WORKSHOP PROCEEDINGS

Artificial Intelligence Methods for Ambient Intelligence

at the

European Conference on Ambient Intelligence
AmI'07

November 7, 2007
Darmstadt, Germany,

Editors: Ralph Bergmann, Klaus-Dieter Althoff,
Ulrich Furbach, Klaus Schmid

Preface

We are pleased to present the workshop proceedings of the workshop Artificial Intelligence Methods for Ambient Intelligence, which takes place as part of the European Conference on Ambient Intelligence.

Ambient Intelligence (AmI) is the vision of our future environment. It will be surrounded by various kinds of interfaces supported by computing and networking technology providing an intelligent, seamless and non-obtrusive assistance to humans. The ambient environment will be aware of the presence and identity of the humans, it will be able to communicate in multi-modal form and to anticipate the humans' goals and needs in order to provide best possible assistance to them. This broad vision addresses all areas of human life, such as home, work, health care, travel and leisure activities. Within the interdisciplinary research aiming at approaching this vision, Artificial Intelligence (AI) provides a rich set of methods for implementing the "intelligence bit" of the AmI vision. Speech recognition, image interpretation, learning (from user interaction), reasoning (about users' goals and intensions) and planning (appropriate user interaction) are core features of AmI to which AI can contribute significantly. The goal of this workshop is to make a step forward towards a common understanding of how AI can contribute to the AmI vision and how to align AI research with it.

Program Committee

Juan Carlos August, University of Ulster, U.K.
Michael Berger, Siemens, Germany
Iryna Gurevych, TU Darmstadt, Germany
Otthein Herzog, Uni Bremen, Germany
Thomas Kirste, Uni Rostock, Germany
Gerd Kortuem, Uni Lancaster, U.K.
Ramon López de Mántaras, IIIA, Barcelona
Max Mühlhäuser, TU Darmstadt, Germany
Jörg Müller, TU Clausthal, Germany
Markus Nick, Fraunhofer IESE, Germany
Mihaela Ulieru, The University of New Brunswick, Canada
Reiner Wichert, Fraunhofer IGD, Germany
Volker Wulf, Uni Siegen, Germany

Workshop Organizers

Ralph Bergmann, University of Trier, Germany
Klaus-Dieter Althoff, University of Hildesheim, Germany
Ulrich Furbach, University of Koblenz-Landau, Germany
Klaus Schmid, University of Hildesheim, Germany

Table of Contents

AI Methods for Smart Environments

A Case Study on Team Assistance in Smart Meeting Rooms

Martin Giersich, Thomas Heider, Thomas Kirste

Department of Computer Science, Rostock University
Albert-Einstein-Straße 21
18059 Rostock, Germany
first.last@uni-rostock.de

Abstract. Ubiquitous computing aims for the realisation of environments that assist users autonomously and proactively in a non-distractive manner. Therefore smart environment infrastructures need to be able to identify users needs (*intention recognition*) and to plan an appropriate assisting strategy (*strategy generation*) without explicit user interaction. In our two-stage approach we address inferring the intention of a team of users during a meeting within a smart multiple display environment and the system decision process – what information to present on which display – on the strategy generation level.

1 Introduction

A central requirement for an assistance architecture for a smart meeting room is that it should support technical infrastructures that are built from individual components in an ad hoc fashion. Our solution approach is a two-stage design, where at the first stage the system components recognize the intention of a team of users, and at the second stage, the system components jointly generate a strategy that fulfills the needs of the team. In this paper we report the results of our current research and the ongoing evaluation. Part one of the paper presents results in intention analysis and part two represents strategy generation and evaluation. At this time we evaluated the parts separately. An evaluation of the complete integrated system will be matter of future work.

Intention recognition becomes a challenge, especially if multiple users are observed by noisy heterogenous sensors. We propose a *team behavior model* based on hierarchical dynamic Bayesian network (DBN) for inferring the current task and activity of a *team* of users. Given (noisy and intermittent) sensor readings of the team members' positions in a meeting room, we are interested in inferring the team's current objective.

A simulation data evaluation of our particle filter based team behavior model shows reasonable inference accuracy and speed for our implemention and demonstrates how additional unreliable knowledge about the meeting agenda improves prediction accuracy and speed. Here, we claim that even unreliable agendas improve intention recognition in smart environments for a compliant team behavior without sacrificing recognition accuracy for the non-compliant case.

We propose to cast the **Strategy generation** problem as an optimization task. As example problem we use the document-display mapping question, which is what to present on what display in a multi-user, multi-display environment. We suggest the definition of an *explicit global quality measure* to achieve coherent ensemble behavior for a team of multiple users with (maybe) diverging interests.

The evaluation of this part shows, that an automated document display mapping based on an explicit global quality measure leads to coherent ensemble behavior and is at least as effective as conventional manual assignment, while at the same time significantly reducing the number of required interactions. This claims are based on user performance data collected in the scope of a comparsion study.

2 Intention Recognition

Especially for the intention recognition used in our prototype smart meeting room we studied whether incomplete and unreliable (i.e., sometimes misleading) knowledge about the needs of a team of users (agenda) can be used to improve the quality of intention recognition. Specifically, we were interested in the usefulness of an unreliable agenda for improving the recognition of team activities during a meeting. Based on Bayesian filtering and an explicit probabilistic team behavior model we have carried on a simulation study that allowed us to answer the following questions:

- How accurate and how fast can we predict team behavior with an agenda assumption and history knowledge?
- What influence do deviations of the team from the planned agenda assumption have on prediction quality (i.e., does a wrong agenda degrade the quality of intention recognition)?
- How flexible does an agenda assumption need to be in order to optimally predict team behavior?

We chose simulation of data rather than real world data as this enabled us to configure the probability distribution of the sensor readings. We used Gaussian and Cauchy distributed sensor readings with a variety of different parameter settings to examine the influence of the sensor model on the prediction quality.

Team Behavior Model – Bayesian Filtering for identifying a user's current task has been successfully used in several projects that aimed at supporting user activities in classrooms, meeting rooms, and office environments [1–3]. Here, *dynamic Bayesian networks* (DBNs) were investigated increasingly for modeling a user's activities [4, 5]. In our own work, we looked at using DBNs for inferring the current task and actions of a *team* of users. Given (noisy and intermittent)

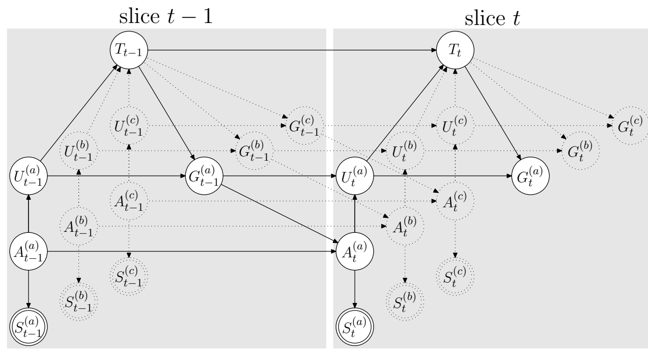


Fig. 1. Two-sliced dynamic Bayesian network (DBN) modeling team intention inference. It shows the intra-slice dependencies between observable (double-contoured) and hidden variables, as well as the inter-slice dependencies between consecutive states.

sensor readings of the team members' positions in a meeting room, we were interested in inferring the team's current objective – such as having a presentation delivered by a specific team member, a moderated brainstorming, a round table discussion, a break, or the end of the meeting.

The basic structure of the DBN we propose for modeling the activities of such a team is given in Figure 1. With this DBN we try to model the behavior of a team of three users during a meeting. In order to exploit agenda information, we need a DBN structure that is able to incorporate an explicit agenda, and that represents the negotiation process between the team and its members during activity selection. At the top level, the team node T_t represents the current team intention. The team's intention at time t depends on what the team has already achieved (T at time $t-1$, T_{t-1}), and what the users i are currently trying to achieve (the $U_t^{(i)}$-nodes, $i \in \{a, b, c\}$). The $G_t^{(i)}$ nodes represent the new individual assignments if the team T will adopt a new intention. So at each time slice, the team looks at what the users have achieved so far and then decides what the users should do next. What the user is doing at time t depends on his previous action (e.g., the user's current position and velocity) and assignment – $A_{t-1}^{(i)}$ and $G_{t-1}^{(i)}$. Finally, the sensor observations of user i at time t – the nodes $S_t^{(i)}$ – depend on the user's activities at that time.

Note that these sensor nodes are the *only* observable nodes in our model: we estimate the team's negotiations from the observable behavior of the team members. Once a probabilistic model is available, it allows us to infer user and team intentions.

Experimental Design and Results – Clearly, agenda information should improve the quality of team intention recognition. However, as soon as a team deviates from the a-priori agenda, recognition quality may drop: The recognizer may be led to wrong conclusions by misleading a-priori information that potentially defeat any benefit. Objective of our evaluation has been to investigate,

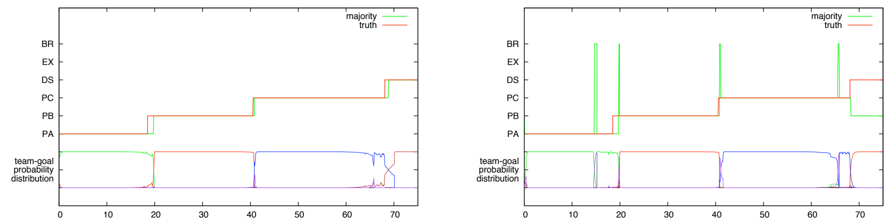


Fig. 2. Inference of a $\langle \mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D} \rangle$ truth from Cauchy distributed sensor data (delay 0.25, error 10.0) with the trackers $T_{.8}$ (left) and $T_{uniform}$ (right).

whether a-priori agenda information can be used to improve recognition quality in case the team complies to the agenda, *without* sacrificing recognition quality in case of non-compliance with the agenda. We were interested in two main questions:

(a) How reliable is agenda based recognition in case of compliance and non-compliance, compared to an agenda-less tracking?
(b) How fast will an agenda based recognizer identify a change in the team objective for these cases?

To analyze the effect of an agenda on reliability and speed of intention recognition in case of compliance and non-compliance we chose three different conference sequences (one compliant, two non-compliant). Further we used four different parameter settings for the sensors. In two settings we used sensor data that is Gaussian distributed. The two other settings sensor data followed a Cauchy distribution. The settings for each distribution differed in delay between consecutive sensor readings and sensor error.

For the evaluation of recognition accuracy, we used four different models for a-priori agenda information – a random model where every activity has the same probability and history is not tracked ($T_{uniform}$) and three models with different start probabilities for user A $\{.6, .8, .95\}$ and the other users respectively ($T_{.6}$, $T_{.8}$, $T_{.95}$). For every tracker model six runs were logged. The illustration of two typical representative of model $T_{.8}$ and model $T_{uniform}$ simulation runs in Figure 2 shows that the main uncertainty about the teams objective pervails during the phase of an objective shift. The left picture shows the advantage of agenda knowledge. For instance the objective shift from *B Presents* (PB) to *C Presents* (PC) around time slice 40 is recognized faster and more reliable. Further it shows that agenda knowledge leads to less misinterpretation of sensor readings. So the overall error rate shrinks. Figure 3 shows solid recognition also for non-compliant cases. Here, tracked with model $T_{.8}$.

The averages over 6 simulation runs for 48 different parameter settings give an delay between true objective shift of the team and the recognition of this shift of $7.36sec$ for $T_{.8}$ versus $10.95sec$ for $T_{uniform}$. The average intention recognition reliability for the best model $T_{.8}$ was measured with 91.16% correct versus 83.1%

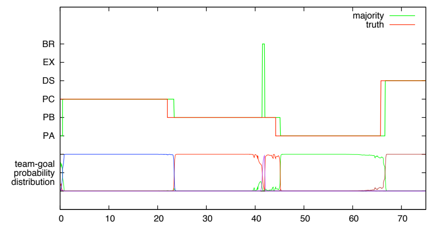


Fig. 3. Inference of the non-compliant truth $\langle \mathsf{C}, \mathsf{B}, \mathsf{A}, \mathsf{D} \rangle$ from Cauchy distributed sensor data (delay 0.25, error 10.0) with the trackers $T_{.8}$.

for the uniform model. Comparison of the reliability values for $T_{.8}$ and $T_{uniform}$ gives the most important result of this study:

> It is possible to improve the recognition accuracy for the compliant case by using an agenda, *without* sacrificing recognition accuracy for the non-compliant case.

Therefore, it always pays to include available a-priori agenda information in the recognition system, even if the correlation between the agenda sequence and the true activity sequence is not very strong.

