in [Bickmore and Mauer, 2006], the use of personal, social assistants on PDAs should be further investigated, as users could access the agent at anytime and may easily consider a *portable assistant* to be part of their life. The authors conducted experiments with various interfaces (text-only, static image and animated virtual agent) for relational agents on PDAs. Animations include facial displays of emotion, head nods, eye gaze movements, and postures shifts. The results of the experiments show that users tend to create relationships more easily with animated agents.

For more information about emotion-based architectures for autonomous agents and human-machine interfaces, recent reviews can be found in [Sarmento, 2004] and [Spinola de Freitas *et al.*, 2005].

### 2.3 TamaCoach

Our goal is not to help users in maintaining a todo-list, but to learn when and how to remind users about what they have explicitly planned to do. However, such an application is a complement to a more complex assistant like PTIME, TASKVISTA or TASKMASTER.

To be as least distractive as possible, TAMACOACH mostly makes use of visual communication, through facial expressions and limited text-based dialogs. Inspired by the famous TAMAGOCHI game, we represent the agent as a virtual creature, which simulates and demonstrates basic emotions. We are investigating the effectiveness of both simulated and expressed emotions regarding three separate goals:

- **Improving agent learning:** a simple emotional system might be an alternative way of computing reward values from user actions.
- **Informing the user:** the expressed emotions should reflect the satisfaction level of the agent, in order to facilitate the interaction with the user and accelerate the learning process.
- **Influencing the user:** the expressed emotions should increase the involvement of the user and thus incite him/her in taking appropriate actions.

In particular, we believe that displaying emotions is a natural and efficient way to inform the user about the estimated satisfaction level of the agent, *i.e.* the global self-evaluation of its usefulness. Moreover, associating an expressive virtual creature to our reminding system might motivate the user in honouring various self-assigned commitments and in better organizing his/her personal schedule.

A basic reminding system does not require an elaborate relational agent to make the user feel comfortable. Creating a strong relationship with the agent could even be counterproductive, as it would distract the user. As with a TAMAGOCHI, we hope that inducing amusement or guilt will lead the user in interacting with the reminding system, and thus make it useful.

In the following sections of this paper, we will mainly focus on the learning module and the user feedback system, and leave aside the emotional system.

## 3 Learning About our Bad Habits

### 3.1 The Learning Module

Our learning system, presented in figure 1, is composed of two cascaded XCS-based modules [Wilson, 1995]. The XCS model has proved to be an efficient, evolutionary, rule-based learning mechanism. Like other Learning Classifier Systems, it has been extensively used for data-mining and to control animats, but not to design interface agents.

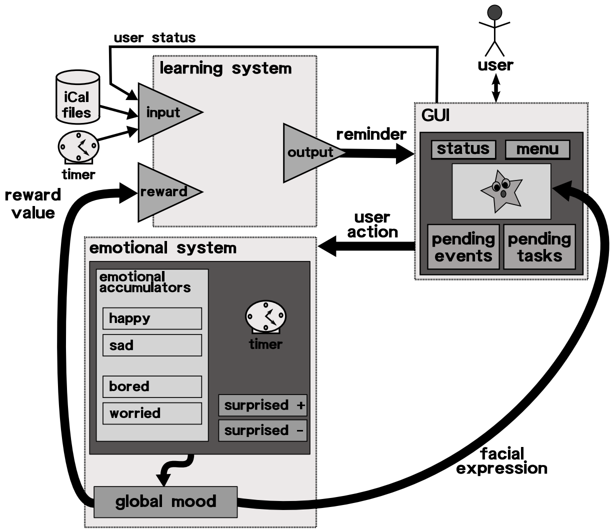



Figure 1: Overall system with emotional feedback and display.

The learning module interacts with an emotional system, inspired from L. Sarmento's architecture [Sarmento, 2004]. The agent emotional state is a set of emotional accumulators, associated with specific evaluation functions, appraisal thresholds and decay values. The global mood is calculated from the values of emotional variables. It is used both as a reward value for the learning system, and to display the emotional state through the graphical representation of the agent.

### 3.2 Direct User Feedback

A reminder can be presented in different but combinable forms, depending on the context: a pop-up dialog box, an e-mail message, a mobile e-mail message, or a sound alarm. In any case, the new reminder is appended to the appropriate list of pending events or pending tasks. A notification will stay in the list until the user answers back, by e-mail or through the GUI.

Together with a reminder, a set of possible replies is presented to the user. In particular, within the GUI, selecting a pending item in a list makes a pop-up window appear, to let the user select one reply among 6 possibilities. This light-weight, explicit feedback system enables TAMACOACH to evaluate the quality of its actions, without annoying busy users too much. The main 3 possible replies are as follow:

- *accept*: the user thinks that the reminder is useful and will act immediately (*e.g.* finish the task or attend the meeting)
- *later*: the reminder has to be presented again later
- *ignore*: the user does not care about the reminder

In the case of a *later* or *ignore* reply, the user can specify respectively whether the reminder was too early or too late. This information is important to avoid taking a neutral feedback as a punishment: the user may have found the reminder useful but would like to be notified again anyway (*e.g.* if the user cannot perform the task immediately), or the user informs the agent about his/her lack of interest in this particular task or event.

All the above replies concern the reminder itself: they give direct feedback on the usefulness of TAMACOACH actions. Nevertheless, we are dealing with ICALENDAR files that are modified by an external calendaring application. We thus added 3 complementary replies, in order to get further information about the actual status of the reminded item:

- *done*: the user has already attended the event or achieved the task
- *postpone*: the starting date of the event or the due date of the task should be changed
- *cancel*: the item should be cancelled (*e.g.* the user has missed the event or given up the task)

The items for which the user has replied *accept*, *ignore*, *done* or *cancel* are considered as *acknowledged*, and will not be processed by the learning system anymore. A *later* or *postpone* reply will trigger a new reminder later.

The rewards associated with user replies are values between -1.0 (higher punishment) and +1.0 (higher payoff); the values of the rewards have been intuitively chosen and will be tuned during the evaluation phase of the learning system.

### 3.3 Remote and Delayed User Feedback

The user can interact with a running instance of TAMACOACH *via* e-mails, from a computer, a PDA or a mobile phone. For now, this kind of remote interaction is possible only when replying to an e-mail reminder (giving one of the 6 possible answers). However, remote control of the agent could be extended to other operations available through the GUI.

The user may not reply immediately to a reminder for various reasons: for instance, he/she forgot to update the busy status before leaving the office, or did not check his/her mobile e-mails. Delays in the user reply will lead to delay the feedback on the triggering rules. It means that the new fitness of those rules will not be computed when the next items will be processed. However, a delayed reply should not be considered as a negative feedback, as we cannot know why the user did not react immediately. We thus just consider that, in case of a delay, the next reminders will not be selected by an updated set of rules.

## 4 Context-Awareness

In order to trigger a reminder at the right time or choose the best form of a notification, the adaptive reminding system should be aware of the context, regarding time, the user and the host device. The user status is obtained from the user through the TAMACOACH GUI, while information about the execution environment can be acquired from the operating system. Some other significant contextual values are computed from ICALENDAR data and past experiences, *e.g.* the temporal distance to a task deadline or the average duration of an item category.

## 4.1 Time-Aware

The difficulty of calculating temporal distances in our application comes from the heterogeneity of the dates and times involved, as well as the various meanings of the urgency of a reminder. In order to present appropriate relative temporal distances to our learning system, we propose to compute discrete, multi-granular values of both brute and contextual distances.

**Temporal Distance and Granularity**

The *relative temporal distance* is the distance between the current date and the starting date of an event or the due date of a task. To be relevant for the learning mechanism of our agent, this distance is approximated and translated into a symbolic value expressing how close the current date is to the deadline or starting date (*e.g.* far, very close or already late). With such a decomposition, a calendar item is easily categorized as happening soon or in a long time.

The temporal information to be manipulated by TAMACOACH is by nature multi-granular [Dyreson *et al.*, 1998; Combi *et al.*, 2004]. Keeping track of the granularity (time scale) is necessary, in order to consider relative symbolic values like *very close* or *far* in an appropriate time context: an event that starts in five minutes is obviously close to happen, but a task that might take 8 weeks to be achieved and is only 5% completed can also be considered as *close* in a *weeks* time scale if the current date is less than 9 weeks before the deadline.

For clarity and efficiency purposes, we chose a set of significant thresholds in order to categorize both the granularity and the approximate value of a distance, instead of proposing functions for computing this categorization. The following table presents the distance values for each granularity. *VC*, *C*, *QC*, *QF*, *F* and *VF* respectively stand for *very close*, *close*, *quite close*, *quite far*, *far* and *very far*. Granularities are represented by durations, which are expressed using different time scales. The *VST*, *ST*, *MT* and *LT* abbreviations respectively stand for *very short-term*, *short-term*, *mid-term* and *long-term*.

| | **Time scale** | **VC** | **C** | **QC** | **QF** | **F** | **VF** |
|---|---|---|---|---|---|---|---|
| **VST** | 0-24 hours | < 0.16 | < 1 | < 4 | < 8 | < 15 | < 24 |
| **ST** | 1-10 days | < 2 | < 3 | < 4 | < 5 | < 7 | < 10 |
| **MT** | 10-42 days | < 12 | < 14 | < 18 | < 24 | < 32 | < 42 |
| **LT** | > 6 weeks | < 7 | < 8 | < 10 | < 13 | < 18 | $\geq$ 18 |

The user can also be *late* for an event or for achieving a task. This additional distance value can be used for each granularity, in order to express the severity of the delay since the starting date or the due date occurred (less than a day to more than 6 weeks).

**Brute and Contextual Temporal Distance**

The brute calculation of a multi-granular distance might be insufficient to evaluate the degree of urgency for notifying the user about an event or a task. Some tasks cannot be achieved if not at home or at the office, and some events should be reminded when happening during a business trip or vacations. Depending on the context of the user, the reminding system should take other temporal items into account, in order to compute a contextual distance.

For example, the user is currently working but will leave soon for two weeks of vacation and an important half-day meeting is planned on the first day of his/her return: it could be useful to remind the user now, because the relative distance to this high-priority event is quite short when the duration of absence is deducted from the time left for preparing the meeting. The calculation of the contextual distance therefore requires two kinds of temporal information:

- well-known user habits: usual working days and hours, list of public holidays, *etc.*
- periods allocated for opaque events, *i.e.* for events during which the user is considered as busy [1]: business trips, vacations, meetings, *etc.*

The first kind of information can be specified by the user when launching the application for the first time, and can be considered as fixed (even if it might be updated later through the TAMACOACH interface).

The second kind of temporal data can be deduced from the ICALENDAR files, but requires further information to be useful for learning: the system needs to know basically if the period is related to work or not. Such information is inferred from user-defined categories associated with events and tasks: when a new category is encountered (while reloading an ICALENDAR file), TAMACOACH asks whether the category is work-related.

We do not take the physical location of the user into account to compute the contextual temporal distance. However, this could be an interesting extension, which would adjust the triggering of a reminder to the time required to get to the location of the planned event.

**Learning From the Past**

When learning a generalized classification of calendar items, the most useful data are the user-defined categories assigned to an event or task. In order to learn about the user habits concerning a kind of activity, we propose to use additional attribute related to the categories of past events and tasks:

- *category duration* : average observed duration for a given category (very short, short, medium, long)
- *category delay* : average observed delay for a given category (very short, short, medium, long)
- *category is early?* : is the user usually early when achieving this kind of task? [2]

On the other hand, notifications can be repeated only if the user explicitly asked for being reminded later. As a frequent repetition of the same reminder content may annoy the user, information about previous reminders enables the system to learn when to present again the same reminder. We thus also store the following historical values:

- *reminders* : number of reminders already triggered for this item (0, 1, 2, 3, $<$6, $<$10, $\geq$15)

[1] The transparency/opacity flag is set for ICALENDAR items through the external calendar application.

[2] In this case, the category delay represents how long before the deadline is this kind of task usually achieved.

- *last reminder* : temporal distance since the last reminder for this item (very close, close, far, very far)

The distance to the previous reminder, and the average category duration and delay have the same value domain as the granularity of temporal distances.

### 4.2 User-Aware

The user status is the agregation of the following information:

- *activity level* ("busy state"): available, busy, very busy, away, disconnected
- *activity context*: work, leasure, vacation, commuting, sick, conference, meeting, *etc.*
- *location*: office, transportation, home, business trip, *etc.*
- *mood*: very good, good, average, bad, very bad

The user status is gathered directly from the user through the GUI for now. Nevertheless, it would be less intrusive to detect the user activity load automatically, through monitoring (keyboard/mouse activity, estimated posture, *etc.*).

However, one indication we can compute from ICALENDAR data concerning the user activity level is the current *task load*. This average value is computed from the number and the priority of current tasks. It completes the contextual temporal distance, which only takes into account opaque events to occur between the current date and the starting/due date of a considered event/task.

### 4.3 Device-Aware

The execution context of TAMACOACH is composed of the following attributes:

- *location* : desktop, laptop, pda, *etc.*
- *has network?* : does the device have an active network connection?
- *has sound?* : can the device play a sound?

Such information is required to adapt the selected actions: for instance, if the device has no network connection, the reminding system cannot send an e-mail.

## 5 Conclusion and Future Work

In this position paper, we have presented our adaptive, emotional and expressive reminding system. We have explained how to get useful but light-weight feedback from the user. We have also introduced the necessary input data, extracted from ICALENDAR files; in particular, we have described various context-aware inputs, including relative temporal distances and user- and device-related information.

We are currently developing a first prototype, incrementally tested on three different platforms: a desktop PC under GNU/Linux, a laptop under MS-Windows, and a Zaurus C-3000 under a native distribution of GNU/Linux. In parallel, we are investigating how to simulate various user profiles for evaluating and tuning the learning system. Then, we will integrate the described emotional system, to improve learning and display facial expressions. Afterwards, we will conduct experiments on human users, in order to validate our hypothesis about the possible influence of an adaptive, emotional time manager on different categories of users.


## Acknowledgments

This research project is founded by the Japan Society for the Promotion of Science (JSPS).


## References


[Bellotti *et al.*, 2003] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI)*, Fort Lauderdale, USA, April 2003.

[Bellotti *et al.*, 2004] Victoria Bellotti, Brinda Dalal, Nathaniel Good, Peter Flynn, Daniel G. Bobrow, and Nicolas Ducheneaut. What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI)*, Vienna, Austria, April 2004.

[Berry *et al.*, 2005] Pauline M. Berry, Melinda T. Gervasio, Tomás E. Uribe, and Neil Yorke-Smith. Mixed-Initiative Issues for a Personalized Time Management Assistant. In *Proceedings of the Workshop on Mixed-Initiative Planning And Scheduling*, Monterey, California, USA, June 2005.

[Berry *et al.*, 2006] Pauline Berry, Ken Conley, Melinda Gervasio, Bart Peintner, Tomás Uribe, and Neil Yorke-Smith. Deploying a Personalized Time Management Assistant. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, Hakodate, Japan, May 2006.

[Bickmore and Mauer, 2006] Timothy Bickmore and Daniel Mauer. Modalities for Building Relationships with Handheld Computer Agents. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Montreal, Canada, April 2006.

[Combi *et al.*, 2004] Carlo Combi, Massimo Franceschet, and Adriano Peron. Representing and reasoning about temporal granularities. *Journal of Logic and Computation*, 14(1):51–77, February 2004.

[Dyreson *et al.*, 1998] Curtis E. Dyreson, William S. Evans, Hong Lin, and Richard T. Snodgrass. Efficiently supporting temporal granularities. Technical report, TimeCenter, September 1998.

[Sarmento, 2004] Luís M. Sarmento. *An Emotion-Based Agent Architecture*. PhD thesis, Faculdade de Ciências da Universidade do Porto, March 2004.

[Spinola de Freitas *et al.*, 2005] Jackeline Spinola de Freitas, Ricardo R. Gudwin, and João Queiroz. Emotion in Artificial Intelligence and Artificial Life Research: Facing Problems. Technical report, 2005.

[Wilson, 1995] Stewart W. Wilson. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

# Ambient Intelligence for Care of the Elderly in Their Homes

**Fariba Sadri**
Imperial College London
180, Queens Gate, London SW7 2AZ, UK
fs@doc.ic.ac.uk

## Abstract

In this paper we explore the application of an agent based model to designing ambient intelligence for use by the elderly in their own homes. The agent model used is based on abductive logic programming. It utilises personalised knowledge, interacts with the user and other agents and devices, and, while autonomous, incorporates user preferences when deciding whether or not to act, and what action to take.

## 1 Introduction

The elderly population is growing in many areas of the world [UN, 2002]. Thanks to better diets, better hygiene and advances in care and medicine more people are able to live relatively independent lives well into their later years. Many elderly people would like to live in their own homes and enjoy independence for as long as possible.

Old age, however, brings with it a range of impairments, amongst these cognitive impairments of varying degrees [Dobbs and Rule, 1990; Light, 1990; Ratner *et al*., 1987]. The quality of life and safety of the elderly can be significantly enhanced if they could have help and advice easily and unobtrusively available to them in their own homes day and night, whenever required. Ambient intelligence (AMI) seems an ideal way of providing such help and advice.

Our vision of ambient intelligence in this application is a collection of ubiquitous services (with sensors and activators) embedded in the home and agent-based computer systems that collate and synthesise information from them and other sources, including knowledge about the users, such as their profiles, routines, preferences, activities and medical needs. The agent systems would monitor the needs and activities of the elderly users and would alert, advise, answer questions, and, where necessary, for example if a risky situation is determined, would activate other resources or contact external bodies.

For such systems to be useful they would have to be acceptable by their users, and even enjoyable to use. To this end they would have to be personalised, not just with respect to specific information about the users' circumstances, but also with respect to their preferences regarding when and how to interact with the users.

In this paper we bring together a number of technologies, all based on abductive logic programming to lay the foundations of this vision of ambient intelligence for the elderly, with particular attention to the needs of the elderly with some memory impairment. We describe three modules:

- The Diary/Appointments Manager (DM): that records diary items and appointments, advises on clashes, proposes alternative arrangements, reminds and advises.
- The Medicine Manager (MM): that records information on the medical needs, such as the frequency and dosage of medications, reminds, advises about combining food and drink with medication, and alerts if medication advice is not adhered to.
- The Situation Recogniser (SR): that merges information received from sensors and draws upon the profile of the user to detect if an unusual or risky situation has arisen and to determine if and what action is needed.

In the next subsection we give examples to motivate how we would like these three modules to function.

### 1.1 Motivation Examples

#### 1.1.1 The Diary/Appointments Manager (DM)

Suppose Alice has an optician appointment on Wednesday at 10 am. For this she needs to take with her both her bifocal and reading glasses. Alice would also like to meet with a friend, Jane, at their favourite café that day at 11. Having consulted a map of the locality and checked transport information, DM will advise Alice that she will not make a meeting with Jane at 11, and suggests 12, instead.

If Alice has indicated that she wishes to be reminded about the optician appointment, then on Wednesday, or the day or week before, according to Alice's wishes, DM will remind her and will also advise that she should take both her spectacles with her.

#### 1.1.2 The Medicine Manager (MM)

Alice is required to take a particular medication, call it Xeron, every 2 hours, but never on an empty stomach. She also needs to take iron tablets twice a day, and should avoid drinking tea or coffee for 2 hours after taking the tablet. She can actively interact with MM and ask, for example:

- When do I take my Xeron tablet next?
- Am I supposed to take any medication now?
- Is it ok to drink coffee now?

Moreover, MM can actively remind Alice, for example:

- Alice, it is time for your Xeron tablets. You should take 2, but first eat something.

- Alice, please do not drink coffee now, it will interfere with your iron tablet. Wait for another half hour. How about some fruit juice or water, now, instead?

#### 1.1.3 The Situation Recogniser (SR)

SR merges information from several sensors around Alice's house. It is after 10pm, the front door is unlocked. Depending on what Alice has requested in her profile, SR can do two different things. It can activate the lock on the front door and lock it, or it can let Alice know that the front door is still unlocked, and maybe turn the light on the hallway for Alice to go and lock the door herself.