However, it is important to assign a suitable probability to the agenda's preferred sequence. If this value is too high (e.g., .95), the agenda becomes too rigid: it will tend to assume that the team follows the agenda, even if the sensor data does tell a different story. On the other hand, further increasing the looseness of the agenda (e.g., to .6) does not improve the recognition of the non-compliant action sequences. We suspect that unnecessary looseness will eventually degrade recognition capability, but we have not observed this in our data.

Finally, simulation results show that an agenda reduces the delay, specifically for the later team actions. (Clearly, the agenda will not reconsider items already worked off, an aspect favorably reducing the degrees of freedom in comparison to $T_{uniform}$.)

3 Strategy Generation

Multi-display environments support collaborative problem solving and teamwork by providing multiple display surfaces for presenting information [6, 7]. One difficulty here is the *display mapping problem* – that is, deciding which information to present on what display in order to optimally satisfy the users' needs for information. Current approaches for controlling multi-display environments rely on manual assignment [8, 9], using a suitable interactive interface and resolving conflicts by social protocols (negotiations). However, manual display assignment has to cope with the following problems:

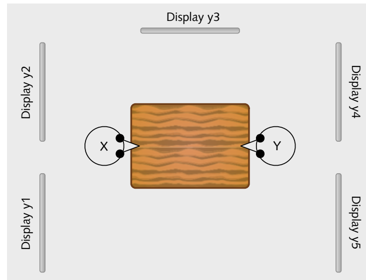


Fig. 4. Experimental setup for strategy generation comparsion study

– **Interest conflicts** between users might be solved faster by computer supported negotiation mechanism: It was observed that social protocols do not always suffice for coordinating the use of shared resources [10].
– The need for **dynamic realignment** of display mapping is caused by topic changes in the user population: In this situation, the user's focus of attention will be on the changing topic rather than on convincing the display infrastructure to change the topic.

So, an automatic display assignment might be helpful in multiple display environments, specifically in multi-user settings. However, to our knowledge, it is not known if suitable automatic assignment heuristics can be found. This is the question we want to answer.

Display Mapping Quality Measure – A *display mapping* is a function m, which assigns documents to sets of displays. For a given document d, $m(d)$ gives the set of displays document d is assigned to. In order for automatic display mapping to be successful it is necessary to identify a well-defined quality measure that sufficiently captures the users needs. Clearly, at least the following aspects are reasonable:

Spatial Layout: For documents of high importance to a user, displays should be preferred that provide a good visibility for the user. Formally, this critierion for m can be defined as

$$q_s(m) = \sum_{\substack{u \in U \\ d \in D}} impt(d,u) * \max_{y \in m(d)} vis(y,u) \tag{1}$$

where $impt(d,u) \in [0\,..\,1]$ denotes the importance of the document d to a user u, and $vis(y,u) \in [0\,..\,1]$ the *visibility* of display y by user u. If a document is assigned to multiple displays, only the best one ("primary display") for a given user is considered when computing the quality for this user (this is the "max *vis*" term). Note, that deriving a reliable estimation of *impt* in general may be a substantial challenge. We think that additional informations available from intention recognition can be used as a surrogate (such as agenda listings, team members roles and associated documents, etc.).

Temporal Continuity: When considering a display for a document, the system should prefer already existing assignments: Documents should not unnecessarily change their place. A relevant display shift occurs between two mappings, if a user's primary display for a document changes. We then try to minimize these shifts relative to the document's importance. Based on these criteria, we have developed an algorithm that is able to automatically compute a display mapping for a set of users and documents (see [11]).

Experimental Design – The objective of our evaluation experiment was to measured the impact of manual vs. automatic display assignment on the performance of a team in solving a semi-cooperative task. In such tasks, the need of cooperation and joint use of information is not evident from the start, but rather arises while working on the task. We think that this kind of aspect pertains to many team processes.

Two-person teams had to solve a semi-cooperative set of comparison tasks as fast as possible. The two team members, X and Y, were given different agendas, each containing the description of an individual comparison. For X the task was to do a simple letter comparison of two documents `A` and `B`, for Y the task was to compare `A` and `C`. In addition, X and Y had to report time information and a random key from another document `Time`. The seemingly unrelated tasks for X and Y were linked into a cooperative task through the shared documents `A` and `Time`.

Every participant was given a simple user interface for document assignment. Manually assignment of a document to a display-surface is done through simple "drag & drop". For automatic assignment, the user just associates an importance value with the documents. As the agendas and task descriptions were mutually unknown, the sharing had to be discovered through a conflict in the manual assignment group.

For each experiment, we recorded the time required for completing the task, the number of interactions and the solution correctness (percentage of letter differences found). After each task set, the subjects were asked to answer a questionnaire regarding user satisfaction. After both task sets, the subjects were asked to complete a final questionnaire regarding the comparison of automatic versus manual assignment.

24 voluntary subjects were recruited from staff members and students of the local university. The teams had to solve two sets comparison tasks in sequence. Group A had to solve the first set using automatic assignment and the second set with manual assignment. The Group M was given the tasks in reverse order. In the evaluation of the results, we will call the first set "Initial Test" and the second "After Training", respectively. (See [12] for a more detailed discussion of both experimental setup and findings.)

Findings – When the teams were using automatic assignment, the average time to complete one set of a comparison task was 4:08*min*, while they required an average time of 4:49*min* using manual assignment. The subjects needed 8.5 interactions on average with automatic and 15 interactions on average with manual

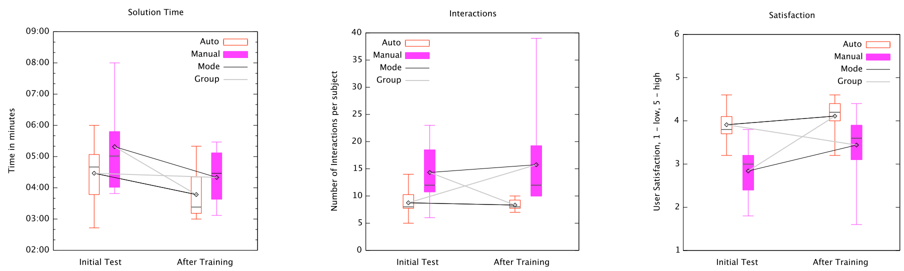


Fig. 5. Boxplots of solution time vs. mode, overall (left); interaction count vs. mode (middle); user satisfaction vs. mode (right)

assignment. This indicates that the automatic assignment is superior to manual assignment, regarding time and interactions.

An overview of the collected data is shown in the boxplots[1] in Figure 5. In these plots, "Mode" refers to the display assignment mode (manual vs. automatic). In the per-task-set plots, grey lines connect the mean values of the two consecutive task sets of a group (Group A or Group M), black lines connect consecutive task sets using the same assignment mode.

As can be seen in Figure 5, left, for both task sets the solution time is shorter when using automatic assignment. In addition, Group M was able to solve the task substantially faster in the second set (i.e., when switching from manual to automatic assignment), whereby Group A was not able to improve performance in the second set (i.e., switching from automatic to manual assignment). The number of interactions (Figure 5, middle) is smaller for the automatic method in both sets.

In the manual assignment mode, both groups initially had no idea that they needed to share documents. So they unwittingly "stole" the shared documents from each others "private" displays. It took a couple of interactions until the participants realized that they needed to cooperate and to assign some of the documents to a display visible to both users. This process of realization and negotiation was the reason for confusion and delay.

In the automatic assignment mode no such conflicts did arise as the system automatically displayed shared documents on a shared screen. If we use the number of interactions as indicator of occurred conflicts, the data shows that with the automatic mode the number of conflicts is considerably smaller than in the manual mode. A detailed survey of the log files showed that documents which had to be shared, very frequently were reassigned in the manual mode. This proves the presumption that resolving conflicts by social negotiation is – in some situations – inferior to a computer supported negotiation, which can be solved by an automatic assignment using a global quality function such as q.

[1] These boxplots show the minimum and maximum values, the 25% and 75% percentiles, the median (horizontal bar inside the box), and the mean (small circle inside the box).

For assessing user satisfaction, we used parts of the technology acceptance model (TAM)[13]. We included the following items, each to be answered on a scale from 1 (strongly disagree) to 5 (strongly agree):

> *The system is easy to use. – The system helps in solving the task efficiently. – It is easy to cooperate with the team partner. – The system helps in solving team conflicts. – I felt comfortable in using the system.*

The final questionnaire used the same items with the request to compare both approaches, automatic and manual assignment, on a scale from 1 (manual assignment strongly preferred) to 5 (automatic assignment strongly preferred).

The distribution of the user satisfaction data using per-questionaire averages is shown in Figure 5 (right). The overall user satisfaction is higher in the auto mode, for both task sets. In addition, user satisfaction *decreases* within a *group* when switching from auto to manual, while it *increases* when switching from manual to auto.

The correlation of the subjective user satisfaction with the objective data from the log files confirm our hypothesis that the automatic display assignment is superior to the manual assignment in multi-user, multi-display situations with conflicting and dynamic document sets.

4 Summary

We have discussed the problem of assisting teams in effectively using multi-display environments for working together and we have addressed the question whether it is possible to infer the intention of the team and to find well-defined quality criteria for automatic display assignment.

Our results regarding team intention recognition, inference accuracy and speed showed that despite noisy observable sensor data and a rather ad hoc prior probability distribution for the occurrence of agenda items a precise and robust inference is possible. Further adding agenda knowledge to a team behavior model was identified as improvement for the compliant case and as non-disturbing for the non-compliant case. So, we can claim that unreliable agendas are useful for inferring team intentions. We will now focus on in-depth development of an appropriate team behavior model and incorporate learning of probabiltiy distributions using *EM-algorithm*.

In the strategy generation part, we have been able to show that automatic assignment enables teams to solve their tasks in a shorter time, with less conflicts between team members, with greater satisfaction and with reduced cognitive load. Future investigations will have to show whether this benefit offers the universality and significance required to incorporate it generally into smart multiple display environments.

Finally the seamless integration of our two-stage design is an issue that we will address in the future work.

References

1. David Franklin, Jay Budzik, and Kristian Hammond. Plan-based interfaces: keeping track of user tasks and acting to cooperate. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 79–86, New York, NY, USA, 2002. ACM Press.
2. Hung Bui. A general model for online probabilistic plan recognition. In *IJCAI '03: Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1309–1315, 2003.
3. Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 838–845, Washington, DC, USA, 2005. IEEE Computer Society.
4. Donald J. Patterson, Lin Liao, Dieter Fox, and Henry A. Kautz. Inferring high-level behavior from low-level sensors. In *Proceedings of UBICOMP 2003: The Fifth International Conference on Ubiquitous Computing*, October 12-15 2003.
5. Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry A. Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation services. In *Ubicomp*, pages 433–450. Springer, 2004.
6. Alois Ferscha, Gerd Kortuem, and Antonio Krüger. Workshop on ubiquitous display environments. In *Proc. Ubicomp 2004*, Nottingham, England, Sep 7 2004.
7. Maribeth Back et al. Usable ubiquitous computing in next generation conference rooms: design, architecture and evaluation. In *Ubicomp Workshop 2006*, Newport Beach, CA, USA, Sep 17 2006.
8. Patrick Chiu, Qiong Liu, John S. Boreczky, Jonathan Foote, Don Kimber, Surapong Lertsithichai, and Chunyuan Liao. Manipulating and annotating slides in a multi-display environment. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction*, 2003.
9. Brad Johanson, Greg Hutchins, Terry Winograd, and Maureen Stone. Pointright: Experience with flexible input redirection in interactive workspaces. In *Proc. ACM Conference on User Interface and Software Technology (UIST2002)*, pages 227–234, Paris, France, 2002.
10. Meredith R. Morris, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. Beyond "social protocols": multi-user coordination policies for co-located groupware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 262–265, New York, NY, USA, 2004. ACM Press.
11. Thomas Heider, Martin Giersich, and Thomas Kirste. Resource optimization in multi-display environments with distributed grasp. In *Proceedings of the First International Conference on Ambient Intelligence Developments (AmI.d'06)*, pages 60 – 76, Sophia Antipolis, France, September 19 - 22 2006. Springer.
12. Thomas Heider and Thomas Kirste. Automatic vs. manual multi-display configuration: A study of user performance in a semi-cooperative task setting. In *Proc. HCI'07: HCI.but not as we know it*, Lancaster, UK, September 3–7 2007.
13. Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. volume 13:3, pages 319–340. MIS Quarterly, Sep 1989.