It is now 10am and Alice is still not up. She is usually up by 8:30. SR can consult DM and MM to see if it can form possible reasons for Alice not being up. Was she out late last night? Did she take medication last thing at night that induces longer sleep? Does MM advise that Alice is depressed and prone to sleeping longer? Putting all the information together SR will decide to take action or not, and what form of action. For example it could try and wake Alice up, and if this does not work to call a friend or a doctor.

### 1.2 Approach

We will model the three modules using an agent-based approach in abductive logic programming. Each of DM, MM and SR can be considered an agent that has its own knowledge base, reactive rules, and interaction policies, incorporating the user's preferences. The three modules can interact with one another, and they can draw upon information from other sources. The reasoning of these agents will be based on an implemented abductive proof procedure called CIFF [Endriss *et al*, 2004]. This proof procedure combines backward and forward reasoning, abduction and constraint satisfaction. This last feature is needed because the knowledge bases of the agents can have temporal features and CIFF was designed, in particular, to deal with temporal constraints. CIFF has been used in the planning and reactivity modules of a logic-based agent model called KGP [Kakas *et al*, 2004]. Our abductive agent based approach is inspired by the KGP and the agent model of [Kowalski and Sadri, 1999].

## 2 Background

In this section we provide brief summaries of the background technologies used in the rest of the paper.

### 2.1 An Abductive Logic Based Agent

Such an agent has, possibly multiple modules of, knowledge formalised as abductive logic programs. It may have its own goals, it can observe its environment and record its observations, it has awareness of other agents in its environment and it can interact with these agents and with its environment. It can pro-actively plan for its goals, as it adjusts and reacts to changes in its environment. Such an agent has been discussed in [Kakas *et al*, 2004; Kowalski and Sadri, 1999].

### 2.2 Abductive Logic Programs (ALPs)

An abductive logic program is a tuple <*P, IC, A*> where:
*P* is a logic program, consisting of rules of the form *Head if Body*, where *Head* is an atom and *Body* is a conjunction of literals. All variables in *Head* and *Body* are assumed universally quantified over the whole clause.
*IC* is a set of integrity constraints of the form $L_1, \ldots, L_n \rightarrow H_1, \ldots, H_m$, where the $L_i$ are literals, the $H_i$ are atoms, possibly *false*, and the "," on both sides of → denotes "*and*".[1] All variables occurring only in the $H_i$ are assumed existentially quantified on the right side of →, and all others are assumed universally quantified over the integrity constraint.
*A* is the set of abducible predicates. In our application the abducibles will be actions. Similar to the KGP model of agents, in our application we can have 3 types of actions, physical, sensing and communicative. Given an ALP and a query *Q* (conjunction of literals signifying goals or observations), the purpose of abduction is to find a set of atoms in the abducible predicates that, in conjunction with *P*, *entails Q* while *satisfying* the *IC*, wrt. some notion of *satisfaction* and *entailment*.

### 2.3 Reactive Rules and Interaction Policies

The integrity constraints in ALPs can be used to model reactive rules and interaction policies - see, for example [Sadri *et al*, 2002; Sadri and Toni, 2000]. The following are examples, the first a reactive rule, and the second an interaction policy:
*unlocked-door(T), T≥22:00 → sound-alarm(T)*,
i.e. if the door in unlocked after 22:00 sound the alarm.
*tell(X, Y, request(R), T), have(R,T) → tell(Y, X, agree-to-give(R), T+5)*, i.e. if Y receives a request for a resource R from X, and Y has R, then Y must answer within 5 time units with an agreement to give R.

### 2.4 Information Integration

Abductive logic programming has been used for information integration [Xanthakos 2003]. We do not have space to address this in any detail in this paper. We just note that in the DM and SR modules where information from various sources are integrated, the ALP technology provides for a smooth incorporation of this integration within the modules.

### 2.5 CIFF

CIFF [Endriss et al, 2004] is a proof procedure for abductive logic programs with constraints. It incorporates a number of re-write rules, such as unfolding, propagation, case analysis and logical simplification as well as reasoning with constraint solving.
Example: Consider the following ALP:
*P: have(R,T) if buy(R,T1), T1<T*
*have(R,T) if borrow-library(R,T1), T1<T*
*IC: buy(X,T), ¬ money(T) → false*
*borrow-library(X,T) → T>9:00, T<17:00*

[1] We differ slightly here from standard definitions of IC in ALP, simply to make the syntax of our examples more succinct.

*A: borrow-library, buy, money*

The ALP states that if one buys some R or borrows it from the library one will have R. But one cannot buy anything without money, and borrowing from the library is possible only between the hours of 9 and 17.

Consider the query:

*have(magic-book, 20:00), ¬money(T)*,

namely we want the *magic-book* at time *20* but have no money at any time. CIFF will compute one answer:

*borrow-library(magic-book,T1),T1>9:00,T1<17:00.*

## 2.6 ALPs with Preferences

ALPs have been extended to incorporate reasoning with preferences – see, for example [Prakken and Sartor, 1997]. We illustrate this informally via a simple example:

g(T): warm(T) → garden(T)

w(T): weekday(T) → work(T)

prefer(w(T), g(T)) if deadline(T).

The two integrity constraints, named, respectively, *g(T)* and *w(T)*, state that if it is warm, garden, and if it is a week-day, work. The *prefer* rule says that the second of these 2 should be preferred if there is a deadline.

We now describe each of the three modules, DM, MM and SR, in turn. Each module is an agent. For each one of them we describe its ALP incorporating its knowledge, reactive rules and interaction policies.

We will not address the user interface in this paper. As shown in Fig. 1, for each module we assume that there is a User Interface and a User Interface Translator, that allow interaction between the module and the user.

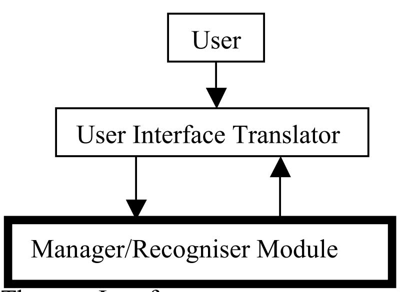



Fig 1. The user Interface

# 3 The Diary/Appointments Manager (DM)

DM manages the diary of the user and advises and reminds, according to user's preferences, based on information it integrates from the diary and other sources such as map and transport sources. DM's function is summarised in Fig. 2.

The DM's ALP *<P, IC, A>* is constructed as follows:

*P* consists of diary items and auxiliary definitions, possibly integrating map, transport and diary information. Each diary item can be viewed as an 8-tuple *<ID, Date, Time, Place, Purpose, Person, What-to-take, Reminder-tag>*, where *ID* is an internal unique identifier of the item and *Reminder-tag* models user's preference for whether and when to be reminded about the appointment. The other items of the 8-tuple have their expected meanings. To facilitate the writing and comprehension of the integrity constraints we represent each diary item by means of 3 facts of the form:

*d1(ID, Date, Time, Place)*

*d2(ID, Purpose, Person)*

*d3(ID, What-to-take, Reminder-tag)*.

*A* consists of the predicate *tell*.

*IC* consists of 3 sets of integrity constraints:

Check Handlers, Query Handlers, Reminder Handlers.

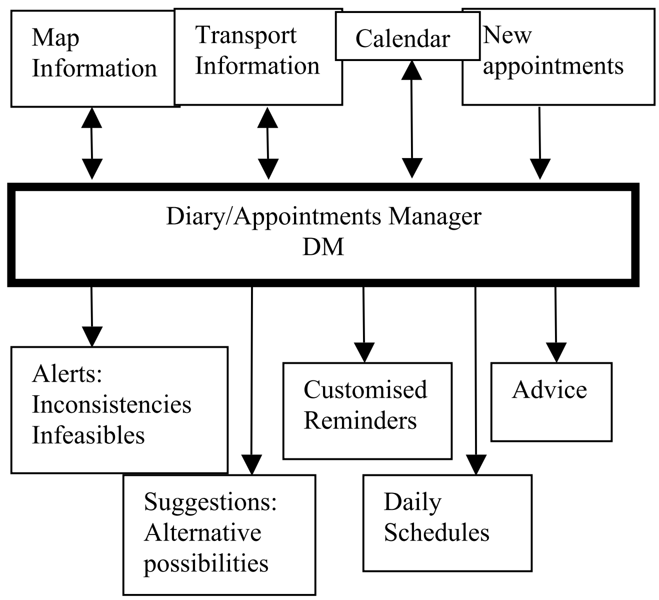



Fig 2. The Diary/Appointments Manager

The Check Handlers:

These check diary items and advise whether or not there are clashes (I1), or insufficient times allowed between any two appointments (I2) and, if possible advise of alternative arrangements (I3).

I1: *d1(ID1, Date, Time, Place1), d1(ID2, Date, Time, Place2), ID1≠ID2→ tell(clash(Date, Time, Place1, Place2))*

I2: *d1(ID1, Date, Time1, Place1), d1(ID2, Date, Time2, Place2), ID1≠ID2, Time1<Time2, estimate-time(Place1, Place2, T), Time2-Time1<T → tell(not-enough-time(Date, Time1, Time2, Place1, Place2))*

I3: *d1(ID1, Date, Time1, Place1), d1(ID2, Date, Time2, Place2), tell(not-enough-time(Date, Time1, Time2, Place1, Place2)), estimate-time(Place1, Place2, T), Time1+T=T3, clear-diary(Date, T3) → tell(alternative(Date, Time2, Place2, T3))*

*estimate-time* and *clear-diary* are auxiliary predicates that would be defined in P, in the former case, integrating information from the map and transport sources.

The Query Handlers:

Using our approach we can easily formalise reactive rules and interaction policies to deal with many types of queries, for example:

- What appointments do I have today, tomorrow, on 12 Sep, etc?
- What appointments do I have between 9 and 12 tomorrow?
- When is my next dentist appointment?
- What time should I leave for my doctor's appointment on Wednesday?
- How do I get from the doctor's to the supermarket?
- Do I have to take all my medicine bottles with me to the doctor's next time?

As examples, we give the IC items for handling the first three types of queries from the list above. We assume that through the User Interface Translator the queries input to DM will be of the form, respectively, *tell(enquire-schedule(Date)), tell(enquire-diary-item(Date, T1, T2)), tell(enquire-next-appointment(Purpose))*.

*tell(enquire-schedule(Date)), d1(ID, Date, Time, Place), d2(ID, Purpose, Person) → tell(inform(Date, Time, Place, Purpose, Person))*

*tell(enquire-diary-item(Date, T1, T2)), d1(ID, Date, Time, Place), d2(ID, Purpose, Person), T1≤Time, Time≤T2 → tell(inform(Date, Time, Place, Purpose, Person))*

*tell(enquire-next-appointment(Purpose)), current-date(D), d1(ID, Date, Time, Place), d2(ID, Purpose, Person), D≤Date → tell(inform(Date, Time, Place, Purpose)).*

The Reminder Handlers:
Via the interface, for each diary item, users can input their preference for whether and when they would like to be reminded about the item. This is recorded in the *Reminder-tag* of the item.

Many different alternatives are possible and can be dealt with. For example:

- If *Reminder-tag*=∅ denoting that the user has expressed no preference then, by default, a reminder can be issued on the day of the appointment:
  *current-date(D), d1(ID, D, Time, Place), d2(ID, Purpose, Person), d3(ID, What-to-take, ∅) → tell(remind(D, Time, Place, Purpose, Person, What-to-take))*
- If *Reminder-tag* is one of *on-the-day*, *the-day-before*, *the-week-before*, etc, then integrity constraints similar to the above can deal with it. For example:
  *current-date(D), d1(ID, Date, Time, Place), d2(ID, Purpose, Person), d3(ID, What-to-take, the-week-before), Date=D+7 → tell(remind(Date, Time, Place, Purpose, Person, What-to-take))*
- If *Reminder-tag=only-when-necessary* then a reminder can be issued if
  - DM knows of an appointment on the day, and
  - The time to prepare to leave home in order to reach the appointment on time (according to DM's references to the map and transport information) is approaching, and
  - DM, through information from the sensor merges, cannot detect any movement in the home that would indicate the user is preparing to leave.

  For example:
  *current-date(D), current-time(T), d1(ID, D, Time, Place), d2(ID, Purpose, Person), d3(ID, What-to-take, only-when-necessary), estimate-time(home, Place, T1), T+T1>Time-1, not preparing-to-go(T) → tell(remind(D, Time, Place, Purpose, Person, What-to-take))*, where *preparing-to-go* has to be determined by sensor information.

## 4 The Medicine Manager (MM)

MM manages information about the user's medication and reminds and advises. Similarly to DM, its reminder is based on the preferences expressed by the user. It is summarised in Fig. 3.

The MM's ALP <*P, IC, A*> is constructed as follows:
*P* consists of

- Facts of the form *med(ID, Frequency, Dosage, Reminder-tag)*, recording information about the user's medication. *Reminder-tag*, as before, models user preferences about reminders, *ID* is a unique identifier for the medicine, and *Frequency* and *Dosage* have their expected meanings.
- Facts of the form *last-intake(ID, Date, Time)*, maintaining information about the last intake of each medicine.
- Facts of the form *restriction(ID, Description, Time-period)*, describing any restrictions associated with a particular medicine identified by ID. For example:
  *restriction(xeron, take-with-food, _)*
  *restriction(iron, no(coffee), 2hr)*.

*A* consists of the predicate *tell*.
*IC* consists of 2 sets of integrity constraints, Query Handlers and Reminder Handlers, which are similar to those in the DM module. An example of the Query Handlers is:
*tell(enquire-next-intake(ID)), med(ID, Frequency, Dosage, _), current-date(D), current-time(T), last-intake(ID, D, Time), Time+Frequency=Next-time→ tell(inform(ID, Next-time, Next-time-T))*.
This will deal with questions of the form "when do I take my Xeron tablet(s) next?" The answer, through an appropriate interface will be, for example, " you should take your Xeron tablet(s) next at 4 o'clock, i.e. 2 hours from now".

As examples of the Reminder Handlers, we present the following two integrity constraints. The first deals with a user preference always to be reminded to take their medicines. The second ensures that if a reminder is issued about taking a medicine, it is followed by a reminder about any associated food or drink restrictions.

*med(ID, Frequency, Dosage, always), last-intake(ID, Date, Time), current-date(Date), current-time(T), T=Time+Frequecy → tell(remind(ID, Dosage))*

*tell(remind(ID, Dosage)), restriction(ID, Description, Time-period)→ tell(remind(ID, Description, Time-period))*

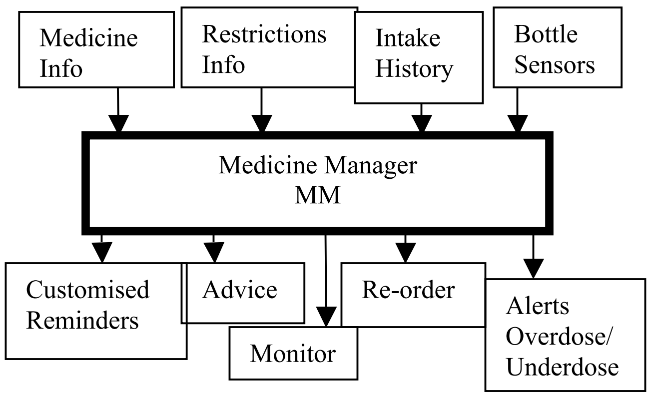



Fig. 3 The Medicine Manager

If medicine bottles are equipped with sensors then MM can also incorporate Monitoring and Re-order functionalities. For the first, information from the sensors can be used to determine if the user is under-dosing or over-dosing. It can also help detect if the user is mixing different types of medication. This information, in turn, can lead to further advice generation, or in serious cases, to alerts issued to the user or others, such as a carer or a doctor.

## 5 The Situation Recogniser (SR)

SR is the third agent-based module. It is summarised in Fig. 4. It merges information from the sensors embedded in the user's home, interacts with the other agent modules DM an MM, integrates all the information available to it, including the user's profile of habits and "normal behaviour" in order to recognise

- if an unusual situation has arisen,
- if it is a risky or dangerous situation,
- if it needs to take action, and
- if so what action.

We divide SR into 2 sub-modules, the Recogniser-Action module (RAC), and the Recogniser-Assessor module (RAS). RAC deals with situations that can be recognised unambiguously and demand clearly what action(s) should be taken. RAS, is much more complicated, and deals with more ambiguous and complex situations. In this paper we give only a brief hint of what is involved in RAS.

### 5.1 The RAC Module

The ALP *<P,IC,A>* of the RAC module is constructed as follows:

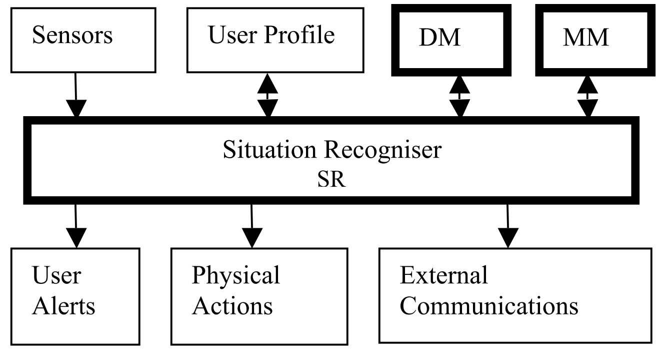



Fig. 4 The Situation Recogniser

*P* consists of a collection of facts fed into RAC via the sensors. The facts are of the form *observed(Property, Time)*. For example *observed(unlocked(front-door), 22)*.
*A* consists of predicates *do* and *tell*.
*IC* consists of integrity constraints and preference rules. The integrity constraints are of the form
*IC(Name): Factors-observed →Actions-and-Messages*
where *Factors-observed* is a conjunction of atoms with the *observed* predicate, *Actions-and-Messages* is a conjunction of atoms with predicates *do* or *tell*, and *IC(Name)* is a unique name for the integrity constraint.

The preference rules have the form
*prefer(IC(Name1), IC(Name2)) if Conditions*,
where the *Conditions* may be missing. For example:

*IC(locking): observed(unlocked(front-door), 22)*
*→do(lock(front-door))*
*IC(alerting): observed(unlocked(front-door), 22)*
*→tell(alert(unlocked(front-door))), do(turn-on-light)*

*prefer(IC(alerting), IC(locking))*.

The two integrity constraints state that if the front door is detected to be unlocked at time *22* then lock it, in the first case, and alert the user that it is unlocked and turn the light on, in the second case. The preference rule states that the second one of these is preferred to the first.
Another example is:
*IC(gas): observed(on(gas), T), observed(¬on-hob, T), observed(¬on-hob, T+5minutes) → do(turn-off(gas))*,

namely that SR should turn the gas off, if it is detected to be on and for 5 minutes later it is detected that nothing is placed on the hob.

Detecting if the front door is left unlocked or if the gas is left on, deciding that the situation requires action and the choice of action, within the bounds of user preferences are relatively straight-forward. But there are situations where these decisions are more difficult, thus the need for RAS.

### 5.2 The RAS Module

To motivate this sub-module we recall the second SR example in subsection 1.1.3. Alice is not up yet, later than her usual waking time. RAS needs to do the following:

- Compare the actual situation S with the expected situation S'. *In the example, S' can be determined from the profile of Alice compiled by monitoring her habits.* For a discussion of ubiquitous user profiles see, for example, [Lorenz and Zimmermann, 2006].
- If S and S' differ, attempt to find an explanation for S. *In the example, the SR agent can send a message to MM to enquire what medication Alice would have taken, say, after 9 pm the night before. It can also ask DM about Alice's activities the day before.*
- If it cannot find an "un-alarming" explanation or if S persists then take some action. *Text a neighbour/friend*.

## 6 Conclusions

In this paper we have proposed an abductive logic programming agent based approach to providing AMI for care of the elderly in their homes. In the application of AMI to elderly care the work of the RoboCare project (see, for example [Bahadori et al, 2000]) is of interest and complementary to ours. They have devised location and tracking systems via stereo vision to monitor the position of robots and humans. Other authors have addressed the application of AMI to healthcare. [Gouaux, et al, 2002], for example use AMI to monitor cardiac risks. In using abductive agens our work is related to that of [Stathis and Toni, 2004]. They use the KGP agent model for providing AMI in a scenario of a traveler with a PDA-like device.

Our paper is a foray into a rich area with many interesting open issues for further investigation. The RAS module, for example, requires exploration and integration of several issues and technologies. Implementation and incorporation of appropriate interfaces are also further areas of work.