A Survey of Semantics-based Approaches for Context Reasoning in Ambient Intelligence

Antonis Bikakis, Theodore Patkos, Grigoris Antoniou, and Dimitris Plexousakis

Institute of Computer Science, FO.R.T.H., Vassilika Voutwn
P.O. Box 1385, GR 71110, Heraklion, Greece
{bikakis,patkos,antoniou,dp}@ics.forth.gr

Abstract. A key issue in the study of Ambient Intelligence is reasoning about context. The aim of context reasoning is to deduce new knowledge, based on the available context data. The endmost goal is to make the ambient services more "*intelligent*"; closer to the specific needs of their users. The main challenges of this effort derive from the imperfect context information, and the dynamic and heterogeneous nature of the ambient environments. In this paper, we focus on semantics-based approaches for reasoning about context. We describe how each approach addresses the requirements of ambient environments, identify their limitations, and propose possible future research directions.

1 Introduction

Pervasive applications aim at providing the right information to the right users, at the right time, in the right place, and on the right device. In order to achieve this, a system must have a thorough knowledge and, as one may say, "*understanding*" of its environment, the people and devices that exist in it, their interests and capabilities, and the tasks and activities that are being undertaken. All this information falls under the notions of *context*.

The need for *reasoning* in context aware systems derives from the basic characteristics of context data. Two of these are *imperfection* and *uncertainty*. Henricksen and Indulska [1] characterize four types of imperfect context information: *unknown*, *ambiguous*, *imprecise*, and *erroneous*. Sensor or connectivity failures result in situations, that not all context data is available at any time. When the data about a context property comes from multiple sources, the context information may become ambiguous. Imprecision is common in sensor-derived information, while erroneous context information arises as a result of human or hardware errors. The role of reasoning in these cases is to detect possible errors, make predictions about missing values, and decide about the quality and the validity of the sensed data. The raw context data needs, then, to be transformed into meaningful information so that it can later be used in the application layer. In this direction, some suitable sets of rules can exploit the real meaning of some raw values of context properties. Finally, context reasoning may play the role of a decision making mechanism. Based on the collected context information, and on a set of decision rules provided by the user, the system can be configured to change its behavior, whenever certain changes are detected in its context.

If we also consider the high rates in which context changes and the potentially vast amount of available context information, the reasoning tasks become even more challenging. Overall, Knowledge Management in Ambient Intelligence should enable: *(a)* Reasoning with the highly dynamic and ambiguous context data; *(b)* Managing the potentially huge piece of context data, in a real-time fashion, considering the restricted computational capabilities of some mobile devices; and *(c)* Collective intelligence, by supporting information sharing, and distributed reasoning between the entities of the ambient environment.

In this paper, we present the various solutions that have been proposed to date, giving more attention to those that employ Semantic Web-based representations to describe context. The use of ontology languages is becoming common in such applications mainly because they offer enough representational capabilities to develop a formal context model that can be shared, reused, extended for the needs of specific domains, but also combined with data originating from other sources. Moreover, the development of the Semantic Web logic layer is resulting in rule languages that will enable reasoning with the user's needs and preferences and with the available ontology knowledge. According to the discussion on *Interactive Context-Aware Systems Interacting with Ambient Intelligence* in [2], ontology-based models manage to satisfy all demands placed concerning context modeling, such as distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality and, also, applicability to existing environments.

The rest of the paper is structured as follows: Section 2 focuses on ontological reasoning solutions, and Section 3 on rule-based approaches. Section 4 describes methods and techniques for distributed reasoning, while Section 5 discusses additional reasoning techniques concerning *learning*, *offline reasoning* and *probabilistic reasoning*. The last section proposes future research directions that may lead to more efficient reasoning solutions.

2 Ontological Reasoning

The SW Languages of RDF(S) and OWL are common formalisms for context representation. Along with their evolution, a number of SW Query languages (e.g. RDQL [3], RQL [4], TRIPLE [5]) and reasoning tools (e.g. FaCT [6], RACER [7], Pellet [8]) have been developed. Their aim is to retrieve relevant information, check the consistency of the available data, and derive implicit ontological knowledge. The studies of [9] and [10] describe the use of RDQL for accessing RDF context data, while the Context-Aware Guide described in [11] demonstrates the use of RQL in location-based mobile services. An interesting study that describes and evaluates the use of description logic for both representation and reasoning over context is presented in [12]. Below, we present representative examples of systems that reason with context data using Description Logics.

The P2P-based mobile environment in [13] consists of stations that provide semantic services and users with mobile devices, which manage their owner's semantic profile. Both the semantic services and the users' profiles are modeled as description logic predicates. The semantic matching between the services and the profiles, which determines whether a given profile is semantically compatible to a particular service and, if so, how well both do match, is accomplished by applying a set of DL rules, which are processed by a RACER reasoning engine.

In [14], they use a case study from the smart home domain (specifically a context-aware door-lock) to present their approach for modeling and reasoning about context using Description Logics. They have built an OWL schema to model the required context entities, and test three DL reasoners (RACER, its commercial successor RacerPro [15], and Pellet) using a real-case application scenario. However, their scenario is rather too simple to evaluate the performance of these reasoners in much broader context-aware applications.

The ontological reasoning approaches have two significant advantages. They integrate well with the ontology model, which is widely used for the representation of context; and most of them have relatively low computational complexity, allowing them to deal well with situations of rapidly changing context. However, their limited reasoning capabilities are a trade-off that we cannot neglect. They cannot deal with missing or ambiguous information, which is a common case in ambient environments, and are not able to provide support for decision making. Thus, we argue, that although we can use them in cases where we just want to retrieve information from the context knowledge base, check if the available context data is consistent or derive implicit ontological knowledge, they cannot serve as a standalone solution for the needs of ambient context-aware applications.

3 Rule-based Reasoning

In the Ambient Intelligence domain, rules are primarily used to express policies, constraints and preferences. Below, we present some representative examples.

In the SOCAM architecture, they use FOL rules to reason about context ([16]). To resolve possible conflicts, they have defined sets of rules on the classification and quality information of the context data. They suggest that different types of context have different levels of confidence and reliability. For example, defined context is more reliable compared to sensed and deduced context. They also have different levels of quality; for example, an RFID-based location sensor may have a 80% accuracy rate whereas a Bluetooth-based sensor may only have a 60% accuracy rate. The reasoning engine is implemented in Jena2.

In the *Semantic Space* Architecture, there are two modules for retrieving and deriving new information from the OWL Knowledge Base ([17]). The Context Query Engine provides an interface for applications to extract desired context information from the knowledge base. The Context Reasoner enables the users to deduce higher level knowledge, based on the context data of the KB, using FOL rules. The system uses Jena2 to perform forward-chaining reasoning over the KB, based on the rules provided by the user. The same approach is also followed in the prototype context-aware implementation descibed in [18].

As part of *Gaia*, Ranganathan and Campbell propose a FOL-based context infrastructure ([19]). The context information is represented as first-order predicates, with the name of a predicate being the type of context described. The model allows both universal and existential quantification over variables. This allows parameterizing context and representing a much richer set of contexts. A predefined set of rules is used to deduce higher-level knowledge based on the raw context data. Whenever a change occurs in the system's context, the rules are re-evaluated and the new inferred context replaces the old one. To resolve conflicts that occur when multiple rules are activated in the same time, they have developed a priority base mechanism, allowing only one rule to fire at each time. For the evaluation of the rules, they use the XSB reasoning engine.

In [20], the use of OWL is proposed both for context representation data, and for the rules expressing the user preferences and security constraints. Once all the context knowledge has been loaded in system (implemented on Jess), some predefined forward-chaining rules are used to complete the core knowledge base. The service invocation rules, and the privacy enforcing rules, both represented as backward-chaining rules are then applied to the knowledge base.

The Semantic Context-Aware Access Control Framework in [21] uses a combination of Description Logics and Logic Programming reasoning. Specifically, they define two types of rules: (a) *context aggregation rules* to support reasoning using property path relationships; (b) *context instantiation rules* to provide OWL assertions for attribute values. Both types of rules are expressed according to the following pattern: *if context attributes* $C_1 ... C_n$ *then context attribute* C_m, which corresponds to a Horn clause, where predicates in the head and in the body are represented by classes and properties defined in the context and application-specific ontologies. A similar hybrid reasoning approach is also implemented in the context-aware service adaptation middleware described in [22]).

Rule languages provide a formal model for context reasoning. Furthermore, they are easy to understand and widespread used, and there are many systems that integrate them with the ontology model. However, all these approaches share a common deficiency; they cannot handle the highly changeable, ambiguous and imperfect context information. In many of the cases that we described, they had to build additional reasoning mechanisms to deal with conflicts, uncertainty and ambiguities. The proposed logic models suit better in cases, where we are certain about the quality of the collected data. Consequently, neither of these models can serve as the solution to the required reasoning tasks.

4 Distributed Reasoning Techniques

In an Ambient Intelligence environment, there coexist many different entities that collect, process, and change the context information. Although they all share the same context, they face it from different viewpoints based on their perceptive capabilities, their experiences and their goals. Moreover, they may have different reasoning, storage and computing capabilities; they may "speak" different languages; they may even have different levels of sociality. This diversity raises additional research challenges in the study of smart spaces, which only few recent studies have addressed. In the following paragraphs, we present these approaches, which have the common feature of employing methods and techniques from the field of Distributed Artificial Intelligence.

One such approach is *sTuples* ([23]). This framework extends Tuple Spaces using SW technologies to represent and retrieve tuples from a Tuple Space. The Tuple Space model uses a logically shared memory, where producers add tuples to a common space, while consumers read or extract tuples from the space using a search template. The sTuples model advances the space lookup operations using DAML+OIL for the representation of context entities and RACER as the reasoning engine. It provides a generic framework to implement clients and services in a pervasive environment by using service and data tuples. Data tuples are semantic descriptions of the context data that an entity is willing to share with other entities in the environment, while service tuples are advertisements of the services offered in the same environment. Each entity uses various types of agents to gain access to the Tuple Space, each of which has a distinct role. Examples of such roles are, managing the addition, removal and state changes of tuples, searching in the Tuple Space, recommending services to the user, and notifying the user about tuple changes.

Similar approaches, which combine SW technologies and shared memory models to support asynchronous communications in ambient environments, are the *Semantic Spaces* ([24]), and the context management framework presented in [25]. The latter follows a *blackboard*-based approach. A mobile terminal system uses a central context manager, which stores context information from any available source. Clients can directly query the manager to gain context information, subscribe to various context change notification services, or use higher level contexts transparently. In the latter case, the context manager assigns the reasoning tasks to dedicated recognition services.

The OWL-SF framework ([26]) combines the OMG's *Super Distributed Objects* (SDO) technology and the OWL language to allow the distribution of semantically annotated services for the needs of ambient context-aware systems. SDOs are logical representations of hardware and software entities that are used to enable distributed interoperability. The proposed framework integrates two basic building blocks, OWL-SDOs and Deduction Servers. The OWL-SDOs are semantic extensions of SDOs; they use the OWL language to describe their status, services and communication interface. Deduction servers are specific OWL-SDOs that provide reasoning services. They contain a deduction engine coordinating reasoning tasks, an RDF inference layer providing rule reasoning support and an OWL-DL reasoner. Besides providing reasoning support, they are responsible for collecting the status of SDOs published using the OWL format, and for building an integrated OWL description accessible to reasoning.

The main feature that distinguishes the latter study is the lack of a central reasoning or control entity; it is fully decentralized. Collecting the reasoning tasks in a central entity certainly has many advantages; we can achieve better control, and better coordination between the various entities that have access to the central entity. Blackboard-based and shared-memory models have been thoroughly studied and used in many different types of distributed systems and have proved to work well in practice. The requirements are, though, much different in this setting. Context may not be restricted to a small room, office or apartment; we must also study cases of broader areas. The communication with a central entity is not guaranteed; we must assume unreliable and restricted wireless communications. Thus, a fully distributed scheme is a necessity. The OWL-SF framework is a step towards the right direction, but certainly not the last one. In order to deal with more realistic ambient environments, we need to eliminate some of the assumptions that they make. For example, different entities are not required to use the same representation and reasoning models, and we cannot always assume the existence of dedicated reasoning machines.

5 Other Reasoning Techniques

This section presents additional techniques that have been used to enhance the reasoning capabilities of AmI applications to deal with certain challenges, such as the ambiguity of context information, and the vast amount of context data.

In AmbieSense ([27]), they deal with the potentially vast amount of context data, using *Case Based Reasoning*. The reasoning mechanism is split into two different parts; the on-line part that resides on the user's mobile device, and the off-line part that resides on the user's backbone system. When new information arrives from the context retrieval module, it is translated to fit a preexistent ontology and sent to a CBR agent. The agent tries to retrieve a known context or case, and classifies the current situation based on the retrieved one. The associated goal is then presented to the task decomposition agent, and the case is stored in the case base. Since the user is expected to experience a few different situations daily, the storage of the cases will quickly fill up the mobile device and the CBR searching process will be hampered. To remedy this, some of the reasoning process is moved into the user's backbone servers.

The ec(h)o audio museum guide, described in [28], uses DAML+OIL ontologies for the representation of context data and user profiles. Its reasoning engine uses a forward-chaining reasoning mechanism to select the sound objects to be presented. The rules use several criteria that correspond to the semantic descriptions of the museum artifacts, the visitor's profile, and the way the visitor moves and interacts with the artifacts. To perform reasoning more efficiently, they build a virtual network that keeps track of possible combinations of facts, and support rule activation using the RETE algorithm (implemented in Jess).

The use of a Bayesian network to deal with the ambiguity of context data has been proposed in some recent studies. In MIRA, a context-based retrieval system capable of recording and indexing MBone videoconferences, they use a Bayesian network, coupled to a cost model, to describe a context-retrieval service that provides performance measures based on reliability and resource usage cost ([29]). In [30], a probabilistic model is used to define uncertain contexts. This model extends the OWL ontology model of SOCAM, by attaching probability values to the context predicates. They also adopt a Bayesian network as an underlying reasoning mechanism, as it has efficient probabilistic reasoning capabilities and allows representing causal relationships between various contexts. Bayesian networks to recognize high-level contexts have also been used in [25].[1]

6 Discussion

The special requirements of ambient environments impose the need of logic models that inherently deal with the imperfect nature of context data. Models that embody the notions of uncertainty, temporal and spatial change, and incompleteness would provide more robust and efficient solutions. A possible solution is the use of *nonmonotonic* reasoning, which has already been studied and used in other settings with similar requirements, such as the Web, e-learning environments, business rules, security specifications, negotiation protocols, and others. Recently, a number of nonmonotonic rule languages have been studied and reasoners that integrate them well with ontologies have been developed.

The main drawback of this approach is its relatively higher computational complexity, which becomes even worse, if we consider the potentially vast amount of available context data. A possible solution is to partition the large knowledge bases into smaller pieces, share these pieces with other computing devices, and deploy some form of partition-based reasoning. This is of course not an easy task, and only few recent studies have focused on this problem. An interesting approach is proposed in [31], which studies the partitioning of a large OWL ABox with respect to a TBox so that specific kinds of reasoning can be performed separately on each partition and the results trivially combined in order to achieve complete answers. In [32], they propose algorithms for reasoning with partitions of related logical axioms in propositional and first-order logic, and a greedy algorithm that automatically decomposes a set of logical axioms into partitions. Applying these ideas in AmI seems to be a very promising research direction.

Finally, to achieve collective intelligence, we must study methods for integrating and reasoning with data coming from heterogeneous sources and possibly described in different vocabularies. Translating all the data in a common format (schema) and performing centralized reasoning (followed by most of the studies that we presented) is one of some possible solutions. This approach is described as the Local-As-View approach in the Data Integration research area ([33]). Other approaches, concerning mainly the integration of heterogeneous data, are the Global-As-View approach and the Both-As-View approach ([33]), which have been recently studied and implemented in semantic P2P management systems. GAV assumes a global virtual schema, which is defined as a set of views over the data source schemas. This enables writing queries and rules using the local language of each data source. In BAV, local schemas are mapped to

[1] The modeling and reasoning approaches, along with the architecture and the aim of the systems referenced in Sections 2-6 are summarized in Table 1.

Table 1. Main Features of Context-Aware Frameworks

System	**Modeling**	**Reasoning**	**Architecture**	**Aim**
CoBrA [9]	OWL	RDQL	centralized (agent-based)	context-aware services
Context Awareness Framework [10]	RDF	RDQL	centralized	service prioritization
CG Platform [11]	RDF	RQL	centralized	location-based services
Semantic Mobile Environment [13]	DL	DL	distributed (P2P)	profile-service matchmaking
Context-Aware Door Lock [14]	OWL	DL	centralized	automatic door lock
SOCAM [16],[30]	OWL	FOL + Bayesian	centralized (middleware)	middleware for mobile services
Semantic Space [17]	OWL	RDQL+FOL	centralized	smart space mobile services
Gaia Context Infrastructure [19]	FOL	FOL	centralized	context-aware services
CONON Prototype [18]	OWL	DL+FOL	centralized	context-aware services
eWallet [20]	OWL	Jess	centralized (agent-based)	context-aware services
Context-Aware Access Control Framework [21]	OWL	DL+LP	centralized	policy evaluation
CARE [22]	OWL	DL+LP	centralized (middleware)	service adaptation
sTuples [23]	DAML+OIL	DL	decentralized shared memory	mobile services
Semantic Spaces [24]	RDF		decentralized shared memory	information sharing
Context Management Framework [25]	RDF	Bayesian	decentralized (blackboard-based)	information sharing notification services
OWL-SF [26]	OWL	DL	distributed (SDOs)	distributed services
AmbieSense [27]	taxonomies	CBR	centralized	context management
ec(h)o system [28]	DAML+OIL	Jess	centralized	audio museum guide
MIRA [29]	XML	Bayesian	centralized	videoconferences management

each other using a sequence of schema transformations (*mappings*). Reasoning with multiple ontologies interrelated with semantic mappings is studied in [34]. Examples of totally distributed reasoning algorithms, where the whole reasoning procedure can be viewed as a chain of reasoning tasks performed by different entities, can be found in [35]. These approaches can also lead to new ideas on how to exploit the different reasoning capabilities of each entity in an ambient environment, in order to make the whole system of entities more intelligent.

References

1. Henricksen, K., Indulska, J.: Modelling and Using Imperfect Context Information. In: Proceedings of PERCOMW '04, Washington, DC, USA, IEEE Computer Society (2004) 33–37
2. Schmidt, A.: Interactive Context-Aware Systems Interacting with Ambient Intelligence. In Riva, G., Vatalaro, F., Davide, F., Alcaniz, M., eds.: Ambient Intelligence, IOS Press (2005)
3. Seaborne, A.: RDQL - A Query Language for RDF. W3C member submission, Hewlett Packard (2004)
4. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M.: RQL: a declarative query language for RDF. In: Proceedings of the 11th International World Wide Web Conference (WWW). (2002) 592–603
5. Sintek, M., Decker, S.: TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In: Proceedings of the First International Semantic Web Conference, Springer-Verlag (2002) 364–378
6. FaCT: FaCT system website. http://www.cs.man.ac.uk/ horrocks/FaCT/ (2003)
7. Haarslev, V., Möller, R.: RACER System Description. In: IJCAR. (2001) 701–706
8. Parsia, B., Sirin, E.: Pellet: An OWL DL Reasoner. In: 3rd International Semantic Web Conference (ISWC2004). (2004)
9. Chen, H., Finin, T., Joshi, A.: Semantic Web in a Pervasive Context-Aware Architecture. Artificial Intelligence in Mobile System 2003 (2003) 33–40
10. Forstadius, J., Lassila, O., Seppanen, T.: RDF-based model for context-aware reasoning in rich service environment. In: PerCom 2005 Workshops. (2005) 15–19
11. Patkos, T., Bikakis, A., Antoniou, G., Plexousakis, D., Papadopouli, M.: A Semantics-based Framework for Context-Aware Services: Lessons Learned and Challenges. In: Proceedings of 4th International Conference on Ubiquitous Intelligence and Computing (UIC-2007). (2007) accepted for publication
12. van Bunningen, A.: Context aware querying - Challenges for data management in ambient intelligence. Technical Report TR-CTIT-04-51, Univ. of Twente (2004)
13. von Hessling, A., Kleemann, T., Sinner, A.: Semantic User Profiles and their Applications in a Mobile Environment. Fachberichte Informatik 2–2005, Universität Koblenz-Landau (2005)
14. Turhan, A.Y., Springer, T., Berger, M.: Pushing Doors for Modeling Contexts with OWL DL a Case Study. In: PERCOMW '06, Washington, DC, USA, IEEE Computer Society (2006)
15. Haarslev, V., Möller, R., Wessel, M.: RacerPro User Guide. http://www.racer-systems.com/products/racerpro/users-guide-1-9.pdf (2005)
16. Gu, T., Pung, H.K., Zhang, D.Q.: A Middleware for Building Context-Aware Mobile Services. In: Proceedings of the IEEE Vehicular Technology Conference (VTC 2004), Milan, Italy (2004)

17. Wang, X.H., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R., Zhang, D.: Semantic Space: an infrastructure for smart spaces. IEEE Pervasive Computing **3**(3) (2004) 32–39
18. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. In: PERCOMW '04, Washington, DC, USA, IEEE Computer Society (2004) 18
19. Ranganathan, A., Campbell, R.H.: An infrastructure for context-awareness based on first order logic. Personal Ubiquitous Comput. **7**(6) (2003) 353–364
20. Gandon, F.L., Sadeh, N.M.: Semantic web technologies to reconcile privacy and context awareness. Journal of Web Semantics **1** (2004) 241–260
21. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments. In: Proc. of 5th International Semantic Web Conference. (2006) 5–9
22. Agostini, A., Bettini, C., Riboni, D.: Experience Report: Ontological Reasoning for Context-aware Internet Services. In: PERCOMW '06, Washington, DC, USA, IEEE Computer Society (2006)
23. Khushraj, D., Lassila, O., Finin, T.: sTuples: Semantic Tuple Spaces. In: First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous04). (2004) 267–277
24. Krummenacher, R., Kopecký, J., Strang, T.: Sharing Context Information in Semantic Spaces. In: OTM Workshops. (2005) 229–232
25. Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., Malm, E.J.: Managing Context Information in Mobile Devices. IEEE Pervasive Computing **02**(3) (2003) 42–51
26. Mrohs, B., Luther, M., Vaidya, R., Wagner, M., Steglich, S., Kellerer, W., Arbanowski, S.: OWL-SF - A Distributed Semantic Service Framework. In: Workshop on Context Awareness for Proactive Systems (CAPS 2005). (2005)
27. Kofod-Petersen, A., Mikalsen, M.: Representing and Reasoning about Context in a Mobile Environment. Revue d'Intelligence Artificielle **19**(3) (2005) 479–498
28. Hatala, M., Wakkary, R., Kalantari, L.: Ontologies and rules in support of real-time ubiquitous application. Journal of Web Semantics, Special Issue on "Rules and ontologies for Semantic Web" **3**(1) (2005) 5–22
29. Castro, P., Mani, M., Mathur, S., Muntz, R.R.: Managing Context for Internet Videoconferences: The Multimedia Internet Recorder and Archive. In: Proc. of Multimedia and Computer Networks, San Jose, California (2000)
30. Gu, T., Pung, H.K., Zhang, D.Q.: A Bayesian Approach for Dealing with Uncertain Contexts. In: Proceedings of the Second International Conference on Pervasive Computing, Vienna, Austria, Austrian Computing Society (2004)
31. Guo, Y., Heflin, J.: A Scalable Approach for Partitioning OWL Knowledge Bases. In: Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006). (2006)
32. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artif. Intelligence **162**(1-2) (2005) 49–88
33. Halevy, A.Y., Rajaraman, A., Ordille, J.J.: Data Integration: The Teenage Years. In: VLDB. (2006) 9–16
34. Serafini, L., Tamilin, A.: DRAGO: Distributed Reasoning Architecture for the Semantic Web. In: ESWC. (2005) 361–376
35. Adjiman, P., Chatalic, P., Goasdoue', F., Rousset, M.C., Simon, L.: Distributed Reasoning in a Peer-to-Peer Setting: Application to the Semantic Web. Journal of Artificial Intelligence Research **25** (2006) 269–314

Distributed Reasoning with Conflicts in an Ambient Peer-to-Peer Setting

Antonis Bikakis and Grigoris Antoniou

Institute of Computer Science, FO.R.T.H., Vassilika Voutwn
P.O. Box 1385, GR 71110, Heraklion, Greece
{bikakis,antoniou}@ics.forth.gr

Abstract. In ambient environments, there coexist many different entities that collect, process, and change the available context information. Although they all share the same context, they face it from different viewpoints based on their perceptive capabilities, experiences and goals. Moreover, they are expected to use distinct vocabularies; they may even have different levels of sociality. This diversity raises additional research challenges in the study of Distributed Artificial Intelligence. In this paper, we present an algorithm for reasoning with distributed rule theories in an ambient setting. The algorithm models the participating agents as nodes in a peer-to-peer system, and considers the potential conflicts that may arise during the integration of the distributed theories taking into account some special characteristics of context knowledge and ambient agents.

1 Introduction

The study of ambient environments and pervasive computing systems has introduced new research challenges in the field of Distributed Artificial Intelligence. These are mainly caused by the imperfect nature of the available context information and the special characteristics of the agents that provide and process this knowledge. Henricksen and Indulska in [1] characterize four types of imperfect context information: *unknown*, *ambiguous*, *imprecise*, and *erroneous*. Sensor or connectivity failures (which are inevitable in wireless connections) result in situations, that not all context data is available at any time. When the data about a context property comes from multiple sources, the context information may become ambiguous. Imprecision is common in sensor-derived information, while erroneous context information arises as a result of human or hardware errors.