## References


[Bahadori et al, 2000] S. Bahadori, A. Cesta, L. Iocchi, G.R. Leone, D. Nardi, F. Pecora, R. Rasconi, L. Scozzafava. *Towards Ambient Intelligence for the Domestic Care of the Elderly*. The RoboCare Technical Reports, RC-TR-0704-3.

[Dobbs and Rule, 1990] A.R. Dobbs and B.G. Rule. *Adult Age Differences in Working memory*. *Psychology and Aging* 4:500-503, 1990.

[Endriss et al, 2004] U. Endriss, P. Mancarella, F. Sadri, G. Terreni, F. Toni. *The CIFF Proof Procedure for Abductive Logic Programming with Constraints.* In *Proceedings of JELIA 2004, International Conference on Logics in AI*, pages 31-43, Lisbon, Portugal, September 2004. Springer Verlag LNAI.

[Gouaux, et al, 2002] F. Gouaux, L. Simon-Chautemps, J. Fayn, S. Adami, M, Arzi, D. Assanelli, M.C. Forlini, C. Malossi, A. Martinez, J. Placide, G.L. Ziliani, P. Rubel. Ambient Intelligence and Pervasive Systems for Monitoring of Citizens at Cardiac Risk: New Solutions from the EPI-MEDICS Project. *Computing in Cardiology* 29:289-292, 2002.

[Kakas et al, 2004] A. Kakas, P. Mancarella, F. Sadri, K. Stathis, F. Toni. The KGP Model of Agency. In Proceedings of ECAI 2004, European Conference on Artificial Intelligence.

[Kowalski and Sadri, 1999] R.A. Kowalski, F. Sadri, *From Logic ProgrammingTtowards Multi-agent Systems*. *Annals of Mathematics and Artificial Intelligence*, 25:391-419, 1999.

[Light, 1990] L.L. Light. *Memory and Language in Old Age*. *In Handbook of the Psychology of Aging*, 3rd Edition, J.E. Birren and K.W. Schaie (Eds.), pages 275-290, San Diego, CA, Academic Press.

[Lorenz and Zimmermann, 2006] A. Lorenz and A. Zimmermann. *Adaptation of Ubiquitous User-Models*. Workshop on Ubiquitous User Modelling, ECAI 2006.

[Mancarella *et al*., 2004] P. Mancarella, F. Sadri, G. Terreni, F. Toni. *Planning Partially for Situated Agents*. In *Proceedings of 5th Workshop on Computational Logic in Multi-Agent Systems (CLIMA V)*, pages 29-30, September 2004, J.Leite and P.Torroni, eds.

[Prakken and Sartor, 1997] H. Prakken and G. Sartor. *Arguments-based Extended Logic Programmimg with Defeasible Priorities*. *Journal of Applied Non-Classicla Logics*, 7(1), 1997.

[Ratner *et al*., 1987] Ratner H. H., Schell D. A., Crimmins A., Mittleman D., Baldinelli L. (1987), Changes in adult prose recall: aging or cognitive demands. *Developmental Psychology*(23), 521-525.

[Sadri and Toni, 2000] F. Sadri, F. Toni. *Abduction with Negation as failure for Active and Reactive Rules*. Selected Papers of AI*IA 99: Advances in Artificial Intelligence, 6th Congress of the Italian Association for Artificial Intelligence, Springer Verlag LNAI 1792, E. Lamma and P. Mello (Eds.), pages 49-60, 2000.

[Sadri *et al*, 2002] F. Sadri, F. Toni, P. Torroni. *Resource Reallocation via Negotiation Through Abductive Logic Programming*. In *Proceedings of JELIA*, 2002, pages 419-431. Springer Verlag LNAI 2424.

[Stathis and Toni, 2004] K. Stathis and F. Toni. *Ambient Intelligence Using KGP Agents*. In *Proceedings of EUSAI 2004*, P. Markopoulos et al. (Eds.), LNCS 3295, pages 351-362, 2004.

[UN, 2002] *UN World Population Prospects (2002 Revision) and Eurostat 2004 Demographic Projections.*

[Xanthakos 2003] I. Xanthakos. *Semantic Integration of Information by Abduction*. PhD thesis, Imperial College London, 2003.

# Efficient Agent Communication in Wireless Enviornment

**Sana Farooq[2], Sana Khalique[2], Aqsa Bajwa[2], Obaid Malik[2], Hiroki Suguri[1], Hafiz Farooq Ahmad[1], Arshad Ali[2]**

[1]Communication Technologies, 2-15-28 Omachi Aoba-ku, Sendai 980-0804 Japan
{suguri, farooq}@comtec.co.jp

[2]NUST Institute of Information Technology, 166-A, Street 9, Chaklala Scheme 3, Rawalpindi, Pakistan
{sana.farooq, sana.khalique, aqsa.bajwa, obaid.malik, arshad.ali}@niit.edu.pk

## Abstract

Advances in communication and technology and the expansion of wireless networks has led an increasingly utilization of handheld devices like PDA's, mobile devices and pocket PC's. Intelligent agents are the technology that is becoming more and more popular in wireless systems. Agents are able to communicate with other entities by applying some intelligence and carry out task on behalf of the users. In wireless environment communication should be reliable and efficient. FIPA ACL messages are currently encoded as human readable strings. Using the binary encoding of FIPA ACL instead of string utilizes wireless link more efficiently. This paper addresses these issues and gives the implementation of FIPA bit-efficient encoding (binary encoding specified by FIPA) It focuses device based context aware architecture of the system and the efficient communication technique in wireless environment.. This work is based on SAGE-Lite, which is a multi-agent system for lightweight devices

## 1 Introduction

Multi-Agent systems are one of the important emerging technologies that provide collaborative environment for a community of social agents for the provision of continuous and dynamic services. Agents and multi-agent systems play a central role in ambient intelligence, Grid computing and ubiquitous computing and many others.

Agents are independent reactive/proactive entities having two main features

***Autonomy:*** they take the task from user and perform task without user intervention.

***Communicative:*** they interact with other agents to perform the task by communicating with other entities.

Agent architecture conceptualizes agents as being made of perception, action, and reasoning components. The perception component feeds the reasoning component, which governs the agents' actions, including what to perceive next. Agent system architecture analyzes agents as interacting service provider/consumer entities. System architectures facilitate agent operations and interactions under environmental constraints, and allow them to take advantage of available services and facilities. Agent infrastructure provides regulations that agents follow to communicate and to understand each other, thereby enabling knowledge sharing. Agent infrastructures mostly deal with the communication among agents based on a communication language using common ontological system. [7]

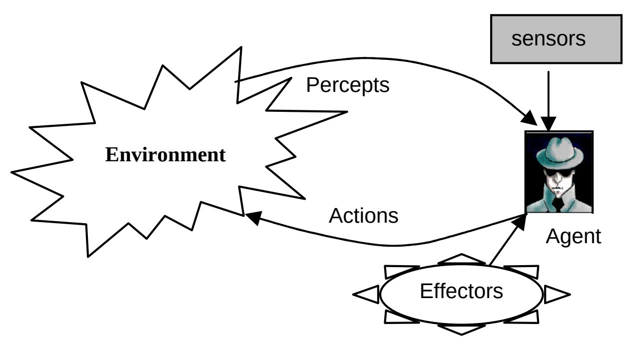



**Figure 1** Agent and its environment

In artificial intelligence, multi agent systems technology is called a new paradigm for conceptualizing, designing and implementing software systems. MAS represent virtual societies where software entities (agents) acting on behalf of their owners or controllers (people or organizations) can meet and interact for various reasons (e.g., exchanging goods, combining services, etc.) and in various ways (e.g., creating virtual organizations etc.)

In order to perform complex jobs agents need to interact with their surroundings, it can be humans, system services or other agents. Agents have communication language which should be declarative, syntactically simple and readable by people. In wireless environment agent communication should be efficient in terms of speed, bandwidth utilization, reliability and security. FIPA-ACL is an outer language that specifies message format and include descriptions of their pragmatics, which is the communicative acts or intentions of the agents. Furthermore, FIPA also define semantic languages to successfully communicate with each other and the bit-efficient protocol for the encoding and decoding of FIPA ACL message, specifically for wireless environment.

SAGE-Lite has the ability to sense the capabilities of the devices in the given context and react accordingly. As the system is based on the latest FIPA specifications, communicating agent uses the FIPA-ACL for communication and bit-efficient protocol for encoding and decoding of ACL message. This paper gives the implementation of FIPA bit-efficient syntax . Software agents communicate with their peers on the same machine and on the remote machine via message transport service, which supports communication through WAP and Bluetooth.

## 2 Related Work

SAGE-Lite is an enhancement of SAGE agent platform, which is designed for the fixed networks. SAGE is an open source FIPA complaint second-generation multi agent system, which is a distributed decentralized architecture. This decentralization feature enables the system to be highly scalable and the objective of fault tolerance is achieved by the indigenous idea of Virtual Agent Cluster, which uses separate communication layers among different machines. The Virtual Agent Cluster works independently regardless of the external environment proceedings, providing a self-healing, proactive abstraction on top of all instances of multi-agent systems. Also the architecture fully supports peer-to-peer communication, which brings scalability, fault tolerance and load balancing among distributed peers as well [2].

## 3 Approach

Wireless devices vary from each other in shape and functionality. Application written for one device do not execute on other devices. Wireless applications developed using Java can be run on different devices from different vendors. J2ME provides a dynamic deployment mechanism that allows applications to be downloaded and installed onto devices over the wireless network Therefore,The development environment chosen for this project is Java 2 Micro Edition (J2ME), in which building blocks can be combined to setup a complete runtime environment that suits a particular type of device.The requirements for the development of SAGE-Lite are:

- can be deployed on devices having at least 1Mb RAM.
- target devices should support J2ME (CLDC 1.0 and MIDP 2.)

## 4 Context Aware Systems

Context is a powerful and longstanding concept in human-computer interaction. The notion of the context is much more widely appreciated today. It refers to the physical and social situation in which computational devices are embedded. One goal of context-aware computing is to acquire and utilize information about the context of a device to provide services that are appropriate to the particular people, place, time, events etc [3]. In order to implement the above mentioned concept of context awareness and an intelligent agent to function effectively in these environments, they need to sense and reason about the current context of the environment and interact smoothly with other agents. [4]

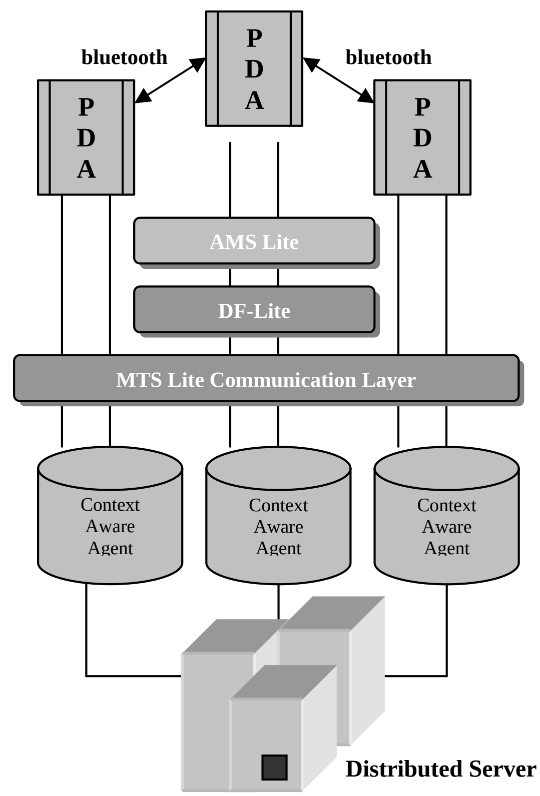



**Figure 2** Surrogate Architecture

Another concept of context-aware system is that the light-weight devices have different features and capabilities. It is the ability to sense the capability of the devices and consider a intelligent action.This concept emphasis on device based context awareness or device capability awareness systems. In order to implement this concept an intelligent agent needs to sense the capabilities of the device e.g. the processing power, memory, GUI availability etc. and then provide appropriate service by interacting with other agents. We have developed a MAS that caters to the above mentioned device based context awareness.

### 4.1 Context Awareness in SAGE-lite

A diversified range of mobile applications are becoming popular day by day. The applications run using the constrained resources provided by today's vast variety of the cellular devices. The resources are constrained in terms of the processing power, memory, graphical user interface support, permanent storage, screen size etc.

SAGE-Lite is an advancement of the SAGE agent platform. We propose surrogate architecture, a set-up in which SAGE and SAGE-Lite works together in combination. SAGE is a static distributed platform consisting of main container providing FIPA interoperability to one or more lightweight platforms, SAGE-Lite. SAGE and SAGE-Lite will communicate using Wireless Application Protocol (WAP). Agents running on different handheld devices will communication with one another via Bluetooth and WAP**.**

Light-weight devices have different features and capabilities. To cater the wide variety of devices, context awareness module has been introduced in architecture. It enhances the capabilities of SAGE and makes it compatible with a vast variety of currently available lightweight devices. The concept presented in this architecture emphasizes on the device based context awareness. A device running a SAGE-Lite agent platform request for a particular service. The request from SAGE-Lite goes to the main container SAGE. Main container has the device capability detection module which by analyzing the context detects the capability of the device to run the particular service. The profile of device along with the unique device address is in return stored into the device information system on the server. Agent providing that service will be sent to the device according to its capabilities and managing information according to the needs of mobile user. For example, the services should be provided to a devices depending upon the capabilities of the devices. If the device does not support GUI, then it should not get the service agent with GUI or if it has lower processing power then the services with minimal features should be sent to the devices. For this purpose the device will first send its full specifications to the main container and then SAGE will send the service to the device according to its specifications.

## 5 System Architecture and Components

There are two main components of system, one is communication language i.e. FIPA ACL and other is the message transport services.

### 5.1 Light-weight Message Transport Service:

Message transport service (MTS) is the backbone of any MAS. It supports the sending and receiving of ACL messages between agents. The agents involved may be local to a single agent platform (AP) or on different APs. SAGE-Lite only provides the Intra platform Communication which involves the communication of two agents residing on the same platform. MTS handles the intra-platform communication by itself, bypassing the extra overhead of Agent Communication Channel (ACC).

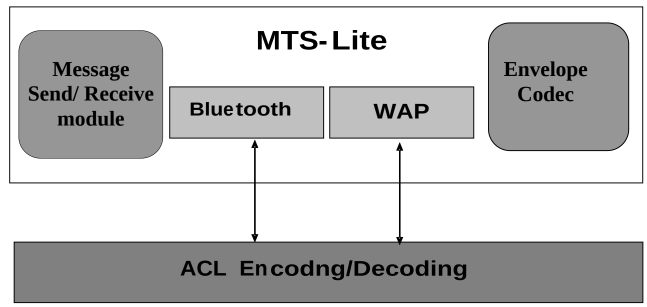



**Figure 3** MTS Light Architecture

**An encoding/decoding service** provides the facility to encode the message content into an encoding representation for use as a transport-message payload. This procedure must also function in reverse for decoding transport messages. To make this communication efficient and effective, we adopted a FIPA-complaint encoding scheme called "Bit Efficient Encoding Scheme" for the representing ACL Messages to be sent over the wireless link. **Figure 3** shows the isolated architecture of MTS-Lite. This module is capable of communicating wirelessly through WAP and Bluetooth. The Message Send/Receive Module is responsible for the **message buffering** and **envelope codec** is responsible for the encoding of the envelope in Bit Efficient encoding representation.

### 5.2 Intra platform Communication Architecture:

Two queues are maintained to minimize the processing power and memory consumed by the message buffering, one for the received messages and the other for the messages to be transmitted.

The WAP and the Bluetooth clients will keep on reading the queue for a message and as soon as they get it, the message is encoded using the ACL Codec and is transported. When MTS-Lite receives the message to be sent, it first of all checks its own DF-Lite to see if the agent resides on the same machine. If it finds the match on the same machine then instead of putting the message in the transmission queue, the message is directly put into the reception queue. However if the message is to be sent to an agent executing on another machine, then the message is put into the transmission queue.

For the message to be sent over to another platform, the envelope is also encoded using the particular codec. [5]

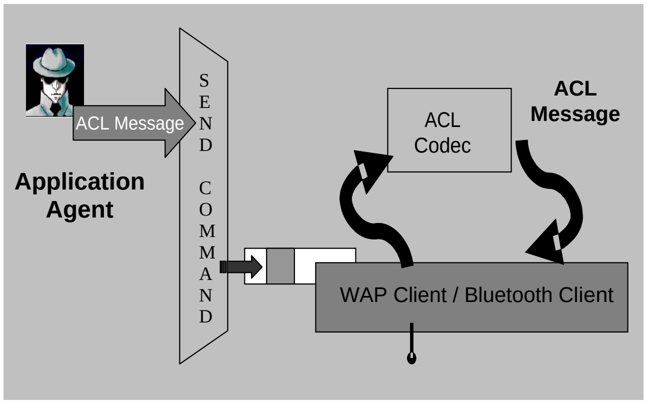



**Figure 4.1** Intra platform transmission

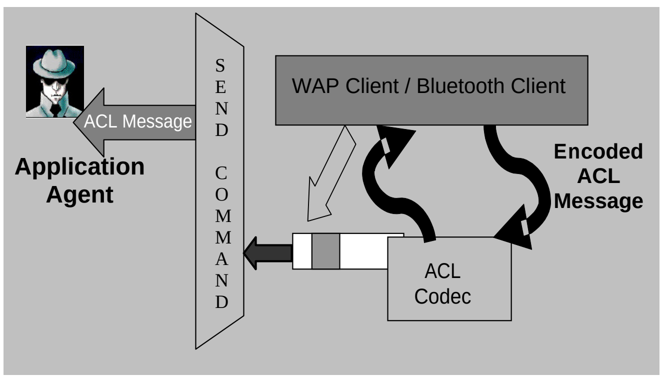



**Figure 4.2** Intra platform reception

## 5.3 Agent Communication Language

We have developed lightweight MAS according to FIPA specifications to achieve high performance and efficiency in agent communication. SAGE-Lite message structure is based on FIPA ACL message structure. One of the objectives is to address the issues of wireless environment and provide the efficient way of communication over a wireless link. Our proposed design of ACL message is given in Figure 5. This is the most complex part of the ACL module because it has to deal with the Content objects created by the programmer and the ontologies created by the programmer and its responsibility is to encode those objects in to a FIPA Compliant Content language. FIPA specifies some content languages e.g SL, XML, RDF etc but we have chosen SL because it is the standard language being used by many FIPA Compliant Multi Agent Systems.

### 5.3.1 Message Encoding Mechanism

Initially the agent has some information, which he wants to transfer to another agent. Once the information is classified, it is stored in to proper corresponding objects called Content Facilitator Objects or CFObjects. The SL Codec converts the CFObjects into proper FIPA complaint SL0 string (content of the ACL message) by referring particular ontology. The content is then passed into the ACL Codec, which further encodes the ACL message using ***Bit-efficient encoding*** scheme and send to the recipient. A function named "encode" takes the ACL message in the arguments and returns an encoded string.

The encoding of ACL message starts by appending the MessageId and the Version Bytes at the beginning of the message as per the standards defined by FIPA

```
byte MessageId = (byte)0xFA, Version  = (byte)0x10;
```

The message performatives, sender, receivers, replyTo, content, language, encoding scheme reference(which in our case is Bit-efficint encoding scheme), ontologies protocol, conversationId, replyWith and inReplyTo fields of the ACL message are then encoded and appended in a single string. Bit-efficient encoding of the "receiver" parameter of the ACL message is programmatically shown below:

```
if (msg.getAllReceivers() != null &&
                              msg.getAllReceivers().size() > 0 )
{   str.append((byte) 0x03);
   str.append('(');
   for (int i = 0; i < msg.getAllReceivers().size(); i++)
   {  str.append(getAgentIdentifier((AgentId)
      msg.getAllReceivers().get(i)));
    if (i!=msg.getAllReceivers().size()-1){str.append('|');}}
```

The formulated string can then be validated with the help of another function which checks the string for nested double quotes and inserts an escape character '\' before every nested double quote and returns a string value.

```
public synchronized static String validateString(String valStr)
   { StringBuffer stb = new StringBuffer(valStr);
   for (int i = 1; i < stb.length() - 1; i++)
      {  if (stb.charAt(i) == '"' && stb.charAt(i - 1) != '\\')
         stb.insert(i, '\\');
      }
   return stb.toString(); }
```

Incase of inter platform communication, an envelop is attached with the ACL message which is also encoded using the bit-efficient encoding scheme. The envelop contains the information like the sender and the receiver name, ACL representation, content length and the encoding scheme used to encode the ACL message. This information at the recipient platform is important to decode the rest of the ACL message.