The agents that operate in an ambient environment are expected to have different goals, experiences and perceptive capabilities. They may use distinct vocabularies; they may even have different levels of sociality. Due to the highly dynamic and open nature of the environment (various entities join and leave the environment at random times), they are not able to know a priori all other entities that are present at a specific time instance nor can they communicate directly with all of them.

Considering these requirements, three main challenges of knowledge management in Ambient Intelligence are to enable:

1. Reasoning with the highly dynamic and ambiguous context data.
2. Managing the potentially huge piece of context data, in a real-time fashion, considering the restricted computational capabilities of some mobile devices.
3. Collective intelligence, by supporting information sharing, and distributed reasoning between the entities of the ambient environment.

So far, most pervasive computing frameworks have followed fully centralized approaches (e.g. [2–11]), while some others have employed models based on the *blackboard* and *shared memory* paradigms (e.g. [12–14]). Collecting the reasoning tasks in a central entity certainly has many advantages. It achieves better control, and better coordination between the participating entities. However, such solutions cannot meet the demanding requirements of ambient environments. The dynamics of the network and the unreliable and restricted (by the range of the transmitters) wireless communications inevitably lead to fully distributed solutions.

The goal of this study is to propose a distributed solution tailored to the special characteristics of ambient environments. The approach we propose to take models the agents of an ambient environment as nodes in a peer-to-peer system. Specifically, it considers nodes that have independent knowledge, and that interact with existing, *neighboring* nodes to exchange information. The internal knowledge is expressed in terms of rules, and knowledge is imported from other nodes through *bridging rules*.

Even if it is assumed that the theory of each node is locally consistent, the same assumption will not necessarily hold for the global knowledge base. The unification of the local theories, which model the viewpoints of the different nodes, may result in inconsistencies that are caused by the bridging rules. To deal with them, we follow a non-monotonic approach; bridging rules are expressed as *defeasible rules* (rules that may be defeated in the existence of adequate contrary evidence), and priorities between conflicting rules are determined by the level of *trust* that each node has on the other system nodes. In this way, the proposed approach manages to exploit the knowledge of every system node, and reason in a consistent and efficient manner, taking into account the viewpoint of each different node with regard to its context and cooperating peers.

The rest of the paper is structured as follows. Section 2 refers to the most prominent recent studies on reasoning in P2P data management systems and contextual reasoning. In Section 3, we present the algorithms that constitute our approach for reasoning with distributed rule theories. The conclusive section briefly describes the next steps of our work.

2 Related Work

Several recent studies have focused on developing formal models and methods for reasoning in peer-to-peer database systems. A key issue in formalizing data-oriented P2P systems is the semantic characterization of the *mappings* (bridging rules). One approach (followed in [15, 16]) is the first-order logic interpretation of P2P systems. In [17], Calavanese *et al.* identifies several drawbacks with this approach, regarding modularity, generality and decidability, and proposes new semantics based on epistemic logic. A common problem of both approaches is that they do not model and thus cannot handle inconsistency. Franconi *et al.* in [18] extends the autoepistemic semantics to formalize local inconsistency. The latter approach guarantees that a locally inconsistent database base will not render the entire knowledge base inconsistent. A broader extension, proposed by Calvanese *et al.* in [19], is based on nonmonontonic epistemic logic, and enables isolating local inconsistency, while also handling peers that may provide mutually inconsistent data. The proposed query evaluation algorithm assumes that all peers share a common alphabet of constants, and does not model *trust* or *priorities* between the peers. The propositional P2P inference system proposed by Chatalic *et al.* in [20] deals with conflicts caused by mutually inconsistent information sources, by detecting them and reasoning without them. The main problem is the same, once again: To perform reasoning, the conflicts are not actually resolved using some external trust or priority information; they are rather isolated.

Relevant to our work are also some recent research studies that combine the fields of multi-context systems (MCS) and nonmonotonic reasoning. The first prominent work in this research line was conducted by Roelofsen and Serafini. They define in [21] a non-monotonic rule-based MCS framework, which contains default negation in the rules. The multi-context variant of Default Logic, introduced by Brewka *et al.* in [22] is a step further towards nonmonotonic contextual reasoning. Specifically, the authors propose to model the bridge relations between different contexts as *default rules*. The latter study has the additional advantage that is closer to implementation due to the well-studied relation between Default Logic and Logic Programming. However, the authors do not provide certain reasoning algorithms, leaving some practical issues, such as the integration of priority information, unanswered.

3 Our Approach

We propose modeling the agents of an ambient environment as nodes in a P2P system. This choice is not arbitrary. The P2P paradigm captures many critical properties of ambient settings:

1. Each different peer independently collects and processes in its own way the available context information.
2. Each peer may not have (immediate) access to all information sources.
3. The peers share their knowledge through messages with their neighboring nodes.
4. Each peer may not trust all the other peers at the same level.
5. Peers join and leave the system randomly.

Below, we define our P2P model, which captures local knowledge, mapping relations through which the nodes exchange information, and trust between the system nodes. We also define the specific reasoning problem that we deal with, and describe the reasoning algorithms that we have developed.

3.1 Definitions

We assume a peer-to-peer system P as a collection of local theories:

$$P = \{P_i\}, i = 1, 2, ..., n$$

Each peer has a proper distinct vocabulary V_{P_i} and a unique identifier i. Each local theory is a set of rules that contain only local literals (literals from the local vocabulary). These rules are of the form:

$$r_i : a_i, b_i, ...k_i \rightarrow x_i$$

where i denotes the peer identifier.

Each peer also defines mappings that associate literals from its own vocabulary (*local literals*) with literals from the vocabulary of other peers (*remote literals*). The acquaintances of peer P_i, $ACQ(P_i)$ are the set of peers that at least one of P_i's mappings involves at least one of their local literals. The mappings are rules of the form:

$$m_i : a_i, b_j, ...z_k \rightarrow x$$

The above mapping rule is defined by P_i, and associates some of its own local literals with some of the literals defined by P_j, P_k and other system nodes. Literal x may belong to whichever vocabulary of these system nodes. Finally, each peer defines a trust order T_i, which includes a subset of the system nodes.

3.2 Problem Statement

Given a peer-to-peer system P, and a query about literal x_i issued at peer P_i, find the truth value of x_i considering P_i's local theory, its mappings and the theories of the other system nodes.

We assume that the local theories are consistent, but this is not necessarily true for the case of the unified theory $T(P)$, which is the collection of the theories (local rules and mappings) of the system nodes. The inconsistencies result from interactions between local theories and are caused by mappings.

An example of such conflicts derives in the following system of theories:

P_1	P_2	P_3
$r_{11} : a_1 \rightarrow x_1$	$r_{21} : \rightarrow a_2$	$r_{31} :\rightarrow a_3$
$m_{11} : a_2 \rightarrow a_1$		
$m_{12} : a_3 \rightarrow \neg a_1$		

P_i's theory is locally consistent, but with the addition of the the two mapping rules (m_{11}, m_{12}), which associate the literals of P_1 with those of P_2 and P_3, a conflict about literal a_1 derives from the interaction of the three theories.

3.3 *P2P_DR* Algorithm

The algorithm follows four main steps. In the first step (lines 1-16), it uses P_i's local theory to prove x_i. If x_i or its negation, $\neg x_i$, derives from the peer's local theory, the algorithm terminates returning Yes/No respectively, without considering the peer's mappings or the theories of other peers in the system.

In the second step (lines 17-41), if neither x_i nor $\neg x_i$ derives from the local theory, the algorithm also uses P_i's mappings. It collects all the rules that support x_i. For each such rule, it checks the provability of the literals in its body. For each local/remote literal, it issues similar queries (recursive calls of the algorithm) to P_i (local literals) or to the appropriate P_i's acquaintances (remote literals). To avoid circles, before each new call, the algorithm checks if the same query has been issued before, during the same query evaluation process. At the end of this step, the algorithm builds the mapping supportive set of x_i; this contains the set of mapping (locally or remotely defined) rules that can be used to prove x_i in the absence of contradictions.

The third step (lines 42-66) involves the rules that contradict x_i. The algorithm builds the mapping conflicting set of x_i, by collecting the rules that support $\neg x_i$.

In the last step (lines 64-71), the algorithm decides about x_i by comparing the supportive and conflicting sets. To compare two mapping sets, a peer uses its trust order T_i. According to this order, one mapping rule m_k is considered to be stronger than m_l from P_i's viewpoint if P_i trusts P_k more than P_l. The strength of a mapping set is determined by the weakest rule in this set. In the followings, we denote as:

r_i^l: a local rule of P_i
r_i^m: a mapping rule of P_i
r_i^{lm}: a rule (local/mapping) of P_i
R^m: the set of all mapping rules
$R_s(x_i)$: the set of supportive rules for x_i
$R_c(x_i)$: the set of conflicting rules for x_i

When a node P_i receives a query about x_i, it runs the **P2P_DR** algorithm. The algorithm parameters are:
x_i: the queried literal
P_0: the peer that issued the query
P_i: the local node
SS_{x_i}: the set of supportive mappings for x_i (initially empty)
CS_{x_i}: the set of conflicting mappings for x_i (initially empty)
$Hist_{x_i}$: the list of pending queries of the form: $[x_1, ..., x_i]$
Ans_{x_i}: the answer returned for x_i (initially empty)

P2P_DR$(x_i, P_0, P_i, SS_{x_i}, CS_{x_i}, Hist_{x_i}, Ans_{x_i})$
1: **if** $\exists r_i^l \in R_s(x_i)$ **then**
2: $\quad localHist_{x_i} \leftarrow [x_i]$

```
    run $local\_alg(x_i, localHist_{x_i}, localAns_{x_i})$
    if $localAns_{x_i} = Yes$ then
        $Ans_{x_i} \leftarrow localAns_{x_i}$
        terminate
    end if
end if
if $\exists r_i^l \in R_c(x_i)$ then
    $localHist_{x_i} \leftarrow [x_i]$
    run $local\_alg(\neg x_i, localHist_{x_i}, localAns_{\neg x_i})$
    if $localAns_{\neg x_i} = Yes$ then
        $Ans_{x_i} \leftarrow \neg localAns_{\neg x_i}$
        terminate
    end if
end if
for all $r_i^{lm} \in R_s(x_i)$ do
    $SS_{r_i} \leftarrow \{\}$
    for all $b_t \in body(r_i^{lm})$ do
        if $b_t \in Hist_{x_i}$ then
            stop and check the next rule
        else
            $Hist_{b_t} \leftarrow Hist_{x_i} \bigcup b_t$
            run $P2P\_DR(b_t, P_i, P_t, SS_{b_t}, CS_{b_t}, Hist_{b_t}, Ans_{b_t})$
            if $Ans_{b_t} = No$ then
                stop and check the next rule
            else
                $SS_{r_i} \leftarrow SS_{r_i} \bigcup SS_{b_t}$
            end if
        end if
    end for
    if $r_i^{lm} \in R^m$ then
        $SS_{r_i} \leftarrow SS_{r_i} \bigcup r_i^{lm}$
    end if
    if $Stronger(SS_{r_i}, SS_{x_i}, T_i) = Yes$ then
        $SS_{x_i} \leftarrow SS_{r_i}$
    end if
end for
if $SS_{x_i} = \{\}$ then
    return $Ans_{x_i} = No$ and terminate
end if
for all $r_i^{lm} \in R_c(x_i)$ do
    $SS_{r_i} \leftarrow \{\}$
    for all $b_t \in body(r_i^{lm})$ do
        if $b_t \in Hist_{x_i}$ then
            stop and check the next rule
        else
```

$Hist_{b_t} \leftarrow Hist_{x_i} \bigcup b_t$
run $P2P\_DR(b_t, P_i, P_t, SS_{b_t}, CS_{b_t}, Hist_{b_t}, Ans_{b_t})$
if $Ans_{b_t} = No$ **then**
stop and check the next rule
else
$SS_{r_i} \leftarrow SS_{r_i} \bigcup SS_{b_t}$
end if
end if
end for
if $r_i^{lm} \in R^m$ **then**
$SS_{r_i} \leftarrow SS_{r_i} \bigcup r_i^{lm}$
end if
if $Stronger(SS_{r_i}, CS_{x_i}, T_i) = Yes$ **then**
$CS_{x_i} \leftarrow SS_{r_i}$
end if
end for
if $CS_{x_i} = \{\}$ **then**
return $Ans_{x_i} = Yes$ and SS_{x_i} and terminate
end if
if $Stronger(SS_{x_i}, CS_{x_i}, T_i) = Yes$ **then**
return $Ans_{x_i} = Yes$ and SS_{x_i} and terminate
else
$Ans_{x_i} = No$ and terminate
end if

The $local\_alg(x_i, localHist_{x_i}, localAns_{x_i})$ is used to determine if x_i is a consequence of P_i's local theory. The algorithm parameters are:

x_i: the queried literal

$localHist_{x_i}$: the list of pending queries in P_i of the form: $[x_i^1, ..., x_i^m]$

$localAns_{x_i}$: the local answer returned for x_i (initially No)

local_alg$(x_i, localHist_{x_i}, localAns_{x_i})$
for all $r_i^l \in R_s(x_i)$ **do**
if $body(r_i^l) = \{\}$ **then**
return $localAns_{x_i} = Yes$
terminate
else
for all $b_i \in body(r_i^l)$ **do**
if $b_i \in localHist_{x_i}$ **then**
stop and check the next rule
else
$localHist_{b_i} \leftarrow localHist_{x_i} \bigcup b_i$
run $local\_alg(b_i, localHist_{b_i}, localAns_{b_i})$
end if
end for

if for every b_i: $localAns_{b_i} = Yes$ **then**
$localAns_{x_i} \leftarrow Yes$
terminate
end if
end if
end for

The $Stronger(S, C, T_i)$ function is used by P_i to check if the S set of mappings is stronger than the C set of mappings based on P_i's trust level order, T_i.

Stronger(S, C, T_i)

$r_s^w \leftarrow r_s \in S$ s.t. $forall\ r_i \in S : r_s$ is not weaker than r_i (according to T_i)
$r_c^w \leftarrow r_c \in C$ s.t. $forall\ r_j \in C : r_c$ is not weaker than r_j (according to T_i)
if r_s^w is stronger than r_c^w **then**
$Stronger = Yes$
else
$Stronger = No$
end if

3.4 Algorithm Properties

The application of the proposed algorithms in real scenarios largely depends on some properties regarding its termination and complexity.

Termination. We assume that there are a finite number of nodes in the system, each of which with a finite number of literals in its vocabulary. As a consequence, there are a finite number of rules that a peer may define. If the algorithm did not terminate, it would have to make indefinite recursive calls, adding each time a new query to the history, without ever returning an answer or detecting a cycle. However, this is impossible, because: (a) the number of recursive calls is bounded by the total finite number of literals in the system; and (b) there can be a finite number of independent (with different history) algorithm calls. These are bounded by the total finite number of rules in the system. Consequently, the algorithm will eventually terminate.

Number of Messages. To reduce the complexity of the algorithm with regard to the number of messages that the system nodes have to exchange, and the computational overhead of the algorithm on each system node, we can make the following optimization: Each node is required to retain two states: (a) the state of the queries it has been requested to process, INC_Q; this contains tuples of the form (q_i, Ans_{q_i}), where q_i is the queried literal, and Ans_{q_i} is $true/false$ in the case the node has completed the computation, or *undetermined* otherwise; and (b) the state of the queries it has requested other peers to process, OUT_Q (of the same form). Before sending a query to one of its neighbors, a node checks if the same query is in OUT_Q. If this is the case, it retrieves the answer stored in OUT_Q if this has the value $true/false$, or waits until the pending query returns a $true/false$ answer. When a new query is issued at a node, the node checks if the same query is in its INC_Q. If it is, the node returns the stored $true/false$ answer for that query if this has already been computed; otherwise, it suspends the new query until the pending query returns a $true/false$ answer. The space overhead of both states is proportional to the number of mappings that a node defines. The two states need to be updated every time a new query is issued at the system from an external source (we assume that the state of the network remains unchanged during the computation of each such query).

With these optimizations, each node will have to make at most one query for each of the remote literals that appear in the body of its mapping rules. In the worst case, that each peer has defined mappings that involve literals from all the other nodes in the system, and needs to apply all these mappings during a query evaluation, each peer will have to make $n \times n_l$ queries, where n is the number of system nodes and n_l is the maximum number of literals that a node may define. So, the total number of messages that need to be exchanged for the computation of a single query is in the worst case $n \times n \times n_l = O(n^2)$ (assuming that the number of nodes is the most critical parameter in the system).

4 Conclusion

We presented an approach for distributed reasoning in P2P settings, taking into account some special properties and constraints of context knowledge and ambient environments. The proposed reasoning algorithm models and reasons with potential conflicts that may arise during the integration of the distributed theories; to resolve these conflicts it uses trust information from the system nodes. We have already proved some desirable algorithm properties regarding its termination and complexity, and we are in the course of studying other properties, such as the computational complexity of the distributed algorithm on a single node. Other planned research directions of the same work are: (a) Study if there is an equivalent defeasible theory that derives from the unification of the distributed theories and produces the same results; (b) Extend the algorithm to support overlapping vocabularies; (c) Extend the algorithm to support defeasible local rules, and non-Boolean queries; and (d) Study applications in the Ambient Intelligence domain, where the theories may represent ontological knowledge (Horn logic subset of OWL DL), policies or regulations.

References

1. Henricksen, K., Indulska, J.: Modelling and Using Imperfect Context Information. In: Proceedings of PERCOMW '04, Washington, DC, USA, IEEE Computer Society (2004) 33–37
2. Chen, H., Finin, T., Joshi, A.: Semantic Web in a Pervasive Context-Aware Architecture. Artificial Intelligence in Mobile System 2003 (2003) 33–40

3. Forstadius, J., Lassila, O., Seppanen, T.: RDF-based model for context-aware reasoning in rich service environment. In: PerCom 2005 Workshops. (2005) 15–19
4. Patkos, T., Bikakis, A., Antoniou, G., Plexousakis, D., Papadopouli, M.: A Semantics-based Framework for Context-Aware Services: Lessons Learned and Challenges. In: Proceedings of 4th International Conference on Ubiquitous Intelligence and Computing (UIC-2007). (2007) accepted for publication
5. Gu, T., Pung, H.K., Zhang, D.Q.: A Middleware for Building Context-Aware Mobile Services. In: Proceedings of the IEEE Vehicular Technology Conference (VTC 2004), Milan, Italy (2004)
6. Wang, X.H., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R., Zhang, D.: Semantic Space: an infrastructure for smart spaces. IEEE Pervasive Computing **3**(3) (2004) 32–39
7. Ranganathan, A., Campbell, R.H.: An infrastructure for context-awareness based on first order logic. Personal Ubiquitous Comput. **7**(6) (2003) 353–364
8. Gandon, F.L., Sadeh, N.M.: Semantic web technologies to reconcile privacy and context awareness. Journal of Web Semantics **1** (2004) 241–260
9. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments. In: Proc. of 5th International Semantic Web Conference. (2006) 5–9
10. Kofod-Petersen, A., Mikalsen, M.: Representing and Reasoning about Context in a Mobile Environment. Revue d'Intelligence Artificielle **19**(3) (2005) 479–498
11. Hatala, M., Wakkary, R., Kalantari, L.: Ontologies and rules in support of real-time ubiquitous application. Journal of Web Semantics, Special Issue on "Rules and ontologies for Semantic Web" **3**(1) (2005) 5–22
12. Khushraj, D., Lassila, O., Finin, T.: sTuples: Semantic Tuple Spaces. In: First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous04). (2004) 267–277
13. Krummenacher, R., Kopecký, J., Strang, T.: Sharing Context Information in Semantic Spaces. In: OTM Workshops. (2005) 229–232
14. Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., Malm, E.J.: Managing Context Information in Mobile Devices. IEEE Pervasive Computing **02**(3) (2003) 42–51
15. Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data Management for Peer-to-Peer Computing : A Vision. In: WebDB. (2002) 89–94
16. Halevy, A.Y., Ives, Z.G., Suciu, D., Tatarinov, I.: Schema Mediation in Peer Data Management Systems. In: ICDE. (2003) 505
17. Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Logical Foundations of Peer-To-Peer Data Integration, ACM (2004) 241–251
18. Franconi, E., Kuper, G.M., Lopatenko, A., Serafini, L.: A Robust Logical and Computational Characterisation of Peer-to-Peer Database Systems. In: DBISP2P. (2003) 64–76
19. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Inconsistency Tolerance in P2P Data Integration: an Epistemic Logic Approach. In: DBPL-05. Volume 3774 of LNCS., SV (2005) 90–105
20. Chatalic, P., Nguyen, G.H., Rousset, M.C.: Reasoning with Inconsistencies in Propositional Peer-to-Peer Inference Systems. In: ECAI. (2006) 352–356
21. Roelofsen, F., Serafini, L.: Minimal and Absent Information in Contexts. In: IJCAI. (2005) 558–563
22. Brewka, G., Roelofsen, F., Serafini, L.: Contextual Default Reasoning. In: IJCAI. (2007) 268–273

Model-Based Default Refinement of Partial Information within an Ambient Agent

Fiemke Both, Charlotte Gerritsen, Mark Hoogendoorn, and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
{fboth, cg, mhoogen, treur}@cs.vu.nl
http://www.few.vu.nl/~{fboth, cg, mhoogen, treur}

Abstract. Ambient agents react on humans on the basis of partial information obtained by sensoring. Appropriate types of reactions depend on in how far an ambient agent is able to interpret the available information (which is often incomplete, and hence multi-interpretable) in order to create a more complete internal image of the environment, including humans. This interpretation process, which often has multiple possible outcomes, can make use of an explicitly represented model of causal and dynamic relations. Given such a model representation, the agent needs a reasoning method to interpret the partial information available by sensoring, by generating one or more possible interpretations. This paper presents a generic model-based default reasoning method that can be exploited to this end. The method allows the use of software tools to determine the different default extensions that form the possible interpretations.

1 Introduction

Ambient Intelligence [1, 2, 16] applications usually involve sensor information about the environment, including humans. As this information is often incomplete, applications that require a high level of context awareness (see also [17, 18, 19]) depend on the availability of methods to analyse such information. One way is to include computational models about environmental and human functioning in ambient agents. However, even when incomplete sensor information is refined on the basis of such models to create a more complete internal image of the environment's and human's state, still this may result in partial information that can be interpreted in different manners. Reactions of ambient agents then depend on in how far they are able to handle the available multi-interpretable information. To do this, the agent needs a reasoning method to generate one or more of the possible interpretations. Tools from the area of nonmonotonic logic can provide adequate analysis tools for reasoning processes concerning partial information. Within nonmonotonic logic approaches it is possible to formalise reasoning processes that deal with multiple possible outcomes, which can be used to model different possibilities of interpretation; see [10] for a similar perspective on the application of nonmonotonic logic tools.

This paper presents a generic model-based default reasoning method that can be exploited to this end. The method exploits the available causal model and allows the use of software tools to determine the different default extensions that form the possible interpretations, given the sensor information and the causal model. Moreover, by formally specifying the default rules in an executable temporal format, according to the approach put forward in [8, 9], explicit default reasoning processes can be generated.

Section 2 describes two case studies used to illustrate the approach. In Section 3 the basic concepts used are briefly introduced. Section 4 presents the approach to use default logic in conjunction with causal graphs to refine partial information by defining multiple interpretations. Finally, Section 5 is a discussion.

2 Case Studies

Two case studies are used throughout this paper; they are introduced below.

Wristband for Elderly

As a case study, the reasoning concerning conditions that occur amongst elderly people is used. Figure 1 shows a simplified causal model for such conditions. On the left hand side five conditions are shown: awake, asleep, syncope (fainted), myocardial infarction (heart attack) and cardiac arrest. The output of the model consists of symptoms that can be measured with a wristband, which are pulse, blood pressure and body temperature. Such a causal model can help in finding out the current condition of an elderly person based on sensory information from the wristband.

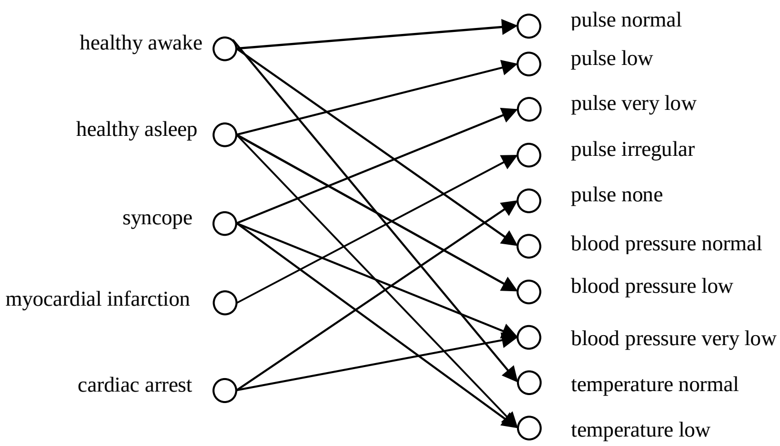


Fig. 1. Causal model for the condition of an elderly person

Crime Case

In this case study, a system is used that can help the police solve a crime using ambient intelligence facilities. A Dutch company (Sound Intelligence) developed microphones that can distinguish aggressive sounds. Consider the situation in which these microphones are distributed at crucial points in the city, similar to surveillance cameras. Furthermore, suppose in this scenario that for some persons ankle bracelets are used as a form of punishment, which can measure the level of ethanol in the person's perspiration, and indicate their position.