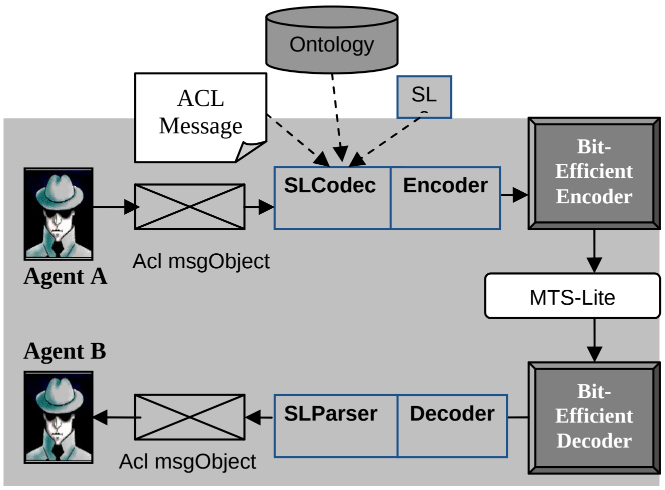



**Figure 5** Encoding/Decoding Mechnism in SAGE-Lite

### 5.3.2 Message Decoding Mechanism

When the agent receives the message, ACL Codec decodes it using bit-efficient decoding mechanism and passes the decoded ACL message to SL Codec which extracts the content from the ACL message in string format. SL Code contains the *Token Manager*, which creates the tokens from the string. Each token is then passed to the parser which validates the FIPA complaint SL0 string and converts the string to corresponding CFObject using the particular ontology which contains the information. Agent gets the same information in CFObjects that the sending Agent intended to send.

Programmatically, a pointer is defined which reads the string and extracts the actual ACL message out of it.

```
int curIndex=0;
```

The code for extracting the recipents from the encoded ACL is below:

```
if (strMsg.indexOf(""+(byte) 0x03,curIndex) > 0)
  {
    String strRec  =
    strMsg.substring(strMsg.indexOf(""+(byte)
  0x03,curIndex),strMsg.indexOf('\n',curIndex));
    strRec     =
  strRec.substring(strRec.indexOf('(')+1,strRec.indexOf(')'));
    int chk=0,index=0;
    while (strRec.indexOf('|',index)>0)
    {  String rec =
strRec.substring(index,strRec.indexOf('|',index));
       aclMsg.addReceiver(getAid(rec));
       int hold=index;
       index = strRec.indexOf('|',index)+1; }
    String rec = strRec.substring(index,strRec.length());
    aclMsg.addReceiver(getAid(rec));
    curIndex = strMsg.indexOf('\n',curIndex)+1; }
```

Similarly, all other parameters are extracted from the encoded string at the receivers end.

## 6 Future work and Conclusion

The architecture described in this paper has been implemented using the FIPA specifications. Future direction of SAGE-Lite is that we have to make it compatible with other FIPA platforms as well.By developing prototypical applications based on the framework, we have shown that it may in fact be utilized to develop context aware services efficiently. Therefore a lot of emphasis has been given not only on the development of an effective system but also on a system that is fully resource efficient. The proposed architecture is enriched with a lot of unique ideas, which have and will pave ways for a lot more research initiatives in times to come.

Future work is also required to ensure that our agents can handle the unreliable nature of the network messages. Efficiency and reliability are the issues that need to be catered in wireless environment and the multi-agent systems. Existing architecture needs to be enhanced in terms of context awareness as in future it will become part of all the multi-agent systems and human life.

SAGE-Lite ACL message that we developed contains the efficient techniques of SL encoding and decoding also the message envelope is encoded and decoded using the bit-efficient encoding scheme, which is best suitable for wireless environment. Using the binary encoding of FIPA ACL instead of string utilizes wireless link more efficiently.

SAGE-Lite framework allows implementing agent-based business applications or e-commerce applications on the resource-constrained devices.

## References

[1] Supporting Nomadic Agent-based Applications in FIPA Agent Architecture - Heikki Helin, Department of Computer science, Series of publication, A report A-2003-2

[2] Arshad Ali, Sarmad Malik, Muazzam Mugal, M. Omair Shafiq, Amina Tariq ,Amna Basharat, NUST Institute of Information Technology (NIIT) National University of Sciences and Technology,(NUST) ,Rawalpindi, Pakistan. “Scalable fault tolerant Agent Grooming Environment - SAGE”

[3] Thomas P. Moran, Paul Daurish, IBM Almaden Research Center. “Introduction to Special Issue on Context Aware Computing”.

[4] Anand Ranganathan, Roy H. Campbell, University of Illionis at Urban-champaign, USA. “A middleware for Context Aware agents in Ubiquitous Computing Environment”.

[5]Aqsa Bajwa, Sana Farooq and Obaid Malik. “Persistent Architecture of Context-Aware Lightweight Multi-agent System”. In Proceedings of the Fourth International Workshop on Programming Multi-Agent systems to be held with Fifth International Conference on Autonomous Agents and Multi-agent Systems, Japan, 2006.

[6] Sana Khalique and Mobeena Jamshed. “Significance of Semantic Language in Multi-agent System”. In Proceedings of the Pacific Rim International Workshop on Multi-Agents. Malaysia, September 2005.

[7] Mujahid ur Rehman and Salman Bashir. “Scalable Message Transport Infrastructure for High Performance Multi-agent Systems.

# Mining Spatial Relationships with Limited Sensory Information

**Chumki Basu, Hui Cheng, Darren Butler**
Sarnoff Corporation
201 Washington Road, CN 5300
Princeton, NJ 08543-5300
{cbasu, hcheng, dbutler}@sarnoff.com

## Abstract

Human navigation in an unknown environment requires an understanding of the spatial relationships of the terrain. For example, a soldier who is on a reconnaissance mission in a new city needs to "know" the spatial layout of the surroundings with high confidence. Oftentimes, this understanding must be acquired within a very short amount of time and with limited sensory inputs. The soldier would benefit from a digital avatar that draws inferences about the spatial layout of the city based on an initial set of observations and guides the soldier either in further exploring the environment or in making decisions based on these inferences. In this paper, we present and evaluate an inductive approach to learning spatial associations using sensory data that is available from the simulation environment of a first-person shooter (FPS) game, *Unreal Tournament*. We study two kinds of spatial relationships between nodes on a level of a game map: nodes that are placed near each other to satisfy some spatial requirement and nodes that are placed near each other to satisfy the design preferences of a level architect. We show that we can infer both kinds of relationships using an association rule mining algorithm. Furthermore, we show how to use an ontology to distinguish between these relationships in order to discover different types of spatial arrangements on a specific map. We discuss how the inferred associations can be used to control an avatar that makes recommendations for navigating unexplored areas on a map. We conclude with some thoughts on the applicability of our methods to real-world scenarios beyond the simulation environment of a game.

## 1 Introduction

Finding and understanding spatial relationships given limited sensory inputs (e.g., limitations of field-of-view, non-optimal lighting, etc.) pose real-world challenges that need to be solved. A game engine such as that provided by *Unreal Tournament* (UT) helps us simulate an environment to test our ideas in understanding spatial relationships on a level of a game map. Ultimately, our goal is to deploy these ideas in the field to achieve broader impacts – e.g., helping a soldier infer the layout of a city by examining just a single city block. Similar to a scenario that describes ambient intelligence, we envision a soldier who through a "digital avatar can filter the avalanche of information directed at him and respond to it in different ways" [Shadbolt, 2003]. For example, the soldier needs to filter visual sensory inputs in order to decide what information can be ignored and what information can be used to support movement-specific decision-making tasks.

The digital avatar (also referred to as computer player or bot) we envision should achieve two goals. First, if the bot is at a specific location, it should have knowledge of other locations in its vicinity given limited sensory inputs from its environment. Secondly, the bot should respond to its environment intelligently based on what it has seen. The analogy we make here is with a person entering a room. By looking at the spatial arrangement of objects in a room and comparing with background knowledge, a person can infer whether the room is a kitchen or a bedroom and can make intelligent decisions about subsequent actions. This requires deriving higher-level meaning from detected spatial patterns.

In this paper, we propose a methodology for addressing these goals and present our initial findings. For the first goal, we employ a data mining algorithm to learn spatial relationships on a UT map. We infer interesting associations between spatially correlated locations using inductive methods to model a level architect's choices. We then propose to derive higher-level meaning from the associations using an ontology of UT objects. We categorize the associations into two groups: those that satisfy some spatial requirements and those that reflect the preferences of the level architect. Both kinds of rules provide predictive information that can be used to guide movement on a level of the UT map.

In the broader context, we believe categorizing association rules is useful in addressing a control problem. When an avatar receives sensory inputs from the environment that conform to background knowledge about a place, processing can be taken offline (e.g., additional spatial relationships can be inferred from knowledge bases such as an ontology). Dynamic knowledge such as the distribution of spatial preferences of a level architect must be learned from observable data, requiring additional sensory inputs from the environment (and resulting in further navigation). Distinguishing between these two kinds of spatial associations is useful feedback to a controller that directs sensory and processing tasks in real-time applications.

In Section 2 of this paper, we will provide some background on association rule mining, discuss its relevance to this problem and evaluation measures. In Section 3, we describe how we used the UT system to define the problem space and discuss the beginnings of a UT ontology that we used for our experiment. We present the results of the experiment in Section 4, discuss the applicability of our methods to the larger problem of inferring a map in Section 5, and conclude in Section 6.

## 2 Spatial Association Rules

Association rule learning, including the widely used algorithm, *Apriori*, has been used to discover patterns of co-occurrences in item sets from transaction data in databases [Agarwal *et al.*, 1993; Agarwal and Srikant, 1994]. A traditional application of association rules has been market-basket analysis: analysis of transactions involving items that are purchased together in order to find frequent item sets. There has also been much work in the application of this algorithm to geographical and spatial data [Koperski and Han, 1995; Malerba *et al*., 2002; Mennis and Liu, 2003; Shekhar and Huang, 2001]. We use the *Apriori* algorithm since it is efficient, handles nominal data, and has been successfully applied in spatial data mining research. Another approach to finding spatial co-locations is spatial statistics; however, in that approach, we would incur the computational expense of finding all possible co-location patterns across a large number of spatial features.

An example of the kind of spatial association rule that we are interested in learning is of the form:

$$lift \rightarrow stairs \tag{1}$$

The above association rule signifies that the antecedent of the rule, "lift", and the consequent of the rule, "stairs", frequently occur together in space or are spatially co-located.

One of the issues in applying association rule learning to spatial data is the creation of "transactions" over space. Shekhar and Huang [2001] describe approaches such as windowing (which has been used, for example, in building recommendation models [Nakagawa and Mobasher, 2003]), the definition of a referential spatial feature or predicate (such as *closeTo*), and an event-centric approach (where events are defined as boolean spatial features) that use neighborhoods in place of transactions.

In our work, transactions are represented as transitions of the states of objects. We record each bot transition (an exploration event) in a UT game session as a neighborhood of spatial features. A spatial feature is defined as a node at a specific location; the neighborhood contains locations "near" the bot's location. We define "nearness" according to visibility and reachability measures using UT coordinates, which will be described in Section 3.

### 2.1 Generalization Hierarchies

In spatial association rule mining, different kinds of hierarchies are applicable. In our work, we rely on the generalization (superclass/subclass) hierarchy found in an object ontology for UT. Koperski and Han [1995] suggest that while it may be difficult to discover strong associations between low-level concepts, many spatial association rules may be expressed at a higher conceptual level. By tapping into an ontology, we find a level of generalization that leads us to discover strong associations. We generalize our spatial features one level up from the leaf level of the UT object ontology (replacing instance names with class names) before running the association rule mining algorithm.

### 2.2 Evaluation

In association rule mining, the "goodness" of the rules is often measured using *support* and *confidence*. For a set of items, *S*, occurring in a rule, support of a rule is defined as the percentage of transactions in some set, *T*, that contains *S*. The confidence of a rule is the ratio of the support of the set of items in the rule and the support of the rule's antecedent. Although there are other measures of rule quality, we are generally interested in identifying strong association rules, which we define as follows:

> An association rule *R* is **strong** if *support(R)> support_threshold* and *confidence(R) > confidence_threshold*.

We set *support_threshold* =.20 and *confidence_threshold* = .80.

## 3 Defining the Problem Space Using Unreal Tournament

Level architects of a game use a combination of their judgment and preferences as well as predefined spatial requirements in creating the navigation network of a map. For example, in *Unreal Tournament*, a first-person shooter (FPS) game, there are pre-defined node types that can be placed on a UT map level. A computer bot uses a *PathNode* to maneuver to different locations on the map. PathNodes are part of the bot navigation network. Typically, there are some requirements that should be followed in the placement of adjacent PathNodes[1]. Some sample rules include "Place PathNodes at intersections and turns so that there is as much clearance as possible", "Paths need a line of sight to each other, clear to the width of the maximum width of creatures who will use this path" and "PathNodes should not be placed more than 700 units apart" (An "Unreal Unit" is approximately .75 inches). The level design task, itself, also extends to placement of other types of nodes representing items (objects) of interest on a map such as weapons and ammunition.

Recent versions of UT have an automatic PathNode generation facility (part of *UnrealEd*). However, this only ac-

[1] http://unreal.epicgames.com/UT_AI.htm

complishes part of the task. For instance, automated methods do not place certain kinds of navigation nodes such as *PlayerStarts* and *LiftCenters*. The inferior quality of these automated methods has prompted many level architects to design paths manually.

To show the significance of using UT for our broader goals, we draw an analogy between the level architect of a game map and an architect of a city. Typically, we do not have access to the architect's plans. However, if we explore a few blocks of a city, we notice spatial characteristics and patterns and make assumptions about how buildings and streets are likely to be positioned in unexplored areas of the city. Combined with background knowledge, this information will eventually help navigate unknown areas.

Using both the Unreal Tournament Engine as well as the GameBots client[2], we are able to run JavaBots[3] that explore a UT map. JavaBots explore a UT map receiving messages from the UT Engine with attributes representing various kinds of sensory information. For our background knowledge, we use existing definitions of concepts and UT class hierarchies (trees) available from the Unreal Engine Documentation Site[4] and extend these with meronyms/holonyms for our initial ontology.

### 3.1 Visible and Reachable Nodes

Messages passed between the client and server allow JavaBots to identify "neighborhoods" for the current location of the bot based on sensory information. We use the sensory inputs to define the following spatial features:

- Visible nodes**:** Locations that are within the bot's line of sight.
- Reachable nodes: Locations that are within the bot's line of sight and that the bot can get a path to without running into an obstruction (e.g., falling into a pit).

In UT, a bot explores the map along navigation points, and when the bot searches for a new, unexplored node as its next target location, we record spatial neighborhoods. Specifically, we enumerate two sets of nodes: {*visible nodes*} and {*reachable nodes*}.

### 3.2 UT Object Ontology

For our purposes, the UT ontology should capture concepts related to important objects found on a UT map. We also represent two types of taxonomic relations: hypernym/hyponym (superclass/subclass) and meronym/holonym (part-of/has-part). An object on a UT map can belong to multiple classes. For example, we would categorize a *LiftCenter* and *LiftExit* as types of *NavigationPoints*. An expansion of the *NavigationPoint* subtree is shown in Figure 1. However, for practical purposes. *LiftCenter* and *LiftExit* are also part-of a "Lift" class. Therefore, in our UT ontology, we create the subtree for lifts shown in Figure 2. As we continue to develop the ontology, some of our existing classes will map to new categories. For example, if we create a category for "Defensive Position", we can add Lifts as one of its subclasses.

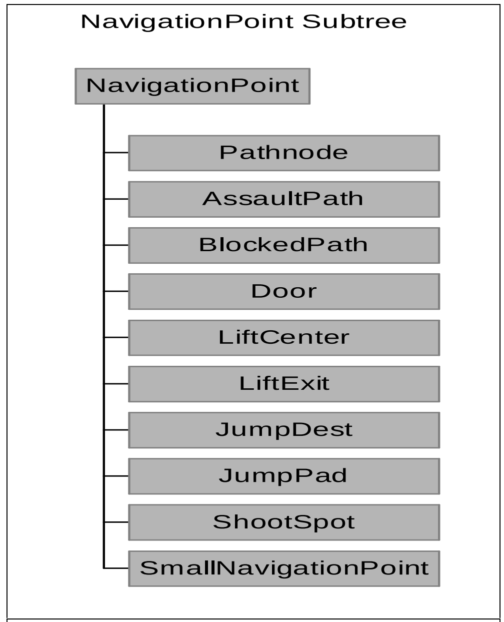



Figure 1. Direct descendants of NavigationPoint using the SubClass relation.

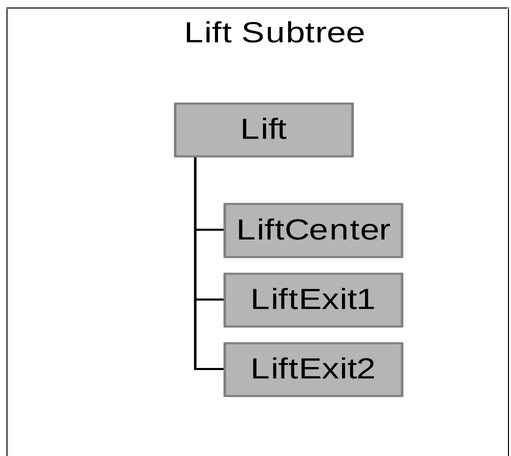



Figure 2. Part-of Expansion of the Lift Class.

We use the ontology to automate the process of differentiating between categories of association rules. There are many kinds of ontology-based similarity and relatedness measures [Rada, 1989; Resnick, 1995; Jiang and Conrath, 1997; Hirst and St. Onge, 1998]. Some measures are path-length metrics extended to account for information content of concepts, taxonomy depth, etc. In Section 4, we describe

[2] http://gamebots.sourceforge.net

[3] The JavaBot (http://utbot.sourceforge.net) API and "ExampleBot" were developed by Andrew N. Marshall at USC-ISI. "CMU_Jbot" (extends "ExampleBot") was created by Steve Schaffer and Chris Sollitto.

[4] http://wiki.beyondunreal.com/

how we use shortest path length (similar to conceptual distance described by Rada *et al.* [1989]), counting taxonomic links in an ontology, as an initial approximation of relatedness between concepts for categorization of rules.

## 4 Experimental Results

We outline the major steps in our experiment:

1) Using the JavaBot API, define discrete exploration events corresponding to bot transitions within the game environment.
2) As the bot explores a location, record sensory information received from the game server – nodes at locations within the bot's line of sight and nodes at locations reachable by the bot (without obstruction) – in two separate tables. Replace node instances with class names (e.g., InventorySpot) using our UT ontology.
3) Mine two event tables and select strong associations.
4) Categorize associations as co-locations satisfying some spatial requirement or novel spatial arrangements using ontology-based similarity measures. A predominance of spatial patterns explained by the ontology suggests a more predictable region (described by many "known" relationships); novel arrangements can be predictive if they occur often.

In Step 1, we record visibility and reachability sets for more than 1000 JavaBot exploration events for the UT game-type, "Friendly Remote Bot CTF" (where CTF stands for "Capture the Flag"), and UT map, "CTF-Command". We used a modified version of CMU_JBot as a starting point. In Step 2, we pre-processed the data substituting class names in place of instance names. For example, in our data set, this involves replacing an instance of "PathNode0" with the class, "PathNode". In Step 3, we ran *Apriori*, controlling the number of rules generated by initially setting minimum support at .10 and minimum confidence at .80. Table 1 shows sample high support, high confidence rules for reachable nodes selected using *support_threshold* and *confidence_threshold* (previously described for strong associations). Table 2 shows sample high confidence rules at different levels of support that were generated for visible nodes.