In this example scenario, someone is beaten up nearby a microphone. The microphone picks up the sound of the fight and records this. After an investigation, the police have three suspects. The first suspect is known to have a high level of testosterone, which often leads to aggressive behaviour. The second suspect is someone who is sensitive for alcohol (causing aggression) and wears an ankle bracelet that measures the level of ethanol in his system. He has been seen in a nearby cafe. The third suspect is diagnosed with Intermittent Explosive Disorder (IED), which is a disorder that can lead to a terrible outburst of rage after an unpleasant or stressful meeting. Witnesses saw suspect 2 in the company of someone else.

Figure 2 shows a causal model that is used for this situation that can help the police officers to figure out what information is missing and help them to plan their strategy. For example, did suspect 2 have a conflict with the person he was with? Did suspect 3 drink alcohol? Aggressive sounds are caused by persons that are aggressive, according to the model. Three possible causes for this aggressiveness are considered, as can be seen in Figure 2: someone can have a high level of testosterone, someone can just have been in a situation of conflict or someone can have a high level of alcohol.

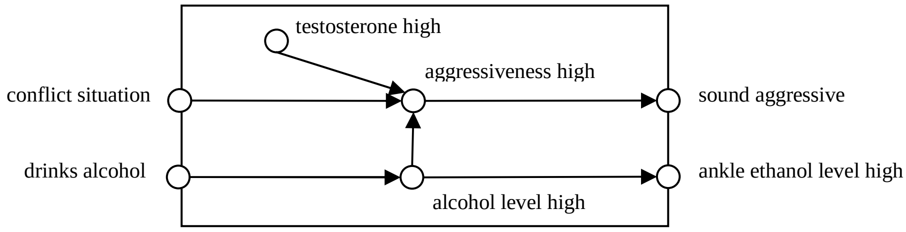


Fig. 2. Causal model for the crime case

3 Basic Concepts Used

In this section the basic concepts used in the paper are briefly introduced.

Causal models

In this paper, this dynamic perspective on reasoning is applied in combination with facts that are labelled with temporal information, and models based on causal or temporal relationships that relate such facts. To express the information involved in an agent's internal reasoning processes, the following ontology is used.

leads_to_after(I:INFO_EL, J:INFO_EL, D:REAL)	state property I leads to state property J after duration D
at(I:INFO_EL, T:TIME)	state property I holds at time T

Multiple Interpretation

Reasoning to obtain an interpretation of partial information can be formalised at an abstract generic level as follows. A particular interpretation for a given set of formulae considered as input information for the reasoning, is formalised as another set of formulae, that in one way or the other is derivable from the input information (output of the reasoning towards an interpretation). In general there are multiple possible outcomes. The collection of all possible interpretations derivable from a given set of formulae as input information (i.e., the output of the reasoning towards an interpretation) is formalised as a collection of different sets of formulae. A formalisation describing the relation between such input and output information is described at an abstract level by a multi-interpretation operator.

The input information is described by propositional formulae in a language L_1. An interpretation is a set of propositional formulae, based on a language L_2.

a) A *multi-interpretation operator* MI with input language L_1 and output language L_2 is a function $MI : P(L_1) \rightarrow P(P(L_2))$ that assigns to each set of input facts in L_1 a set of sets of formulae in L_2.
b) A multi-interpretation operator MI is *non-inclusive* if for all $X \subseteq L_1$ and $S, T \in MI(X)$, if $S \subseteq T$ then $S = T$.
c) If $L_1 \subseteq L_2$, then a multi-interpretation operator MI is *conservative* if for all $X \subseteq L_1$, $T \in MI(X)$ it holds $X \subseteq T$.

The condition of non-inclusiveness guarantees a relative maximality of the possible interpretations. Note that when MI(X) has exactly one element, this means that the set $X \subseteq L_1$ has a unique interpretation under MI. The notion of multi-interpretation operator is a generalisation of the notion of a nonmonotonic belief set operator, as introduced in [6]. The generalisation was introduced and applied to approximate classification in [10]. A reasoner may explore a number of possible interpretations, but often, at some point in time a reasoner will focus on one (or possibly a small subset) of the interpretations. This selection process is formalised as follows (see [10]).

a) A *selection operator* s is a function $s : P(P(L)) \rightarrow P(P(L))$ that assigns to each nonempty set of interpretations a nonempty subset: for all A with $\phi \neq A \subseteq P(L)$ it holds $\phi \neq s(A) \subseteq A$. A selection operator s is *single-valued* if for all non-empty A the set s(A) contains exactly one element.
b) A *selective interpretation operator* for the multi-interpretation operator MI is a function $C : P(L_1) \rightarrow P(L_2)$ that assigns one interpretation to each set of initial facts: for all $X \subseteq L_1$ it holds $C(X) \in MI(X)$.

Representation in Default Logic

The *representation problem* for a nonmonotonic logic is the question whether a given set of possible outcomes of a reasoning process can be represented by a theory in this logic. More specifically, representation theory indicates what are criteria for a set of possible outcomes, for example, given by a collection of deductively closed sets of formulae, so that this collection can occur as the set of outcomes for a theory in this nonmonotonic logic. In [13] the representation problem is solved for default logic, for the finite case. Given this context, in the current paper Default Logic is chosen to represent interpretation processes. For the empirical material analysed, default theories have been specified such that their extensions are the possible interpretations.

A *default theory* is a pair $\langle D, W \rangle$. Here W is a finite set of logical formulae (called the background theory) that formalise the facts that are known for sure, and D is a set of default rules. A default rule has the form: $\alpha: \beta / \gamma$. Here α is the precondition, it has to be satisfied before considering to believe the conclusion γ, where the β, called the justification, has to be consistent with the derived information and W. As a result γ might be believed and more default rules can be applied. However, the end result (when no more default rules can be applied) still has to be consistent with the justifications of all applied default rules. *Normal default theories* are based on defaults of the form $\alpha: \beta / \beta$. In the approach *supernormal* default rules will be used: normal default rules where α is trivial: true. Such supernormal rules are denoted by β / β or $: \beta / \beta$; they are also called prerequisite-free normal defaults. For more details on Default Logic, such as the notion of extension, see e.g. [12, 15].

Temporal Specification of Reasoning Processes

In this paper a dynamic perspective on reasoning is taken, following, e.g. [8, 9]. In practical reasoning situations usually different lines of reasoning can be generated, each leading to a distinct set of conclusions. In logic semantics is usually expressed in terms of models that represent descriptions of conclusions about the world and in terms of entailment relations based on a specific class of this type of models. In the (sound) classical case each line of reasoning leads to a set of conclusions that are true in all of these models: each line of reasoning fits to each model. However, for non-classical reasoning methods the picture is different. For example, in default reasoning or abductive reasoning methods a variety of mutually contradictory conclusion sets may be possible. It depends on the chosen line of reasoning which one of these sets fits.

The general idea underlying the approach followed here, and inspired by [8, 9], is that a particular reasoning line can be formalised by a sequence of *information states* $M_0, M_1, \ldots\ldots$. Here any M_t is a description of the (partial) information that has been derived up to time point t. From a dynamic perspective, an inference step, performed in time duration D is viewed as a transition $M_t \rightarrow M_{t+D}$ of a current information state M_t to a next information state M_{t+D}. Such a transition is usually described by application of a deduction rule or proof rule, which in the dynamic perspective on reasoning gets a temporal aspect. A particular reasoning line is formalised by a sequence $(M_t)_{t \in T}$ of subsequent information states labelled by elements of a flow of time T, which may be discrete, based on natural numbers, or continuous, based on real numbers.

An information state can be formalised by a set of statements, or as a three-valued (false, true, undefined) truth assignment to ground atoms, i.e., a partial model. In the latter case, which is followed here (as in [8, 9]), a sequence of such information states or reasoning trace can be interpreted as a partial temporal model. A transition relating a next information state to a current one can be formalised by temporal formulae the partial temporal model has to satisfy.

Executable Temporal Specification

To specify models and to execute these models, the language LEADSTO, an executable sublanguage of Temporal Trace Language (TTL), is used. The basic building blocks of this language are causal relations of the format $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, which means:

if state property α holds for a certain time interval with duration g,
then after some delay (between e and f) state property β will hold
for a certain time interval of length h.

where α and β are state properties of the form ‘conjunction of literals’ (where a literal is an atom or the negation of an atom), and e, f, g, h non-negative real numbers. For the sake of simplicity, especially when they are always the same, these subscripts may be left out of the notation and indicated separately. As an example, a modus ponens deduction rule in time duration D can be specified in temporal format as:

$$\text{derived}(I) \wedge \text{derived}(\text{implies}(I, J)) \twoheadrightarrow_D \text{derived}(J)$$

So, inference rules are translated into temporal rules thus obtaining a temporal theory describing the reasoning behaviour. Each possible line of reasoning can be described by a linear time model of this theory (in temporal partial logic). This representation format will be used to formalise this and other types of model-based reasoning methods, as is shown more extensively in Appendix A†.

† http://www.few.vu.nl/~fboth/default-refinement

4 Representing Model-Based Interpretation in Default Logic

In this section it is discussed how a model-based interpretation operator can be represented in default Logic.

4.1 Default logic for model-based refinement of partial information

The *causal theory* CT of the agent consists of a number of statements a $\rightarrow$ b for each causal relation from a to b, with a and b atoms. Sometimes included in this set are some facts to indicate that some atoms exclude each other (for example, $\neg$(has_value(temperature, high) $\wedge$ has_value(temperature, low) assuming that temperature can only be high or low), or that at least one of a set of atoms is true, (for example: has_value(pulse, high) $\vee$ has_value(pulse, normal) $\vee$ has_value(pulse, low)). A set of literals S is *coherent* with CT if S $\cup$ CT is consistent. The set S is called a *maximal coherent* set for CT if it is coherent, and for all sets T coherent with CT with S $\subseteq$ T it holds S = T. Let X be a set of formulae. The multi-interpretation operator $MI_{CT}(X)$ is defined by

$$MI_{CT}(X) = \{ Cn(X \cup CT \cup S) \mid S \text{ maximal coherent with CT} \}$$

This operator defines for the partial information the agent may have at some point in time (indicated by set of literals X) the set of all complete refinements of X which are coherent with the causal model. This operator has been defined above in an abstract manner, and only indicates the possible outcomes of a reasoning process, not the steps of the reasoning process itself. A next step is to obtain a representation of this operator in a well-known formalism such as default logic. Based on this default logic representation, reasoning processes can be defined that can be performed to obtain one or more of the interpretations.

The following Default Theory $\Delta_{CT}(X) = \langle W, D \rangle$ can be used to represent the multi-interpretation operator MI_{CT} (notice that this is a supernormal default theory); see also [13], Theorem 5.1:

$$W = CT \cup X$$
$$D = \{ (\text{true}: a / a) \mid \text{a literal for an atom occurring in CT} \}$$

Here a literal is an atom or a negation of an atom. That this default theory represents MI_{CT} means that for any set X indicating partial information the set of interpretations defined by $MI_{CT}(X)$ can be obtained as the set of all extensions of the default theory $\Delta_{CT}(X)$. This representation allows to determine the interpretations by using known methods and tools to determine the extensions of a default theory. One of these methods is worked out in a tool called Smodels, based on answer set programming; cf. [14]. The use of this for the two case studies will be discussed in the next two Subsections 4.2 and 4.3. Another method to determine the extensions of a default theory is by controlled or prioritised default reasoning. This method is illustrated in Appendix A.