In Step 4, we measured whether strong spatial associations between concepts also implied semantic relatedness. We define semantic relatedness as follows:

> Concepts *A* and *B* that are taxonomically linked in the ontology are **semantically related** if *shortest_path_length*$(A,B) \leq 2$.

For example, we are interested in relations such as direct hypernymy/hyponymy, etc. For every strong association, we categorize it by assigning a label:

> **Non-Preference** (N-P): If the rule antecedent (concept A) and rule consequent (concept B) are semantically related.
>
> **Preference** (P): If the above condition does not hold.

| Association Rule | Support | Confidence |
|---|---|---|
| CTF-Command.bioammo → CTF-Command.goowand | 36.0% | 92.8% |
| CTF-Command.InventorySpot → CTF-Command.goowand | 30.5% | 87.3% |

**Table 1. Sample Association Rules for Reachable Nodes**

For a "Non-Preference" label, we conjecture the level architect was following a spatial requirement or guideline in the placement of objects on the map. Otherwise, the architect exhorts personal design preferences resulting in "semantically novel" spatial arrangements.

In Table 3, we summarize our association labeling results. For the first association, we note that the level architect placed two different kinds of objects such that they are reachable from one another in the explored space: *goowand*, which is a kind of *weapon*, and *bioammo*, which is a kind of *ammunition*. Both *goowand* and *bioammo* are children of different subtrees in the ontology and are at a distance > 2 links apart. Given the strong association, we can predict that for this map, the level architect has a preference for this association and is likely to repeat this spatial arrangement in other parts of the map.

Another interesting spatial association exists between *goowand* and *InventorySpot* – especially as goowands are not considered a type of inventory item. (Typical inventory items include: JumpBoots, Armor, Health/Healthpack, HealthVial/MedBox). This association also satisfies our criteria for labeling as "P".

Another kind of association is captured between 1) *LiftExits* and *PathNodes* and 2) *LiftExits* and *translocdest*. From the ontology fragment shown in Figure 1, we find that *LiftExits* are a kind of *NavigationPoint*. According to UT definitions, *translocdest* (not shown in figure) is a special type of *LiftCenter*, and *LiftCenter* is a kind of *NavigationPoint*.

In this first case, we know that *LiftExit* and *PathNode* are just 2 links away in the UT object ontology (sharing a common ancestor, *NavigationPoint*). We label this association as "N-P"; we also note that this relationship isn't very interesting. In the second case, we must traverse 3 links in the ontology to establish a path between *LiftExit* and *translocdest* (with *LiftCenter* and *NavigationPoint* as intermediate points). We incorrectly label this relationship as "P".

For the above data, we were able to assign a correct label 75% of the time. We repeated this exercise on single antecedent rules derived from the visible nodes data set and were able to correctly label associations approximately 80% of the time. Currently, we are investigating the use of other semantic relatedness metrics, such as the Hirst and St. Onge measure [1998], which takes into account a variety of relationship types as well as directionality of the links. We would like especially to explore the use of "horizontal relations" (at the same level of specificity) [Patwardhan *et al.*, 2003], applying this to coordinate terms or "siblings" in the ontology.

| Association Rule | Support | Confi-dence |
|---|---|---|
| CTF-Command.LiftExit → CTF-Command.PathNode | 50.2% | 100.0% |
| CTF-Command.InventorySpot → CTF-Command.goowand | 51.8% | 100.0% |
| CTF-Command.translocdest → CTF-Command.LiftExit | 23.0% | 100.0% |
| CTF-Command.PlayerStart → CTF-Command.LiftExit | 16.6% | 91.5% |
| CTF-Command.PlayerStart → CTF-Command.PathNode | 18.0% | 99.0% |
| CTF-Command.PlayerStart → CTF-Command.InventorySpot | 17.5% | 96.5% |

**Table 2. Sample Association Rules for Visible Nodes**

There are three association rules in Table 2 with *PlayerStart* (location where Player is spawned) as the rule antecedent. We do not consider these rules as strong associations. However, if we applied the above analysis, we observe that all three associations would be labeled, "P", since they do not meet the criteria for semantic relatedness (*PlayerStart* is a descendent of *SmallNavigationPoint* in the ontology). While it may be common practice to put *PlayerStarts* next to certain node types, the rules also give us the distributional preference of the architect in placing *PlayerStarts* near *LiftExits*, *PathNodes*, and *InventorySpots*.

Currently, we have two broad classes for categorization of association rules: spatial requirements and design preferences. Each of these can be refined further. For example, while design preferences give us distributional data about how often *X* is spatially correlated with *Y* for an individual (level architect), we do not distinguish between arrangements that are "common practices" (e.g., placing ammunition close to a weapon) in a game versus those that are "unexpected". We are exploring this topic further.

There has been prior work in mining association rules derived from text for novelty, using relations in *WordNet* [Fellbaum, 1998] and ontology-based similarity/relatedness measures [Basu *et al*, 2001]. Whereas Basu *et al.* [2001] applied ontological distance to find novel co-occurrences in text, we hypothesized that similar measures would be useful to label co-occurrences of objects in space. (That work used a derivative of the Hirst and St. Onge measure in their distance metric.) In this study, we have shown that we can *combine* both spatial and semantic knowledge to label associations as important and interesting. Our interpretation of interestingness is similar to that of the HR system [Colton *et al.*, 2000] (also discussed in [Johnson, 2005]) – i.e., co-locations of items that do not normally appear together are considered "interesting". We further label the interesting associations as expressing the preferences of an architect, which need to be learned by sensing the environment, or reflecting some "known" spatial requirement, which can be inferred from ontologies.

| Association Rule | Semantically Related Concepts | Label/ Correct |
|---|---|---|
| CTF-Command.bioammo → CTF-Command.goowand | No | P/Yes |
| CTF-Command.InventorySpot → CTF-Command.goowand | No | P/Yes |
| CTF-Command.LiftExit → CTF-Command.PathNode | Yes | N-P /Yes |
| CTF-Command.translocdest → CTF-Command.LiftExit | No | P/No |

**Table 3. Categorizing Association Rules**

## 5 Discussion

How can inferred associations be used to control an avatar that makes recommendations for navigating unexplored areas on a map? We believe that strong, non-preferential associations identify physical regions where an avatar can combine observed spatial knowledge with background knowledge to predict where to move next, thereby lessening the need for additional sensory inputs/processing.

A larger and more difficult problem we are interested in addressing is how to infer a map given limited sensory information about the world. We believe that one of the reasons humans are proficient at navigating unexplored areas is that they rely on background knowledge to help them understand spatial relationships. However, there are imminent challenges. Outside of the game environment, our knowledge of the world is often imperfect (inconsistent) and incomplete. How then can we use background knowledge effectively in a real-world scenario? We discuss how having a richer, dynamic ontology with spatial features and relations can be a part of an approach.

Even if we don't have access to an architect's plan for a city, we could create a specialized ontology from archival data sources. For example, we would extract different kinds of features from 2-dimensional maps of a city such as size, shape, height, and function of structures, locations of windows/doors, street names, street/sidewalk widths, property boundaries, building use, and block numbers (e.g., Sanborn maps[5] have captured this kind of information for New York City for many years). We would encode both common spatial features and their relationships in the ontology.

When no other source of data is available, we need to combine local exploration with more generalized world knowledge. For example, given an ontology of "Buildings" and "Rooms", a person walking into a room of a building for the first time finds associations between known spatial features of a room type (e.g., finding both a "refrigerator" and "oven" in a "kitchen") using relationships such as "containment" or "part-of". If an avatar detected any of these spatial features at a given location of a new room, it would be able to predict the room type. We should also include

[5] http://www.sanborn.com

typical spatial relations in the ontology – e.g., "next to" would connect concepts such as "door" and "stairs". So, even if the location of the main door of a building is not known, an avatar observing stairs in front of a building will predict that a door is close by. In this scenario, the avatar would begin with generic world knowledge in an ontology, observe the local context or environment, find relationships that either match or do not match known relationships in the ontology, and continually update the ontology based on available sensory information.

## 6 Conclusions

In this paper, we have described a method for discovering high-level relationships about spatial features on a map layout of a computer game. We have presented a study of two kinds of spatial relationships between nodes or spatial features on a game map that can be inferred using association rule mining and differentiated using an ontology. In a real-world scenario, we envision applying our methods as a discovery step for a digital avatar looking for both common and novel spatial co-location patterns that either reinforce or update (modify) a navigator's knowledge of the surroundings. Ultimately, the avatar would assist a human navigator in making faster and more informed decisions about how to respond to the sensory inputs emanating from the immediate environment.

## References


[Agarwal *et al*., 1993] R. Agarwal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data,* pages 206-216, 1993.

[Agarwal and Srikant, 1994] R. Agarwal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the International Conference on Very Large Databases (VLDB-94)*, pages 487-489, May, 1994.

[Basu *et al*., 2001] S. Basu, R. J. Mooney, K. Pasupuleti, and J. Ghosh. Evaluating the Novelty of Text-Mined Rules using lexical knowledge. In *Proceedings of the Knowledge Discovery and Data Mining Conference (KDD-2001)*, pages 233-238, 2001.

[Colton *et al.*, 2000] S. Colton, A. Bundy, and T. Walsh. On the Notion of Interestingness in Automated Mathematical Discovery, *International Journal of Human-Computer Studies*, Volume 53, Academic Press, 2000.

[Fellbaum, 1998] C. Fellbaum (ed.). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[Fortin and Liu, 1996] S. Fortin and L. Liu. An Object-Oriented Approach to Multi-level Association Rules. In *Proceedings of the International Conference on Information and Knowledge Management* (*CIKM-1996*), pages 65-72, 1996.

[Hirst and St. Onge, 1998] G. Hirst and D. St. Onge. Lexical Chains as Representations of Contexts for the Detection and Correction of Malapropisms. In C. Fellbaum, ed., *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.

[Jiang and Conrath, 1997] J. Jiang and D. Conrath. Semantic Similarity based on Corpus Statistics. In *Proceedings on International Conference on Research in Computational Linguistics*, Taiwan, 1997.

[Johnson, 2005] C. G. Johnson. Search and Notions of Creativity. Computational Creativity Workshop, Edinburgh, Scotland, July, 2005.

[Koperski and Han, 1995] K. Koperski and J. Han. Discovery of Spatial Association Rules in Geographic Information Databases, *Proceedings of the 4th International Symposium on Advances in Spatial Databases*, page 47-66, Portland, Maine, August 1995.

[Malerba *et al.*, 2002] D. Malerba, F. Esposito, F. A. Lisi, and A. Appice. Mining Spatial Association Rules in Census Data. *Research in Official Statistics*, 5(1):19–44, 2002.

[Mennis and Liu, 2003] J. Mennis and J. W. Liu. Mining Association Rules in Spatio-Temporal Data. In *Proceedings of the 7th International Conference on GeoComputation*, University of Southampton, UK, September, 2003.

[Nakagawa and Mobasher, 2003] M. Nakagawa and B. Mobasher. Impact of Site Characteristics on Recommendation Models Based on Association Rules and Sequential Patterns. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2003)*, 2003.

[Patwardhan *et al.*, 2003] S. Patwardhan, S. Banerjee, and T. Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 2003.

[Rada *et al.*, 1989] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and Application of a Metric on Semantic Nets. In *IEEE Transactions on Systems, Man, and Cybernetics*, 19 (1989), pages 17-30.

[Resnick, 1995] P. Resnick. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 448-453, 1995.

[Shadbolt, 2003] N. Shadbolt. Ambient Intelligence. *IEEE Intelligent Systems*, Vol. 18, No. 4, pages 2-3, 2003.

[Shekhar and Huang, 2001] S. Shekhar and Y. Huang. Discovering Spatial Co-location Patterns: A Summary of Results. In *Proceedings of the 7th International Symposium on Spatial and Temporal Databases*, 2001.

# Real Time Identification of Operating Room State from Video

**Beenish Bhatia, Tim Oates**
University of Maryland at Baltimore County
Department of CSEE
1000 Hilltop Circle, ITE 376, Baltimore, MD 21250
beenish1@umbc.edu, oates@cs.umbc.edu
**Yan Xiao, Peter Hu**
University of Maryland at Baltimore
Department of Anesthesiology
685 W. Baltimore Street, MSTF 534, Baltimore, MD 21201
yxiao@umaryland.edu, phu@umm.edu

## Abstract

Managers of operating rooms (ORs) and of units upstream and downstream of the OR (e.g., post-anesthesia care) seek real-time information about OR occupancy to make decisions about managing OR workflow and coordinating resources. Nursing and anesthesia staff typically record patient in/out times by hand, and OR managers spend time walking about the OR suite to estimate the time each case will finish. This paper describes a system for using real-time video to automatically identify the state of an ongoing operation. This state information is relevant to determining OR occupancy and estimating time to case completion. The system, which uses support vector machines to learn to identify image features relevant to state identification and hidden Markov models to capture the sequential nature of the domain, was tested on video captured over a two day period in one of the nineteen ORs in Baltimore's R. Adams Cowley Shock Trauma Center. An overall accuracy of 99.5% was obtained for identifying for each video frame the corresponding operation state, which was one of the following: OR ready, patient entering, operation in progress, operation ending, patient exiting. Our results are contrasted with the current state-of-the-art system used by the Shock Trauma Center which is based on patient vital signs data

## 1 Introduction

T R. Adams Cowley Shock trauma center in Baltimore is the nation's only Trauma Hospital. It is a state-of-the art facility, with helicopters flying in patients who seek emergency treatment for unexpected injuries, trauma, etc. It has given the world the concept of the golden hour, which is the first hour after an injury occurs. This trauma center is especially important in a city like Baltimore because of the high crime rate. Scheduling is a difficult problem in general, and it is even more complicated in a hospital environment. The trauma center has a unique blend of cases – some are short and some are long, some patients are healthy and some are sick, and frequent disruptions in the planned OR schedule make this a very dynamic scheduling problem. Managers of ORs and of units upstream and downstream seek real time information about OR occupancy to coordinate resources and other workflow [Dexter *et al*., 2004]. The nursing and anesthesia staff record patient in/out times by hand and the OR managers physically walk around and look into the OR suite to check on the status of ongoing operations. You would typically see in the cubicle of nurses and other staff sheets of paper recording the patient in/out time as well mentioning the source from which the time was obtained (wrist watch, wall clock, clock on desk etc) [Moss and Xiao, 2004]. With advances in technology like electromechanical systems, telecommunication and video board systems there is potential for OR management decisions in real time [Overdyk *et al.*, 1998] [Hu *et al*., 2006]. It is necessary to improve the quality of scheduling in the OR as both the underutilization and over utilization (staff staying late) of an OR is very expensive. Many times a surgeon has to wait as the OR was not scheduled at the right time. This is a waste of time, money (doctors are very expensive) and resources. Also, in a country where there is such shortage of nursing staff, it is vitally important that their utilization is optimum.

The current system in use [Xiao *et al*., 2005] is based on Vital Signs (VS) which handles two cases - patient in/out as well as patient in-time/out-time. It uses the data based on VS coming from a patient to determine OR occupancy. VS data is gathered from various sources like ECG, $SpO_2$, non-invasive blood pressure, temperature and arterial blood pressure. Much effort is required to collect this data with automated technology and electromechanical systems without human intervention. It has time lag of 3-5 minutes, which is a big drawback. Also other core issues not dealt with include whether the OR is ready, an operation is going on, etc. which makes it impossible to schedule room cleanups, making up beds, and so on.

The system proposed is based on sampling images from video to determine the OR state. Our system uses support vector machines as an initial classifier to identify features which are essential states required for scheduling the OR. A hidden Markov model is used to capture the sequential nature of these states. Our system has been shown to perform better in the patient in/out case and as well does not have any time lag. Also, it is much easier to capture video data compared to different sources of VS data.

The managers of the OR could use this information to schedule the OR in real time and as all of this is done automatically with no human intervention it makes it very cost effective. Also the staff assigned to do this task could be relieved for other tasks. It could be used for calculating standard scheduling issues like workload, and efficiency issues about the different ORs. The system developed is totally independent of any patient or staff collected information and hence can be deployed with consummate ease at other hospitals.

## 2 Video/Image Pre-processing

The video used in the experiments has been gathered from real ORs where cameras were placed in two different positions. The data were collected for twenty four continuous hours at 30 frames per second in avi format. It was further split into 24 one hour videos. Each of the one hour videos were converted from avi to mpeg format using cleaner software. JPEG images were made of each of the videos using ImagetoVideo software at 3 images per second. The total number of images was 2*24*60*60*3= 518400 and each hour of data has 10800 images. All the images are in RGB format with size of 352*240 pixels. Images were converted from RGB to HSV (Hue, Saturation and Value) format. Color histograms were made of the images using the Hue value with bin size of 256. Histogram counts were normalized in the range of 0-1 by dividing with the largest count. Each one hour of data had 256 features with the corresponding data label. The data was given the label manually.

Four important states are targeted for classification:

- Single Bed: Is the state when the OR is ready for a patient (Fig 1.1)
- Double Bed: Actual time when the patient is wheeled in and transferred to the operating bed (Fig 1.2)
- Drapes on: In this the OR is ready for the actual surgery (Fig 1.3)
- Drapes off: Actual surgery has been completed and the patient will be wheeled out soon (Fig 1.4)

7 hours of data was available for the four states mentioned above, 4 hours for drapes on/ drapes off (DON/DOF) and 3 hours for single bed/double bed (SB/DB). These four states were targeted as they are the core issues for scheduling ORs. For example the transition from Single Bed to Double bed and vice versa corresponds to patient in and patient out. Similarly transition from Drapes On to Drapes Off and vice versa corresponds to start of operation and end.

Figure 1.1: Single Bed

Figure 1.2: Double Bed

Figure 1.3: Drapes On

Figure 1.4: Drapes Off

## 2.2 Support Vector Machines

SVM-light, an implementation of Support Vector Machines in C, was used [Joachims, 2002]. All the images in seven hours of video data were given an initial label manually. Two separate classifiers were trained, one each for SB/DB and DON/DOF. All the images with a single bed were labeled +1 and those with a double bed were labeled -1. Similarly all the images with Drapes On were labeled +1 and Drapes Of -1. From the 4 hours of DON/DOF data we learn the best classifier by choosing 3 hours of data that best classifies the 1 hour of data, similarly in SB/DB we chose 2 hours of data that best classifies 1 hour of data. We used different kernels like linear, polynomial, radial basis function, Gaussian and tried the different parameters. Out of the different kernels, the linear kernel with default values performed the best. After the two classifiers that performed the best were generated, both of them were used to classify the other data. To be precise, the classifier learned from SB/DB was used for classification of images from 1 hour of test data of DON/DOF and the classifier learned from DON/DOF was used for classification of images from SB/DB. This gives us the following outcomes:

- DON, SB
- DON, DB
- DOF, SB
- DOF, DB

Further we calculated the observation probability as follows:

- Observation probability of an outcome given a state (SB/DB/DON/DOF) = count of each outcome/Number of instances in that state
- For example the probability of (SB,DOF/Single Bed) should be high (8780/8922 = 0.9839) and the probability of (DB,DOF/Single Bed) should be low (0.0019).

### 2.3 Hidden Markov Model

Hidden Markov Model (HMM) [Rabiner, 1989] implemented in Matlab was used. Two initial classifiers using SVMs were used as a starting point for HMM. Four states HMM with SB, DB, DON and DOF were used, (Fig 2).

We used an HMM because of the nature of the domain and similar methods have been applied in other fields [Xie *et al*., 2002]. If you notice the OR procedure is a sequential process, where there is single bed (state 1) then a patient is wheeled in and momentarily there is double bed (state 2). Next the double bed is taken out and the drapes are put on for operating (state 3). After the operation is finished the drapes are taken off (state 4) and then again the bed is brought to wheel out the patient (state 2). After the patient is wheeled out in we return back to single bed (state 1).

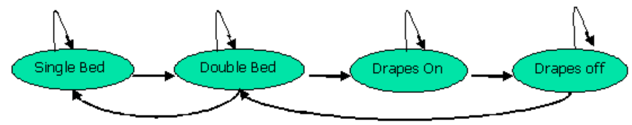



Figure 2: HMM for the domain

Training an HMM involves setting the values of transition and emission probabilities in the four states. Basically, we made the transition probability from one state to another very low ($10^{-40}$). This biases the model to be in the state which it is in currently. State changes which are transitions in the state of the OR are rare. The emission probabilities were determined from the observation probabilities calculated using the different states in the SVM classifier (Table 1). The columns (2-5) are the different states in the HMM and the rows (2-5) are the outcomes of the states.

| Different outcomes | Single bed (state 1) | Double bed (state 2) | Drapes on (state 3) | Drapes off (state 4) |
|---|---|---|---|---|
| DB,DOF | 0.0019 | 0.7891 | 0.0019 | 0.0435 |
| DB,DON | 0.0023 | 0.0315 | 0.0849 | 0.0771 |
| SB,DOF | 0.9840 | 0.1757 | 0.0128 | 0.4604 |
| SB,DON | 0.0118 | 0.0037 | 0.9004 | 0.4190 |

Table 1: Observation probabilities of different outcomes in various states of HMM

Further investigation of errors made us consider windowing as a good enhancement. The window procedure involved taking some number of images into a buffer and then predicting the outcome based on voting. We tried using windows of different size, the details of which follow in the next section.

## 3 Results

We used two hours of image data for our experiments. These images consisted of four outcomes - single bed, double bed, drapes on and drapes off. The images were labeled manually. SVM$^{light}$ Version 6.01, an implementation of Support Vector Machines (SVMs) in C, was used. Matlab scripts were used for preprocessing of images, as well as for the Hidden Markov Model (HMM) implementation

## 3.1 Support Vector Machines

The following section presents the training and testing results obtained using the SVMs.

### 3.1.1 Training and Testing

Training of SVM: Two separate SVMs were trained:

1. Single Bed, Double Bed: Training of SVM involved combining two hours of data and testing on one hour of data. We did not use leave one out as the criterion for training because the instances were not independent and identically distributed (i.i.d) due to the fact that we were sampling at three images per second (very little changes). This basically meant that we were testing on instances that we trained upon. Total number of images: 32377 (SB=24534, DB=7843); for training: 21583 (SB=15612, DB=5971); for testing: 10794 (SB=8922, DB=1872). Best case accuracy of SVM on test set was 96.54 (Table 2).

| Training | Testing | Accuracy (%) |
|---|---|---|
| 07_08-12_13 | 13_14 | 96.54 |
| 07_08-13_14 | 12_13 | 70.93 |
| 12_13-13_14 | 07_08 | 57.33 |

Table 2: Accuracy of SVMs on SB/DB

2. Drapes on Drapes off: Similar training method was followed by combining three hours of data. Total number of images: 43164 (DON=22529, DOF=20635); for training 32371: (DON=14885, DOF=17486); for testing 10793: (DON=7644, DOF=3149). Best case accuracy of SVM on test set was 93.43 (Table 3). We did not use the classifier with the highest accuracy as 90% of the instances in the test set were of the same label. Instead we used the classifier with accuracy 84.18 as the instances had more even distribution (70% +ve, 30% -ve).

| Training | Testing | Accuracy (%) |
|---|---|---|
| 08_09-11_12-14_15 | 18_19 | 93.43 |
| 08_09-11_12-18_19 | 14_15 | 88.03 |
| 08_09-14_15-18_19 | 11_12 | 65.52 |
| 11_12-14_15-18_19 | 08_09 | 84.18 |

Table 3: Accuracy of SVMs on DON/DOF

## 3.2 Hidden Markov Model

The following section presents the training and testing results obtained using HMM.

### 3.2.1 Training and Testing

Training of HMM: Four states HMM was trained as shown in Fig 2. The transition probability from one state to another was kept very low $10^{-40}$. The observation probability was set by substituting values from Table 1. Viterbi algorithm was used for testing by providing it the sequence of outcomes and then predicting the sequence of states which most likely generated it.

Three separate cases were handled:

1. Single bed/Double bed: Total number of images: 32377 (SB=24534, DB=7843); for training: 21583 (SB=15612, DB=5971); for testing: 10794 (SB=8922, DB=1872). The accuracy was 99.95 (Fig 3).

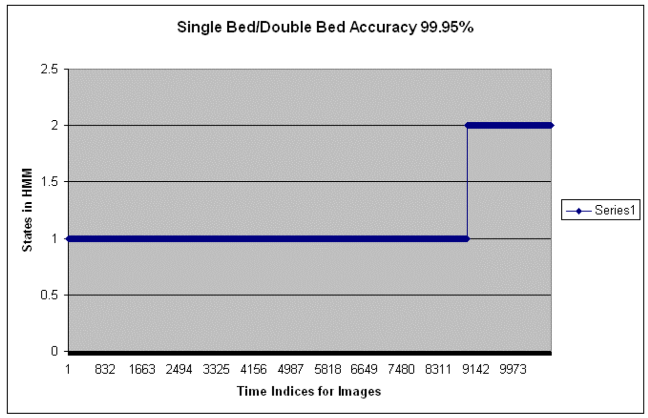



Figure 3: Accuracy of HMM on SB/DB

2. Drapes On/Drapes Off: Total number of images: 43164 (DON=22529, DOF=20635); for training: 32371 (DON=14885, DOF=17486); for testing: 10793 (DON=7644, DOF=3149). The accuracy was 99.30% (Fig 4).

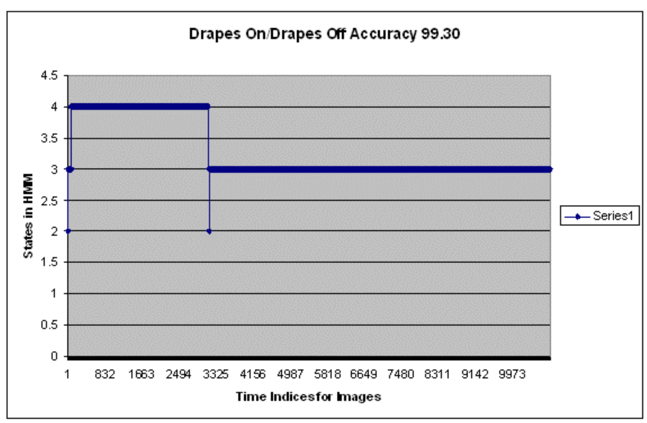



Figure 4: Accuracy of HMM on DON/DOF

3. Merged Single bed/Double bed-Drapes On/Drapes Off: Total number of images: 32377+43164 = 75541 (SB=24534, DB=7843, DON=22529, DOF=20635); for training: 21583+32371 = 53954

(SB=15612, DB=5971, DON=14885, DOF=17486); for testing: 10794+10793= 21587 (SB=8922, DB=1872, DON=7644, DOF=3149). The accuracy was 99.49% (Fig 5).

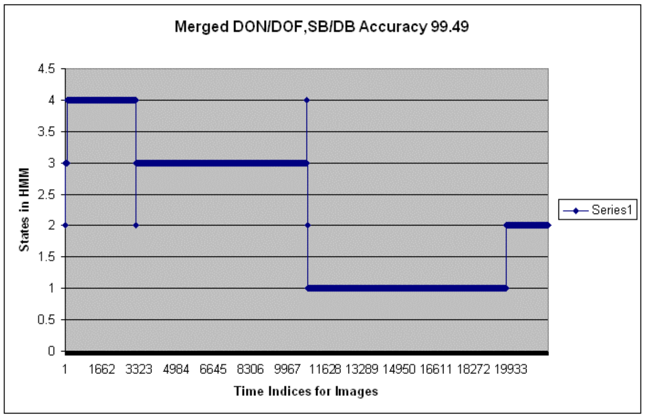



Figure 5: Accuracy of HMM on merged DON/DOF-SB/DB

### 3.3 Windowing Procedure

Results with Windowing Procedure: The window procedure involved taking some number of images into a buffer and then predicting the outcome based on voting. Again three separate cases were handled, single bed/double bed, drapes on/drapes off and merged single bed/double bed-drapes On/drapes off having 10794, 10793 and 21587 test images respectively. Table 4 below gives the results in detail with the different sizes of the window considered and accuracy.

| Size of Window in images | Single bed/Double bed | Drapes On/Drapes Off | Merged SB/DB-DON/DOF |
|---|---|---|---|
| 13 | 100 | 99.30 | 99.49 |
| 23 | 99.95 | 99.29 | 99.45 |
| 53 | 99.81 | 99.29 | 99.31 |
| 93 | 99.63 | 99.29 | 99.19 |
| 193 | 99.15 | 99.01 | 98.58 |
| 293 | 98.67 | 98.52 | 97.87 |
| 393 | 98.17 | 98.03 | 97.15 |
| 493 | 97.67 | 97.52 | 96.43 |

Table 4: Accuracy in (%) with windowing procedure

If you notice that with window size of 13 images we obtain similar accuracy as without window. We therefore dropped the window procedure and kept the prediction totally real time.

## 4 Conclusion and Future Work

We have presented a novel way of scheduling ORs using support vector machines and hidden markov models. Our approach of development of support vector machines as initial classifier to provide hidden markov model states is unique. We have shown that this approach is robust to changes in the states of the OR. Also, the data used is from real ORs, collected over twenty hours from Shock trauma center in University of Maryland at Baltimore hospital.

Our approach has been shown to have higher classification accuracy than systems currently in use (99.5% compared to 95%). It also has instantaneous classification which is not present in current systems (have 2-5 minute delays). Our system is more robust using video than system presently in use with vital signs (there can be loss of signal in vital signs).

In future, we plan to deploy our system in Shock Trauma Center replacing the current system based on vital signs. This involves doing more rigorous testing of the method developed. We plan to test on five days of video data coming from ORs. Also, we would be testing it on different ORs (current data was collected from OR5, there are 6 ORs OR1-6). This is essential as the structure of each OR is different.

We also want to incorporate other states in the hidden markov model which would split the OR procedure into more stages. For example, we have modeled only four core issues and it would be interesting to know about room clean/not clean, black bed/white bed, etc. This would help in scheduling the OR in more precise manner with respect to various stages. Other goals are to make this system commercial and help hospitals all over the country in scheduling ORs.

## References


[Xiao *et al*., 2005] Yan Xiao, Peter Hu, Hu Hao, Danny Ho, Franklin Dexter, Colin F. Mackenzie, F. Jacob Seagull, and Richard Dutton. An Algorithm for processing vital sign monitoring data to remotely identify operating room vacancy in real time. *Anesth Analg.,* 101(3):823–829, September 2005.

[Xie *et al*., 2002] L. Xie, S.-F. Chang, A. Divakaran and H. Sun. Structure analysis of soccer video with hidden Markov models. *In Proceedings of IEEE International conference on Acoustics, Speech and Signal Processing (ICASSP)*, Orlando, FL, USA 2002.

[Hu *et al*., 2006] Peter Hu, Danny Ho, Colin F. Mackenzie, Hu Hao, Douglas Martz, John Jacobs, Roger Voigt and Yan Xiao. Advanced Telecommunication Platform for Operating Room Coordination: Distributed Video Board System. *Surgical Innovation*, 13(2):129–135, June 2006.

[Joachims, 2002] Thorsten Joachims. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.)*, MIT-Press, 1999.

[Rabiner, 1989] Lawrence Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, vol 77, no 2, February 1989

[Dexter *et al*., 2004] Dexter F, Epstein RH, Traub RD and Xiao Y. Making management decisions on the day of surgery based on operating room efficiency and patient waiting times. *Anesthesiology,* 101:1444-53, December 2004.

[Moss and Xiao, 2004] Moss J and Xiao Y. Improving operating room coordination: communication pattern assessment. *The Journal of nursing administration*, 34:93-100, February 2004.

[Overdyk *et al.*, 1998] Overdyk FJ, Harvey SC, Fishman RL and Shippey F. Successful strategies for improving operating room efficiency at academic institutions. *Anesth Analg,* 86(4):896-906, 1998.

# Ambient Intelligence in Emotion Based Ubiquitous Decision Making *

**Goreti Marreiros[1,2], Ricardo Santos[1,3], Carlos Ramos[1,2], José Neves[4], Paulo Novais[4], José Machado[4], José Bulas-Cruz[5]**

[1]GECAD – Knowledge Engineering and Decision Support Group, Porto, Portugal
[2]Institute of Engineering – Polytechnic of Porto, Porto, Portugal
[3]College of Management and Technology– Polytechnic of Porto, Felgueiras, Portugal
[4]University of Minho, Braga, Portugal
[5]University of Trás-os-Montes e Alto Douro, Vila Real, Portugal

## Abstract

As the time goes on, it is a question of common sense to involve in the process of decision making people scattered around the globe. Groups are created in a formal or informal way, exchange ideas or engage in a process of argumentation and counter-argumentation, negotiate, cooperate, collaborate or even discuss techniques and/or methodologies for problem solving. In this work it is proposed an agent-based architecture to support a ubiquitous group decision support system, i.e. based on the concept of agent, which is able to exhibit intelligent, and emotional-aware behaviour, and support argumentation, through interaction with individual persons or groups. It is enforced the paradigm of Mixed Initiative Systems, so the initiative is to be pushed by human users and/or intelligent agents.

## 1 Introduction

Despite the great variety of Decision Support Systems (DSS) tools and techniques, most are simple artefacts developed to help a particular user involved in a specific decision process. However, groups are used to make decisions about some subject of interest for the organization or community in which they are involved. The scope of such decisions can be diverse. It can be related to economic or political affairs like, for instance, the acquisition of new military equipment. But it can also be a trivial decision making as the choice about a holiday destination by a group of friends. It may be claimed, therefore, that Group Decision Support Systems (GDSS) have emerged as the factor that makes the difference when one assess the behaviour and performance of different computational systems in different applications domains, with a special focus on socialization.

If the group members are dispersed in time and space, the need of coordination, informal and formal forms of communication and information sharing will increase significantly. In this work it is proposed an architecture for a ubiquitous group decision support system that is able to help people in group decision making processes and considers the emotional factors of participants and their associated processes of argumentation. This system is intended to be used for intelligent decision making, a part of an ambient intelligence environment where networks of computers, information and services are shared. As an example of a potential scenario, it is considered a distributed meeting involving people in different locations (some in a meeting room, others in their offices, possibly in different countries) with access to different devices (e.g. computers, PDAs, mobile phones, or even embedded systems as part of the meeting room or of their clothes) (Figure 1). This meeting is distributed but it is also asynchronous, so participants do not need to be involved at any time. However, when interacting with the system, a meeting participant may wish to receive information as it appears. Meetings are important events where ideas are exposed, alternatives are considered, argumentation and negotiation take place, and where the emotional aspects of the participants are so important as the rational ones. This system will help participants, showing available information and knowledge, analyzing the meeting trends and suggesting arguments to be exchanged with others.

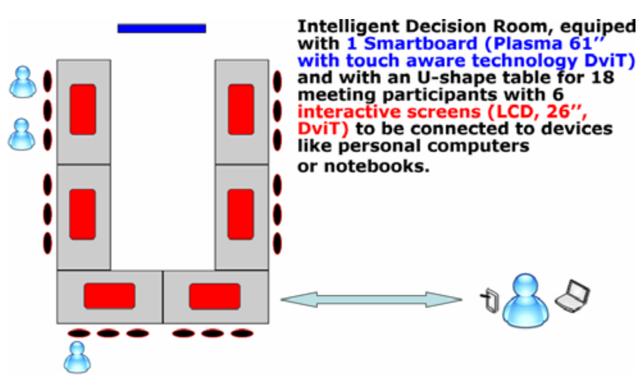



**Figure 1 - Distributed decision meeting**

* The work is partially supported by ArgEmotionAgents project (POSI / EIA / 56259 / 2004), which is a project supported by FCT (Science & Technology Foundation – Portugal).

The use of multi-agent systems is very suitable to simulate the behaviour of groups of people working together and, in particular, to group decision making modelling, once it caters for individual modelling, flexibility and data distribution [Marreiros et al., 2005a]. In classical decision theory, proposals are chosen by individual decision makers in order to maximize the expected coefficient of utility. However, when those choices are transposed to quotidian life, it is almost impossible to say that decisions are not influenced by emotions and moods.

## 2 Ubiquitous group decision making

Jonathan Grudin [Grudin, 2002] classifies the digital technology to support the group interaction into three phases: the pre-ubiquitous, the proto-ubiquitous and the ubiquitous ones. In the pre-ubiquitous phase, that begun in the 70's, it was supported face-to-face meetings. In the proto-ubiquitous phase, distributed meetings were supported. This phase come to life in the 90's. The ubiquitous phase is now getting under way, supports meetings, and it is distributed in time and space. The proposed system will be built to develop distributed and asynchronous decision meetings or social events.

Ubiquitous computing was introduced by Mark Weiser [Weiser, 1991], and anticipates a digital world which consists in many distributed devices that interact with users in a natural way. This vision was too far ahead for its time, however the hardware to implement Mark Weiser's vision is now commercially available and at a low cost. In an ambient intelligent environment, people are surrounded with networks of embedded intelligent devices providing ubiquitous information, communication and services. Intelligent devices are available whenever needed, enabled by simple or effortless interactions, attuned to senses, adaptive to users and contexts, and acting autonomously. High quality information and content may therefore be available to any user, anywhere, at any time, and on any device.

Today, there is an increasing interest in the development of Group Decision Support Systems (GDSS) to formalize and develop "any time and any place" group decision making processes, instead of "same place and same time" ones. This interest came with the need of joining the best potential group of participants. With the economy globalization, possible participants to form the group, like specialist or experts in specific areas, are located in different points of the world and there was no way to put them in the same decision room. Until some years ago, a way out of this scenario was to wait until all the participants meet together. Actually, there is a growing interest in developing systems to hold up such scenarios.

There are many areas where ubiquitous group decision making apparently makes sense. One of the most cited areas in literature is healthcare, since patient's treatment involves several specialists, like physicians, nurses, laboratory assistants, radiologists. These specialists could be distributed along departments, hospitals or even living in different countries. The HERMES system, a web-based GDSS was tested according to this scenario [Karacapilidis et al., 2001]. There are other GDSS that support ubiquitous decision making (GroupSystems software; WebMeeting [Marreiros et al., 2004]; VisionQuest software).

## 3 Mixed initiative systems

In the last years researchers had pondered on the merits of a total automation of user necessities (via intelligent agents) versus a total user control of operations and decisions (via Graphical User Interfaces) [Shneiderman and Maes, 1997].

Indeed, there is an interesting dualism between Artificial Intelligence (AI) and Human Computer Interaction (HCI). In AI, one tries to model the human thinking by creating computer systems able to do intelligent actions. In HCI, one designs computer interfaces to attract the user, supporting him/her to execute intelligent actions. The link between these two fields is the so-called Mixed Initiative Interaction. This refers to a flexible interaction strategy, where each agent (human or computer) can contribute to problem resolution with their best at their right moment [Hearst, 1999]. The concept of Mixed Initiative Systems is quite adequate to the group decision field. It is certainly very useful for a participant in a meeting to be assisted by a system able to show the available information and knowledge, analyzing the meeting trends and suggesting arguments to be exchanged with other specific participants.

In this work, a mixed initiative system is seen as a collection of joint processes linking users and a community of computer agents that work as a whole, like a personal assistant to satisfy user needs. Agent-based systems are ideal to accomplish this purpose due to their autonomy and independence to execute repetitive, boring and time consuming tasks. Cesta and D'Aloisi [Cesta and D'Aloisi, 1999] identified three characteristics needed in order to make the agents to participate in a mixed initiative interaction with users:

- Agents should adapt their behaviour to the user they are supporting, according to the philosophy of adaptive interfaces;
- Agents should act according to some principles of initiative shift taking among them, their users and other agents in the environment (when they exist);
- Agents should give to the user a level of 'super-control' in order to enhance the sense of trust between them. The user must have the possibility of inspecting the agent and then it will be able to prevent any undesirable operation and/or failure.

The proposed model is a duo between participants and agents with a various level of trust in order to work more effectively. This solution based on the levels of trust should solve the problem of the reliance on the agents. Any participant of the meeting would question if the agent representing him/her was doing the right thing. In this way, participants can inspect the agent to see what arguments are being used in the interaction process with other agents or participants in order to increase or decrease its reliance. A key aspect of the agent is its profile that should be specified in an uninterrupted process, continuously changing in order to have the lager possible amount of information in order to be able to predict the participant actions and needs.

## 4 Ubiquitous system architecture

One's aim is to present a ubiquitous system able to exhibit an intelligent and emotional behaviour in the interaction with individual persons and groups. This system supports persons in group decision making processes considering the emotional factors of the intervenient participants, as well as the argumentation process.

Groups and social systems are modelled by intelligent agents that will be simulated considering emotional aspects, to have an idea of possible trends in social/group interactions.

The main goals of the system are:

- The use of a simplified model of Groups and Social Systems for Decision Making processes, balancing Emotional and Rational aspects in a correct way;
- The use of a decision making simulation system to support meeting participants. This will involve the emotional component in the decision making process;
- The use of an argumentation support system, suggesting arguments to be used by a meeting participant in the interaction with other participants;
- The mixed initiative interface for the developed system;
- The availability of the system in order to be used in any place (e.g. meeting room, using a web based tool), in different devices (e.g. computers, notebooks, PDAs) and at different times (e.g. on-line meeting, asynchronous meetings).

The system consists of a suite of applications as depicted in Figure 2.

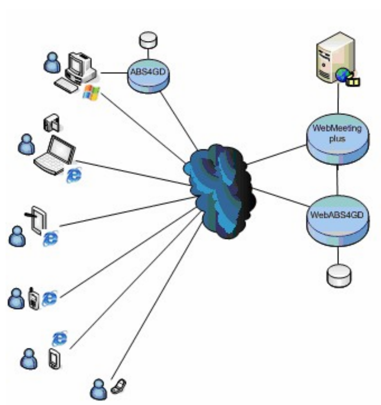



**Figure 2 - System architecture**

The main blocks of the system are:

- **WebMeeting Plus** – this is an evolution of the *WebMeeting* project with extended features for audio and video streaming. In its initial version, based on *WebMeeting* [Marreiros et al., 2004], it was designed as a GDSS that supports distributed and asynchronous meetings through the Internet. The *WebMeeting* system is focused on multi-criteria problems, where there are several alternatives that are evaluated by various decision criteria. Moreover, the system is intended to provide support for the activities associated with the whole meeting life cycle, i.e. from the pre-meeting phase to the post-meeting phase. The system aims to support the activities of two distinct types of users: ordinary group "members" and the "facilitator". The system works by allowing participants to post arguments in favour/neutral/against the different alternatives being discussed to address a particular problem. It is also a window to the information repository for the current problem. This is a web based application accessible by desktop and mobile browsers and eventually WML for WAP browsers;
- **ABS4GD** – this is the simulation tool resulting from the *ArgEmotionAgents* project. ABS4GD (Agent Based Simulation for Group Decision) is a multi-agent simulator system whose aim is to simulate group decision making processes, considering emotional and argumentative factors of the participants. ABS4GD is composed by several agents, but the more relevant are the participant agents that simulate the human beings of a decision meeting (this decision making process is influenced by the emotional state of the agents and by the exchanged arguments). The user maintains a database of participant's profiles and the model´s history of the group; this model is built incrementally during the different interactions of the user in the system.
- **WebABS4GD** – this is a web version of the ABS4GD tool to be used by users with limited computational power (e.g. mobile phones) or users accessing the system through the Internet. The database of profiles and history will not be shared by all users, allowing for a user to securely store its data on the server database, which guarantees that his/her model will be available for him or her at any time.

## 5 Multi-agent model

Multi-agent systems seem to be quite suitable to simulate the behaviour of groups of people working together [Marreiros et al., 2005a], as well as to assist the participants presenting new arguments and feeding the simulation model of the group by observing the interaction and history of the meeting.

Each participant of the group decision making process is associated with a set of agents to interact with other participants. The community should be persistent because it is necessary to have information about previous group decision making processes, focusing credibility, reputation and past behaviours of other participants.

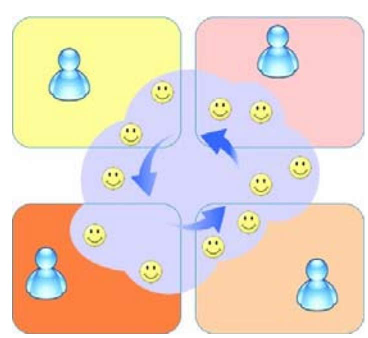


**Figure 3 - Multi-agent model**

The participant should have access to an Agent Based Simulation Tool for Group Decision (AGS4GD) developed under the *ArgEmotionAgents* project. This tool will improve the knowledge of the community of agents, then making possible to predict the behaviour of other participants and to advice on the best practice.

This support to the participants will be implemented using mixed initiative interaction. According to this concept, Intelligent Agent Based Systems can offer solutions where the user is allowed to change the proposed ones (e.g. to a particular problem), permitting the user to learn at the same time with his/her interactions, changing algorithms and models, therefore closing the gap on its view of the world in future interactions.

# 6 ABS4GD description

According to Zachary and Ryder [Zachary and Ryder, 1997] there are two different ways to give support to decision makers. The first one is supporting them in a specific decision situation. The second one intends to give them training facilities in order to acquire competencies and knowledge to be used in a real decision group meeting.

In our approach the decision making simulation process considers emotional aspects and several rounds of possible argumentation between meeting participants. The simulator is composed of several agents, but the more relevant are the participant agents that simulate the human participants of a meeting. This decision making process is influenced by the emotional state of the agents and by the exchanged arguments [Marreiros et al., 2006a]. A database of profiles and history with the group's model is maintained and this model is built incrementally during the different interactions with the system. It is important to notice that this simulator was not developed in order to substitute a meeting or even to substitute some meeting participants. The simulator is a tool that can be used by one or more participants to simulate possible scenarios, to identify possible trends and to assist these participants (in this way it can be seen as a what-if tool of a decision support system). However, the criteria used by this decision support system are not just rational, since they will consider emotions [Santos et al., 2006].

In this section it is characterized the decision problem and the group decision making protocol, also detailing the main components of this simulator, with particular focus on argumentation and emotion.

## 6.1 Decision problem configuration

The alternatives are completely identified by the participant agents. Let $A=\{A_1,A_2,\ldots,A_n\}$ be an enumerated set of $n$ alternatives, where $n >= 2$. The criteria are also known. Let $C=\{C_1,C_2,\ldots,C_m\}$ be an enumerated set where $m >= 2$. The decision matrix will be composed of n alternatives and m criteria. Let $D = [D_{ij}]_{nxm}$ where $D_{ij}$ represents the value of the alternative $Ai$ respectively to criterion $Cj$, and $i = 1,\ldots n$ and $j = 1,\ldots m$.

The participants of a specific simulation constitute the set $AgP = \{AgP_1,\ldots,AgP_k\}$, where k is the number of participants and $k >= 2$. Each $AgP_i$ has defined a set of weights for the criteria. Let $W_{AgPi} = \{ W_{C1},\ldots,W_{Cm}\}$ be the set of weights for $AgPi$, where $\sum_{j=1}^{m} W_{C_j} = 1,\ W_{C_j} \geq 0$, standing for the definition of a multi-criteria problem.

## 6.2 Group decision making simulation protocol

It is possible to find several classifications of decision models and problem solving. One of the most cited is Simon's classification that identifies the following phases: intelligence, design, choice and implementation [Simon, 1960]. Another classification is based on the political model, in which the decision is seen as a consequence of strategies and tactics used by individuals, aiming that the final result is the most advantageous [Salancik, 1977]. In this model, it is assumed that group members have different and possibly conflicting goals, leading to problems of conflict resolution and of power relations among them.

The proposed protocol combine the ideas mentioned before, with the particularity that here one is only considering the choice phase (Figure 4). It is not handled the pre-decision one, where the decision problem is taken in consideration as well as the simulation parameters (e.g. approving rule, duration).

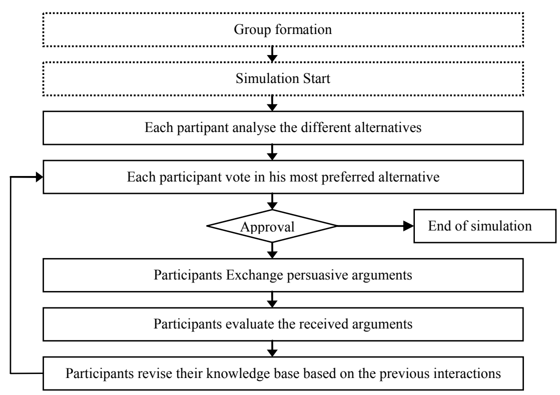



**Figure 4 - Group decision protocol**

## 6.3 Argumentation system

During a group decision making simulation, participants' agents may exchange the following locutions: request, refuse, accept, request with argument.

*Request* $(AgP_i, AgP_j, \alpha, arg)$ - in this case agent $AgP_i$ is asking agent $AgP_j$ to perform action $\alpha$, the parameter *arg* may be void and in that case it is a request without argument or may have one of the arguments specified in the end of this section.

*Accept* $(AgP_j, AgP_i, \alpha)$ - in this case agent $AgP_j$ is telling agent $AgP_i$ that it accepts its request to perform $\alpha$.

*Refuse* $(AgP_j, AgP_i, \alpha)$ - in this case agent $AgP_j$ is telling agent $AgP_i$ that it cannot accept its request to perform $\alpha$.

In Figure 5, it is possible to see the argumentation protocol for two agents. However, note that this is the simplest scenario, because in reality, group decision making involves more than two agents and, at the same time $AgP_i$ is trying to persuade $AgP_j$ that, this agent may be involved in other persuasion dialogues with other group members.

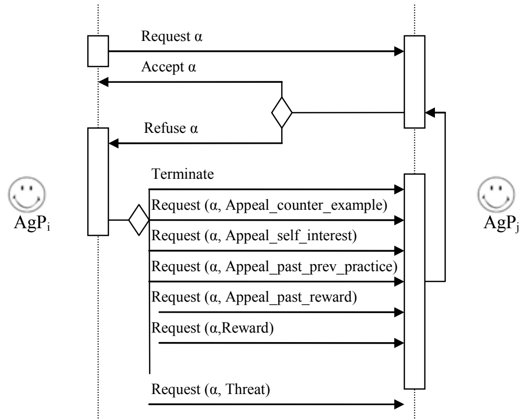



**Figure 5 - Argumentation protocol for two agents**

Argument nature and type can vary, however six types of arguments are assumed to have persuasive force in human based negotiations [Karlins and Abelson, 1970][O'Keefe, 1990] [Pruitt, 1990]: threats; promise of a future reward and appeals; appeal to past reward; appeal to counter-example; appeal to prevailing practice; and appeal to self interest. These are the arguments that agents will use to persuade each other. This selection of arguments is compatible with the power relations identified in the political model [French e Raven, 1959]: reward, coercive, referent, and legitimate.
This component will generate persuasive arguments based on the information that exists in the participant's agent knowledge base [Marreiros et al., 2006b]

### 6.4 Emotional system

The emotions that will be simulated in our system are those identified in the reviewed version of the OCC model: joy, hope, relief, pride and gratitude, like distress, fear, disappointment, remorse, anger and dislike [Marreiros et al., 2005b]. The agent emotional state (i.e. mood) is calculated in this module based on the emotions felt in past and in the other agents' mood [Santos et al., 2006].
Each participant agent has a model of the other agents, in particular the information about the other agent's mood. This model deals with incomplete information and the existence of explicit negation, following the approach described in [Neves, 1984]. Some of the properties that characterize the agent model are: gratitude debts, benevolence, credibility [Andrade et al., 2005].
Although the emotional component is based on the OCC model, with the inclusion of mood, it overcomes one of the major critics that usually is pointed out to this model: OCC model does not handle the treatment of past interactions and past emotions.

### 6.5 Implementation

Some details of the implementation of the simulator are described here. The system was developed in Open Agent Architecture (OAA), Java and Prolog. OAA is structured in order to: minimize the effort involved in the creation of new agents, that can be written in different languages and operating on diverse platforms; encourage the reuse of existing agents; and allow for dynamism and flexibility in the makeup of agent communities [OAA, URL].
Some screens of the prototype may be found in Figures 6 and 7.

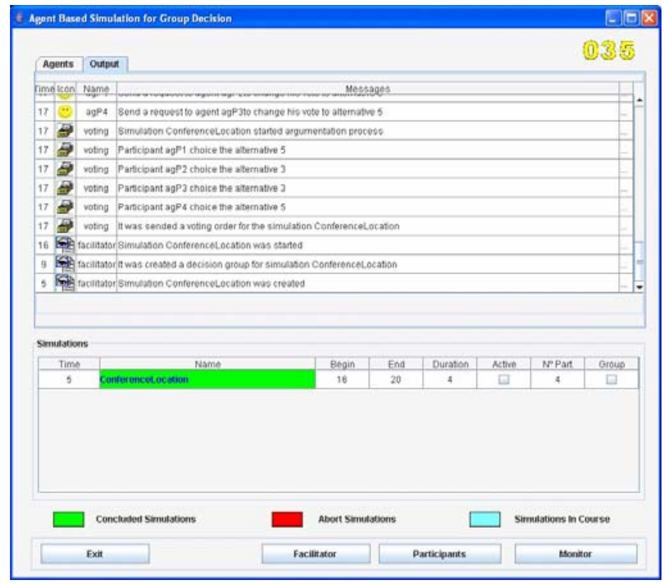



**Figure 6 – Argumentation dialogues**

Figure 6 shows an extract of the arguments exchanged by the participant agents. Once a simulation is accomplished, agents update the knowledge about the other agent's profile (e.g. agent credibility).


**Figure 7 - Community of agents**

Figure 7 shows the collection of agents that work at a particular moment in the simulator: ten participant agents, the facilitator agent (responsible for the follow-up of all simulations), the voting agent, the clock agent (OAA is not specially designed for simulation, for that reason it was necessary to introduce a clock agent to control the simulation), the *oaa_monitor* (i.e. an agent that belongs to the OAA platform, and is used to trace, debug and profile communication events for an OAA agent community) and the application agent (responsible for the communication between the community of agents and the simulator interface).

## 7 Conclusions

This work proposes a simple architecture for a ubiquitous group decision making system able to support distributed and asynchronous computation. This system will support a group of people involved in group decision making, being available in any place (e.g. at a meeting room, when using a web based tool), in different devices (e.g. computers, notebooks, PDAs) and at different time (e.g. on-line meeting, asynchronous meetings). One of the key components of this architecture is a multi-agent simulator of group decision making processes, where the agents present themselves with different emotional states, being able to deal with incomplete information, either at the representation level, or at the reasoning one [Neves, 1984]. The discussion process between group members is made through the exchange of persuasive arguments, built around the same premises stated to above. Future work includes the refinement of the architecture, as well as the improvement of the interaction between the simulator and the group members, here declared under the umbrella of the mixed initiative systems paradigm.

## References


[Andrade et al., 2005] F. Andrade, J. Neves, P. Novais, J. Machado, A. Abelha. *Legal Security and Credibility in Agent Based Virtual Enterprises.* In Collaborative Networks and Their Breeding Environments. Springer-Verlag, ISBN 0-387-28259-9, pp 501-512, 2005.

[Cesta and D'Aloisi, 1999] A. Cesta, D. D'Aloisi. *Mixedinitiative-Initiative Issues in an Agent-Based Meeting Scheduler.* User Modeling and User-Adapted Interaction, Vol. 9, No.1-2, pp 45 – 78, 1999.

[French and Raven, 1959] J. French, B. Raven. *The bases of social power*. In: Studies in social power. The University of Michigan, pp.150-167, 1959.

[Grudin, 2002] J. Grudin. *Group Dynamics and Ubiquitous computing*. Communications of the ACM, Vol 45 No 12, 2002.

[Hearst, 1999] M. Hearst. *Trends & Controversies: Mixedinitiative interaction*. IEEE Intelligent Systems, 1999.

[Karacapilidis et al., 2001] N. Karacapilidis, D. Papadias. Computer supported argumentation and collaborative decision making: The Hermes system, Information Systems, Vol. 26 No. 4 pp. 259-277, 2001.

[Karlins and Abelson, 1970] M. Karlins and H. Abelson *Persuasion: How opinions and attitudes are changed.* Springer Publishing Company, Inc, 1970.

[Marreiros et al., 2004] G. Marreiros, J.P. Sousa, C. Ramos. *WebMeeting - A Group Decision Support System for Multi-criteria Decision Problems*. International Conference on Knowledge Engineering and Decision Support, Porto, Portugal pp. 63-70, 2004

[Marreiros et al., 2005a] G. Marreiros, R. Santos, C. Ramos, J. Neves. *Agent Based Simulation for Group Formation*. 19th European Simulation Multi-Conference, Latvia pp. 521-526, 2005.

[Marreiros et al., 2005b] G. Marreiros, C. Ramos, J. Neves. *Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision.* Lecture Notes in Computer Science. Vol. 3823 pp. 41-50, 2005.

[Marreiros et al., 2006a] G. Marreiros, C. Ramos, J. Neves. *Modelling group decision meeting participants with an Agent-based approach*. Forthcoming in International Journal of Engineering Intelligent Systems, 2006

[Marreiros et al., 2006b] G. Marreiros, C. Ramos e J. Neves. *Multi-Agent Approach to Group Decision Making through Persuasive Argumentation*. 6th International conference on argumentation. Amesterdam, 2006.

[Neves, 1984] Neves, J. *A Logic Interpreter to Handle Time and Negation in Logic Data Bases*. Proceedings of the ACM'84 Anual Conference - The fifth Generation Challenge. San Francisco, California, USA.

[OAA, URL] *The Open Agent Architecture™*. www.ai.sri.com/~oaa/.

[O'Keefe, 1990] D. O'Keefe. *Persuasion: Theory and Research*. SAGE Publications, 1990.

[Pruitt, 1981] D. Pruitt. *Negotiation Behavior. Academic Press*, New York, N.Y., 1981.

[Salancik, 1977] G. R. Salancik, J. Pfeffer. *Who Gets Power – And how they hold on to it- A Strategic Contingency Model of Power*. Organizational Dynamics, Vol 5 No 3, pp 3-20, 1977.

[Santos et al., 2006] R. Santos, G. Marreiros, C. Ramos, J. Neves, J. Bulas-Cruz. *Multi-agent Approach for Ubiquitous Group Decision Support Involving Emotions*. Ma et al. (Eds.) Lectures Notes for Computer Science 4159, pp. 1174 – 1185, 2006.

[Shneiderman and Maes, 1997] B. Shneiderman, P. Maes. *Direct Manipulation vs. Interface Agents*. Interactions, Vol. 4, No 5, 1997.

[Simon, 1960] H. Simon. *The New Science of Management Decision*. Harper and Row, New York, 1960.

[Weiser, 1991] M. Weiser. *The Computer for the Twenty-First Century*. Scientific American, 1991.

[Zachary and Ryder, 1997] W. Zachary, J. Ryder. *Decision Support Systems: Integrating Decision Aiding and Decision Training*. In: Handbook Of Human-Computer Interaction, pp. 1235-1258. The Netherlands: Elsevier Science, 1997.

# What About Tests In Smart Environments?
# On Possible Problems With Common Sense In Ambient Intelligence

**Rafal Rzepka and Kenji Araki**
Language Media Laboratory
Graduate School of Information Science and Technology
Hokkaido University, Japan

## Abstract

Ambient Intelligence seems to be a perfect field to evaluate common sense reasoning algorithms as environments full of smart transmitters and receivers could resemble the real world where human senses the surroundings and processes input to use it in an intelligent way. But is it real enough? Will it be really easier to use common sense processing to predict what somebody's behavior means? By simple experiments with automatically created pseudo-Schankian Scripts we will show that naturalness of human action varies strongly - probably depending on evaluator's personal background, experiences and imagination. To limit the contextual situation we created virtual house by describing it with words for actors, places, tools, objects and motions common for such a place. Then we made the system check the commonness of actions randomly performed in this environment and see if an action is significantly against common sense or not. We will show that evaluation of naturalness becomes clearer and fairer when context is set but we will also call into question that ambient technologies can describe the context well while they could do just opposite - they could limit it. In this paper we also describe other problems of common sense when used in AmI and suggest some possible solutions.

## 1 Introduction

It has been almost a half of century since Mortimer Taube wrote the first anti-AI book, ”Computers and Common Sense: The Myth of Thinking Machines”[Taube, 1961]. Although it did not receive much attention, it clearly shows that from the very beginning of less or more advanced technologies people doubted that machines could process the vastness of knowledge called ”Common Sense” (in this paper often abbreviated to CS) which we gather from the first moments of our lives. It has been always called ”the Holy Grail of AI” but none of the developed systems could be claimed fully successful. The turn of the century brought us repositories as CyC being built by specialists, OpenMind CommonSense being created by mass collaboration of Web users[Singh, 2002] or ThoughtTreasure built mostly by one person[Mueler, 1998] and lexical databases like WordNet[Fellbaum, 1998]. Gathering common sense knowledge entries became main trend in the field of CS and we also concentrated on this task, though we try to achieve more automatic acquisition methods. However, the more experiments we performed, the more they made us doubtful about the main-stream ”collecting approach”. The main problem - lack of contextual flexibility which brings the lack of universality - is still being ignored. We can collect plenty of commonsensical data about some objects and their properties but the usability of this knowledge will change in different situations or with different actors. Is Common Sense really universal knowledge? It surely changes its commonness depending on the situation and we showed it in a very simple experiment. Evaluating such knowledge without context does not make much sense but is it the Ambient Intelligence where Common Sense researchers could prove their systems work well? When Locke[Locke, 1690] proposed the term “common sense” he meant a sense which combines input form other senses into “human understanding”. At first sight Ambient Intelligence seems to give us what common sense researches lack most often nowadays - real applications with real input from real environment. From the arificial senses which could create Lockian Common Sense. However, a kitchen stuffed with sensors, RFID tags / readers and intelligent fridges still does not understand what is really going on inside it except predicted input combinations. Even if everyone wears “smart clothes” and the system gets hold of who does what - it will not necessarily be able to recognize what such behavior pattern could possibly mean and to what consequences it could lead. The problem of preprogramming every possible scenario will remain and Common Sense in Ambient Intelligence will be common only for a given ambient even if sharing data from similar devices of other users. In the next section we describe above mention experiment demonstrating problems of common sense evaluation and then we will present a model scenario in a fashion of famous ISTAG's scenarios for 2010[Ducatel et al., 2001] explaining problems and our ideas for solving them.

## 2 Confirming Some Obvious Things About Common Sense

Before we go to the crux of the matter, we will briefly describe the experiment which confirms facts most of us feel to be obvious but which are widely ignored by “Knowledge Collection-Oriented” main-stream of current CS research.

### 2.1 Automatic Minimal Script Retrieval

In our research we use several languages from different language groups but here we will introduce results of experiments performed with Japanese (to implement our method to some language it is enough to change the grammar forms for retrievals, for example to use English prepositions instead of Japanese particles). The system analyses one pattern sentences describing simple actions in form of one noun, one particle and one verb and tries to retrieve most common previous and following actions imitating an idea of Schankian scripts[Schank et al., 1998]. A sentence (Place)-Noun (+place particle) Noun (+object particle) Verb (called here ”motion”) is inputted and such combination is being confronted with the Web. Then the neighbors are retrieved for extracting previous and next actions (for faster retrieving we use web-based bigram and trigram dictionaries of nouns, particles and verb combinations). This algorithm allows to automatically create minimal scripts, for example ”Give birth to a child → take care of (it) → do best” or ”wait for friends → have fun → go back home”. The next step was to evaluate these retrievals.

### 2.2 Experiments and Results

We picked results of 34 inputs made by users. Wide range of different words was chosen as we wanted to keep the broadness of common sense knowledge - any kind of action was allowed and only thing we set was telling users that these action would be faced and analyzed by house-helping devices. We presented 10 judges an evaluation list where every input was also divided into ”place + motion”, ”object + motion”. Additionally possible previous and following actions were proposed. Every such output was also labeled by the system as ”natural” or ”unnatural” and a user was to mark such label in a scale from 0 to 5 in two levels: how precise the label is and how useful it would be for an intelligent house. Previous and next actions were evaluated for naturalness instead of preciseness. The average marks given by ten judges for the preciseness of automatic naturalness labeling were 66.4% for the whole action sentence, 85% for object-verb form and (quite surprisingly) only 57% for ”place+verb” form. The usefulness for these three forms was evaluated similarly : 61.2%, 80.2% and 54%. Previous actions were evaluated natural in 67.8% and useful in 61.2% cases while next actions were evaluated natural in 55% and useful in only 45.5% cases. After taking a closer look to the evaluation data, it became clear that what is *natural* and *useful* depends deeply on individual factors which are hard to estimate - probably from own experiences to richer or poorer imagination. Figure 1 shows an example of opinion dispersion among judges.

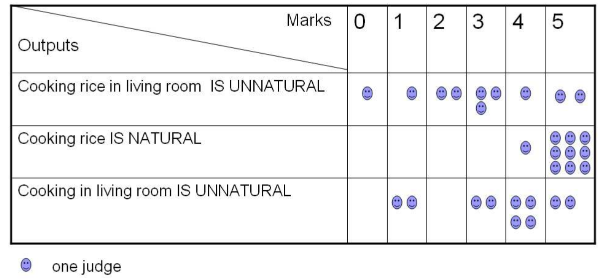



Figure 1: Example of opinion dispersion while evaluating system’s measurement of common action naturalness

## 3 When “Cooking Rice” Becomes Uncommon?

It can be said that thinking too much about commonsensical happenings distorts the commonness of these happenings. Deep thinking about shallow things may be one of the key problems in this case. As seen in the example in Figure 1, *cooking rice* is of course natural for every judge (output was judged by native Japanese) but ideas about naturalness of cooking and rice varies heavily depending on the place. Very wide expressions are easy to be agreed on, however more details triggers some psychological mechanisms and the ideas of commonness start to vary. To keep everyone’s imagination focused and still be able to increase the number of details, we decided to limit the user’s imagination by preparing more detailed experiment environment.

### 3.1 Limiting User’s Imagination

As we are thinking about implementing commonsensical knowledge into a intelligent system helping at home, we decided to set a simulation environment at the house with limited number of actors (4, e.g. child, dog) performing limited ”motions” (24, e.g. to eat, to open, to bring) with limited objects (32, e.g. knife, pot, trash) in limited places (8, e.g. toilet, balcony, shelf).

### 3.2 Experiments in More Specific Ambient

Random actions were created in following form: actor (subject particle) - place (place particle) - object (object particle) - verb (”motion”). When motions as ”giving” were involved, ”place” was being replaced with another ”actor” chosen randomly from the set. Then the system was checking the Web frequency for every combination of actor, place, object and motion. Depending on these results, the system was 1) approving the action as natural (15% of cases) 2) suggesting which part is unnatural (55%) or 3) claiming that action has too many uncommon parts that is not worthy analyzing (30%). For instance ”mum gives a child spaghetti” was evaluated natural, ”a kid puts back a soap to the balcony” drew suggestion that ”soap” is unnatural part of this action and ”a robot opens a rice-cooker at the toilet” was rejected as being ”too strange”. Then we made 14 judges familiar with the environment and actors evaluate 40 outputs. The average evaluation estimated in the middle as in the experiment described in the previous section but what we were interested in was the opinion dispersion. In comparison to the previous experiment, the number of asymmetric or flat evaluations (where judges opinions varied heavily) dropped from 57% (only the whole actions were considered) to 15%. Most of the dispersed opinions was on actions where robot was an actor or balcony was a place.

### 3.3 Conclusions From This Trial

Three conclusions may be drawn:

- Very obvious truths are evaluated natural and useful but the context may make them unclear, less useful or even untrue;
- Common sense knowledge should be gathered together with context or, as is our choice, should be retrieved on spot for given context;
- Technologies allowing to recognize given situation of real life should be a good place to test CS algorithms.

As it is easy to guess, the last conclusion led us to Ambient Intelligence. Common sense knowledge system experiments should be performed in environments where programs are tested in the same situations with the same set of details. Although we specialize in retrieving common sense knowledge from the Internet, we felt that Ambient Intelligence could become very convenient training ground. Surely a discussion on how to stimulate progress in interdisciplinary field of CS reasoning should be done together with AmI specialists, hoewver the open question is - will AmI lead to achieving Holy Grail of AI? In our opinion it will help but we have to be really careful. We need as much sensor input as possible but we must remember how specific is the aim of AmI. Gathering knowledge about one user, one group or one environment will tend to build semantic dependencies concentrating on market-driven choices of input. The holistic approach for learning CS from all sensors data will be simply impossible just because companies and users will naturally protect their data. One of the possible solutions could be sensor input simulation which basic ideas were already introduced back in 2003[Rzepka et al., 2003]. Before explaining our ideas on this matter, we will present a short scenario showing some of the possible problems.

## 4 Scenario 5: 'Susan' - Housewife

After not so long but sound sleep[1], Susan wakes up at her bedroom. Her Private DayScheduler (PDS) had no reason to wake her up as it didn't recognize anything in "TV-guide" which could be interesting for her. If it did, the program would be recorded and the day rescheduling would be performed depending on the tasks planned for this particular day. Her kids have gone to school already after being waken up by their PDS - they are big enough to eat what Smart Kitchen suggested Susan yesterday. She cares much about what her kids eat everyday and Smart Kitchen always helps her to choose new healthy products which people recommend in their blogs. But today Susan has one problem which is her terrible headache. The Smart Kitchen discovered her unusual state but had no idea that is because of drinking too much gin and tonic on the previous night. Susan asks for a pain-killer and it soon helps with her headache, however she feels heavy raising herself up and taking some milk from the fridge. Her movement caused the preprogrammed reaction - the coffee maker switched on. For Susan the scent of coffee during the hangover is the worst thing she can imagine, so she feels bad again and goes out to take a deeper breath of fresh air. Her PDS reacts immediately assuming that the shopping planned for the afternoon is now rescheduled. Within a few seconds she hears the beep and a little screen recommends her the nearest grocery. A bar code for all the things which must be bought appears - now it is enough to bring it to the store where special assistant will suggest the recommended equivalents. It (assistant is a little tool) already knows that Susan loves novelties even if the price is a bit higher than usual. But she is not in mood for shopping. She enters the park and sits on the bench dying for some rest. PDS beeps constantly trying to show her way to the grocery, so she switches it off. She knows that the security system calls her husband and the nearest police station if her PDS goes off in a dangerous place. What she does not know is that this park has been just listed on the Metropolian List of Dangerous Places. Somebody had his intelligent clothes stolen here last night...

[1]Similarity to the ISTAG scenarios is intentional however the purpose of writing it is a bit different.

## 5 Sensors Need Help As Well As We Do

The simplest way to avoid some of the situations described above would be a natural language interface which asks what is going on with Susan and understands the possible consequences of having hangover. However, people are not always in mood to explain a machine their feelings. It would be much easier to guess what is happening if commonsensical knowledge as "too much gin and tonic in the evening = possible hangover in the morning" was taken into consideration. Of course questions as "how much is too much" or "what does it mean in numbers that human becomes slower during the hangover" remain unanswered until Susan repeats her "drinking" and "feeling unwell" actions several times. But what does it mean that she drinks every night? Why her busy husband is not informed about it but he has a call about her resting on the park bench? Analysing human behavior just upon partial input will always be dangerous but the more possible conclusions are drawn, the less likely it is to be sure about the situation. The machine needs other sources of confirmation to recognize danger correctly or to avoid turning user's life into a nightmare where he or she is afraid to do anything unusual just to avoid possible problems with gadgets designed to make their life easy. Socio-political factors are not the only issue here, we need solutions already on a very prosaic level. In next subsections we introduce some of our thoughts which could bring some help or at least pass some hints to AmI developers.

### 5.1 Digging Treasures In the Garbage Can - Common Sense From the Web

As we already mentioned above, we retrieve common behavior patterns from the WWW. This is one of the practical

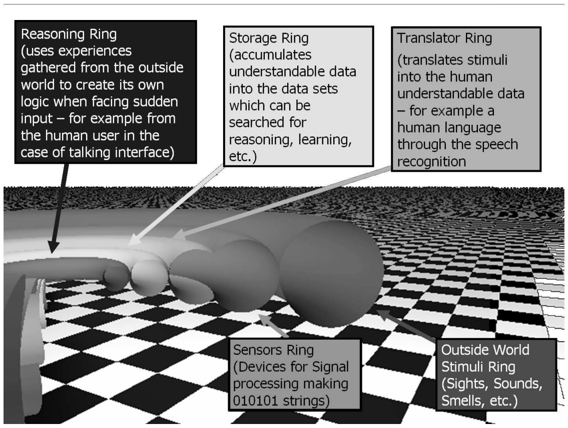



Figure 2: Representation of A.I. in a shape of five rings showing flat dependency between outside layer of real-world and inside layer of processing outside data to e.g. learn commonsense knowledge.

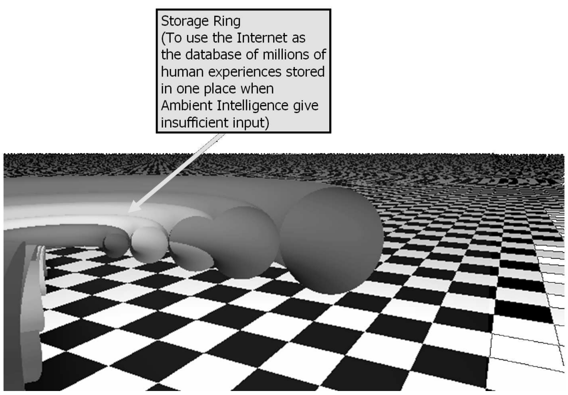



Figure 3: Our idea of supplementing the missing data by text mining the Web in order to achieve missing or ignored data.

examples of our philosophy which could be represented as shown in Figures 2 and 3.

The main assumption is based on observations nicely described in James Surowiecki's bestseller "The Wisdom of Crowds"[Surowiecki, 2004] and says that the ordinary people's knowledge is almost always correct especially if the specialist opinions are mixed with jokes or very silly opinions. Receiving a global input from all the devices of AmI could help us a lot - but as we mentioned before - the business would not let it happen and even if open source version would arise - the users would probably avoid giving away their private data. AmI will need the support of the Web - if not semantically, then at least as a model to mine behaviors just as web-mining analyses browsing patterns now. However, our claim is to retrieve lexical structure in order to build semantic structures to cooperate with any input from the outside world as any of such input can be represented in natural language. It is the point when our approach of "Crowds-supervised learning" starts. The next step, or rather the step to be made simultaneously, is an universal theory (or rather assumption) for all humans behaviors.

## 6 AmI And Global Ethics

We presented our idea of Crowds being ethically correct during the first Machine Ethics Session of AAAI Fall Symposium in 2005[Rzepka, 2005] although we were developing algorithms telling Good from Bad years ago[Rzepka, 2002]. Last year we presented a supplement proposing to calculate Bentham's Felicific Calculus[2] by using our techniques[Rzepka, 2006]. If we succeed, the program should be able to measure following variables (vectors):

- Intensity - how strong is user's feeling? "Losing a parent" and "losing a watch" is a different world but it is hard to conclude from the lexical surface only - affective computing must be performed in order to measure the intensity of possible human reactions.
- Duration - how long an emotion lasts? What is the difference between "one hour party" and "whole night party"? Will it be better for the user if I set up one minute of his favorite music or one hour?
- Certainty or Uncertainty - how sure can I be about being punished or awarded? Buying will assure human the pleasure but counting on somebody giving me something leaves the risk that I will not feel the expected amount of pleasure because I will not get what user wanted.
- Propinquity or Remoteness - how close is the positive/negative emotion from the user? What is the difference if his close friends has won something than a case where such thing happens to my neighbor.
- Fecundity - the probability that one sensation will be followed by the same kind of sensation. Will a good state of something remain good or is it going to worsen? Will my decision keep my user in good mood?
- Purity - the probability that one sensation of the opposite kind will be followed by sensations of opposite kind. By default the machine should always try to preserve user's good emotional state and do everything to avoid worsening it.
- Extent - the number of users to whom my decision or some happening extends. For instance if the machine's decision affects one coworker of its user instead of ten, it is preferable choice.

Algorithms based on utilitarian ideas will probably have more acceptance within AmI communities than in other fields but still there will be a large margin for discussion on hedonistic

[2] Jeremy Bentham's famous quote "Nature has placed mankind under the governance of two sovereign masters, pain, and pleasure. It is for them alone to point out what we ought to do, as well as to determine what we shall do?"[Bentham, 1789] is controversial, we agree, but probably it is easier to implement than Kantian ethics and we will avoid the discussion which philosophy fi ts us most!

background to all human behaviors. But on the other hand using evaluation techniques on common happenings and objects could lead to safety valve letting the system recognize danger even if given case was not inputted beforehand. Research on retrieving opinions about e.g. products by using specific language features could help on retrieving the emotional load of human behaviors and their consequences.

## 7 Final Conclusions

By setting different environments giving different set of inputs (RFID or any other kind of sensors) we would achieve a method for equal evaluation for Common Sense Reasoning specialists, their databases and algorithms. In our opinion we are still far away from universality of intelligent systems using Common Sense because its researchers have problems with comparing their systems. Collected knowledge should be tried not only in single, still limited applications but in constantly changing real world simulations. However, even if bumping into reality seems to be a great method for testing CS systems, it does not mean that Ambient Intelligence will solve problems of digitalizing Common Sense. In this paper we tried to show remaining problems by creating an example scenario and suggested several ideas which might be helpful for AmI. These ideas could be described in two bigger groups - "Web Support for Knowledge from Sensors" and "Crowds Ethics for Intelligence Systems". Our future work is to continue developing our methods testing them in smart environments whenever it is possible. We are also open for any kind of cooperation as Japan has implemented several AmI technologies already but our knowledge on the field is insufficient to prepare professional tests.

## References


[Taube, 1961] Mortimer Taube. *Computers and Common Sense: The Myth of Thinking Machines* New York, Columbia University Press, 1961.

[Mueler, 1998] Erik T. Mueller. *Natural language processing with ThoughtTreasure* New York, Signiform, 1998.

[Singh, 2002] Push Singh. The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Palo Alto, CA: AAAI, 2002.

[Fellbaum, 1998] Christiane Fellbaum (ed.). *Wordnet: An Electronic Lexical Database* Cambridge, Massachussets, USA: The MIT Press, 1998.

[Locke, 1690] John Locke *An Essey Concerning Human Understanding* www.gutenberg.org, orignal 1690.

[Schank et al., 1998] Roger Schank and Robert Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures (Chap. 1-3)* L. Erlbaum, Hillsdale, NJ, 1977.

[Matsumoto et al., 1999] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda and Masayuki Asahara. *Japanese morphological analysis system ChaSen version 2.0 manual 2nd edition*

[Ducatel et al., 2001] Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten and Jean-Claude Burgelman. *Scenarios for ambient intelligence 2010* ISTAG report, European Commission, Institute for Prospective Technological Studies, Seville, 2001.

[Rzepka et al., 2003] Rafal Rzepka, Kenji Araki and Koji Tochinai. *Five Senses Input Simulation Based on Web Resources* Technical Report of Language Engineering Community Meeting (Kotoba kougaku kenkyuu-kai), Kyoto, Japan, 2003.

[Surowiecki, 2004] James Surowiecki. *The Wisdom of Crowds* Anchor Books, New York, 2004.

[Rzepka, 2005] Rafal Rzepka and Kenji Araki. *What Statistics Could Do for Ethics? - The Idea of Common Sense Processing Based Safety Valve* Machine Ethics, Papers from AAAI Fall Symposium, Technical Report FS-05-06, pp. 85-87, Arlington, USA, 2005.

[Rzepka, 2002] Rafal Rzepka, Kenji Araki and Koji Tochinai. *Is It Out There? The Perspectives of Emotional Information Retrieval from the Internet Resources* Proceedings of the IASTED Artificial Intelligence and Applications Conference, pp. 22-27, Malaga, Spain, 2002.

[Rzepka, 2006] Rafal Rzepka and Kenji Araki. *Emotion-Driven Common Sense Retrieval for More Natural Language Processing* 12th Annual Conference of The Association for Natural Language Processing, "Emotions, Evaluation, Language and Attitudes" Workshop Proceedings, pp.49-52, Yokohama, 2006.

[Bentham, 1789] Jeremy Bentham. *An Introduction to the Principles of Morals and Legislations* (Chap. 4), London, 1789.