4.2 A Default Theory for the Wristband for Elderly Case

In order to represent the knowledge introduced in Section 2.1, the following default theory has been specified. First, the causal background theory (W = CT) is defined, based on the causal graph shown in Figure 1. Furthermore, inconsistent values are defined for the various facets (i.e. pulse, temperature, blood pressure, and condition):

```
inconsistent_values(pulse, normal, low)
inconsistent_values(condition, healthy_awake, healthy_asleep)
etc.
```

If an attribute has a certain value and this value is inconsistent with another value, then this other value is not the case.

has_value(y, x1) $\wedge$ inconsistent_values(y, x1, x2) $\rightarrow \neg$ has_value(y, x2)

Besides the background theory, also the default theory Δ_{CT} has been generated from this causal theory CT. The default rules for the atoms are simply as follows:

has_value(condition, healthy_awake) / has_value(condition, healthy_awake)
has_value(condition, healthy_asleep) / has_value(condition, healthy_asleep)
has_value(condition, syncope) / has_value(condition, syncope)
has_value(condition, myocardial_infarction) / has_value(condition, myocardial_infarction)
has_value(condition, cardiac_arrest) / has_value(condition, cardiac_arrest)
has_value(pulse, normal) / has_value(pulse, normal)
has_value(pulse, low) / has_value(pulse, low)
has_value(pulse, very_low) / has_value(pulse, very_low)
has_value(pulse, irregular) / has_value(pulse, irregular)
has_value(pulse, none) / has_value(pulse, none)
has_value(blood_pressure, normal) / has_value(blood_pressure, normal)
has_value(blood_pressure, low) / has_value(blood_pressure, low)
has_value(blood_pressure, very_low) / has_value(blood_pressure, very_low)
has_value(temperature, normal) / has_value(temperature, normal)
has_value(temperature, low) / has_value(temperature, low)

Besides these default rules, similar defaults for the negations of these atoms are included. Using a system called Smodels [14], the extensions for the default theory specified can be calculated. Using the theory above, 30 extensions result. Hereby, in 19 out of 30 cases neither of the 5 conditions holds (i.e. awake, asleep, syncope, myocardial infarction and cardiac arrest). However, by adding strict rules which express that at least one of the conditions holds, only 11 extensions are found. The extensions that follow after adding these strict rules are shown in Table 1.

Table 1. All extensions of the default theory

#	Condition	Values	#	Condition	Values
1	healthy_awake	has_value(pulse, normal) has_value(blood_pressure, normal) has_value(temperature, normal)	7	myocardial_ infarction	has_value(pulse, irregular) has_value(blood_pressure, normal) has_value(temperature, low)
2	healthy_asleep	has_value(pulse, low) has_value(blood_pressure, low) has_value(temperature, low)	8	myocardial_ infarction	has_value(pulse, irregular) has_value(blood_pressure, low) has_value(temperature, low)
3	syncope	has_value(pulse, very_low) has_value(blood_pressure, very_low) has_value(temperature, low)	9	myocardial_ infarction	has_value(pulse, irregular) has_value(blood_pressure, very_low) has_value(temperature, low)
4	myocardial_ infarction	has_value(pulse, irregular) has_value(blood_pressure, normal) has_value(temperature, normal)	10	cardiac_arrest	has_value(pulse, none) has_value(blood_pressure, very_low) has_value(temperature, normal)
5	myocardial_ infarction	has_value(pulse, irregular) has_value(blood_pressure, low) has_value(temperature, normal)	11	cardiac_arrest	has_value(pulse, none) has_value(blood_pressure, very_low) has_value(temperature, low)
6	myocardial_ infarction	has_value(pulse, irregular) has_value(blood_pressure, very_low) has_value(temperature, normal)			

Partial information X may be given that includes the information that the person has a normal temperature. Such a set X can be added to the background theory W. Table 2 shows the extensions resulting when the following facts are added to W:

X = { has_value(temperature, normal), has_value(pulse, irregular) }

Table 2. All extensions given the changed background theory

#	Condition	Values
1	myocardial_infarction	has_value(pulse, irregular) has_value(blood_pressure, normal) has_value(temperature, normal)
2	myocardial_infarction	has_value(pulse, irregular) has_value(blood_pressure, low) has_value(temperature, normal)
3	myocardial_infarction	has_value(pulse, irregular) has_value(blood_pressure, very_low) has_value(temperature, normal)

Finally, Table 3 shows the extensions when the following set X is added to W:

X = { has_value(temperature, normal) , has_value(pulse, normal) , has_value(blood_pressure, normal) }

Table 3. All extensions of the default theory

#	Condition	Values
1	healthy_awake	has_value(pulse, normal) has_value(blood_pressure, normal) has_value(temperature, normal)

4.3 Crime Case Default Theory

Similar to the Elderly Wristband, the default theory Δ_{CT} for the crime case has been generated from the causal model:

has_value(situation, conflict) / has_value(situation, conflict)
has_value(situation, drinks_alcohol) / has_value(situation, drinks_alcohol)
has_value(testosterone, high) / has_value(testosterone, high)
has_value(sounds, aggressive) / has_value(sounds, aggressive)
has_value(ankle_ethanol_level, high) / has_value(ankle_ethanol_level, high)
has_value(aggressiveness, high) / has_value(aggressiveness, high)
has_value(alcohol_level, high) / has_value(alcohol_level, high)
not(has_value(situation, conflict) / not(has_value(situation, conflict))
not(has_value(situation, drinks_alcohol) / not(has_value(situation, drinks_alcohol))
not(has_value(testosterone, high) / not(has_value(testosterone, high))
not(has_value(sounds, aggressive) / not(has_value(sounds, aggressive))
not(has_value(ankle_ethanol_level, high) / not(has_value(ankle_ethanol_level, high))
not(has_value(aggressiveness, high) / not(has_value(aggressiveness, high))
not(has_value(alcohol_level, high) / not(has_value(alcohol_level, high))

Furthermore, aggressive sound has been observed, therefore the following fact is added to W:

X = {has_value(sound, aggressive)}

The resulting number of extensions is 18. Hereby however, the reasoning has not been performed using a closed world assumption, whereby values can only occur in case they result from a known causal relation or in case they are input variables (i.e. the situation). In order to perform reasoning with such a closed world assumption, the following rules have been added. First, a rule expressing that in case there is only one source from which a value can be derived, then this source should have the appropriate value (in this case, this holds for all variables except for aggressiveness).

has_value(X1, Y1) $\wedge$ leads_to(has_value(X2, Y2), has_value(X1, Y1)) $\wedge$ X1 $\neq$ aggressiveness $\rightarrow$
has_value(X2, Y2)

For the aggressiveness a different set of rules is used, since only one out of three conditions needs to hold. An example of one instance of such a rule is the following:

has_value(aggressivness, high) $\wedge$ not(has_value(testosterone, high) $\wedge$ not(has_value(situation, conflict) $\rightarrow$ has_value(alcohol_level, high)

Given that these rules are added, 7 extensions result using Smodels as shown in Table 4. Note that the sound is not shown since that is fixed in advance already. The last column shows to which suspect this extension is applicable. Hereby the suspect with high testosterone is marked with 1, the oversensitive alcohol suspect with 2, and the IED suspect with 3.

Table 4. Extensions given that aggressive sound has been observed

#	Situation	Testosterone	Aggressiveness	Alcohol level	Ankle Ethanol level	Suspect
1	¬conflict; ¬drinks_alcohol	high	high	¬high	¬high	1
2	conflict; ¬drinks_alcohol	high	high	¬high	¬high	1
3	conflict; ¬drinks_alcohol	¬high	high	¬high	¬high	3
4	conflict; drinks_alcohol	high	high	high	high	1
5	conflict; drinks_alcohol	¬high	high	high	high	2, 3
6	¬conflict; drinks_alcohol	¬high	high	high	high	2
7	¬conflict; drinks_alcohol	high	high	high	high	1

5 Discussion

This paper shows how a number of known techniques and tools developed within the area of nonmonotonic reasoning and AI can be applied to analyse model-based interpretation. The formal techniques exploited in the approach, are causal graphs and causal reasoning in conjunction with techniques from the nonmonotonic reasoning area such as: multi-interpretation operators as an abstract formalisation multiple interpretation and a default theory to represent this multi-interpretation operator. Model-based default refinement can be useful to obtain (on top of sensor information) a high level of context awareness; see also [17, 18, 19]. The properties and default rules presented in this paper have all been specified in a generic fashion, such that they can easily be reused for studying other cases.

More formalisms for handling causal or temporal reasoning within ambient intelligence have been proposed, see e.g. [11]. The application of nonmonotonic logic as put forward in this paper adds the possibility to specify human like reasoning in a natural way, possibly even resulting in multiple stable sets that can be the outcome of such a reasoning process.

Currently, the approach put forward is a theoretical framework, whereby case studies have been conducted on paper. Future work is to see how well such a theoretical framework can be applied in a practical setting, for example for elderly care or crime analysis. Issues such as how to extract the appropriate information needed within the system from domain experts, how useful the system can be in supporting human decision makers, and how accessible the method can be made for people not familiar with formal methods will need to be addressed.

References

1. Aarts, E.; Collier, R.; van Loenen, E.; Ruyter, B. de (eds.) (2003). *Ambient Intelligence. Proc. of the First European Symposium, EUSAI 2003*. Lecture Notes in Computer Science, vol. 2875. Springer Verlag, 2003, pp. 432.
2. Aarts, E., Harwig, R., and Schuurmans, M. (2001), Ambient Intelligence. In: P. Denning (ed.), *The Invisible Future*, McGraw Hill, New York, pp. 235-250.
3. Bosse, T., Jonker, C.M., and Treur, J. (2006). Formalization and Analysis of Reasoning by Assumption. *Cognitive Science Journal*, volume 30, issue 1, 2006, pp. 147-180.
4. Bosse, T., and Treur, J. (2006) Modelling Dynamics of Cognitive Agents by Higher-Order Potentialities. In: Stone, P. and Weiss, G. (eds.), *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'06*. ACM Press, 2006, pp. 117-119.
5. Bosse, T., and Treur, J., (2007). Higher-Order Potentialities and their Reducers: A Philosophical Foundation Unifying Dynamic Modelling Methods. In: M.M. Veloso (ed.), *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, IJCAI'07*. AAAI Press, 2007, pp. 262-267.
6. Engelfriet, J., Herre, H., and Treur, J. (1998), Nonmonotonic Reasoning with Multiple Belief Sets, *Annals of Mathematics and Artificial Intelligence,* vol. 24, pp. 225-248.
7. Engelfriet, J., Jonker, C.M., and Treur, J. (2002). Compositional Verification of Multi-Agent Systems in Temporal Multi-Epistemic Logic. *Journal of Logic, Language and Information,* vol. 11, 2002, pp. 195-225.
8. Engelfriet, J., and Treur, J., (1995). Temporal Theories of Reasoning. *Journal of Applied Non-Classical Logics*, vol. 5, 1995, pp. 239-261
9. Engelfriet J., and Treur J. (2000). Specification of Nonmonotonic Reasoning. *Journal of Applied Non-Classical Logics,* vol. 10, 2000, pp. 7-27.
10. Engelfriet, J., and Treur, J. (2003), Multi-Interpretation Operators and Approximate Classification. *Int. Journal of Approximate Reasoning*, vol. 32, pp. 43-61.
11. Galton, A., Causal Reasoning for Alert Generation in Smart Homes. In: J. C. Augusto and C. D. Nugent (eds.), *Designing Smart Homes*, Springer-Verlag LNAI 4008, 2006, pp.57-70.
12. Marek, V.W., and Truszczynski, M. (1993), *Nonmonotonic Logics*. Springer Verlag.
13. Marek, V.W., Treur, J., and Truszczynski, M. (1997), Representation Theory for Default Logic. *Annals of Mathematics and AI*, vol. 21, pp. 343-358.
14. Niemelä, I., Simons, P., and Syrjänen, T. (2000). Smodels: a system for answer set programming. In: *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning*, Breckenridge, Colorado, USA, April 2000.
15. Reiter, R. (1980) A logic for default reasoning. *Artificial Intelligence*, 13:81-132.
16. Riva, G., F. Vatalaro, F. Davide, M. Alcañiz (eds.) (2005). *Ambient Intelligence*. IOS Press, 2005.
17. Schmidt, A., Interactive Context-Aware Systems - Interacting with Ambient Intelligence. In: G. Riva, F. Vatalaro, F. Davide, , M. Alcañiz (eds.), *Ambient Intelligence*. IOS Press, 2005, pp. 159-178.
18. Schmidt, A., Beigl, M., and Gellersen, H.W. (1999), There is more to Context than Location. *Computers & Graphics Journal*, vol. 23, 19, pp.893-902.
19. Schmidt, A., Kortuem, G., Morse, D., and Dey, A. (eds.), Situated Interaction and Context-Aware Computing. *Personal and Ubiquitous Computing*, vol. 5(1), 2001, pp. 1-81.

CAMPUS NEWS - Artificial Intelligence Methods Combined for an Intelligent Information Network

Markus Maron, Kevin Read, Michael Schulze

Department of Computer Science, Artificial Intelligence Research Group
University of Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz
[maron|kread|schulzemich]@uni-koblenz.de

Abstract: In this paper we describe a network for distributing personalised information with the usage of artificial intelligence methods. Reception of this information should be possible with everyday mobile equipment. Intelligent filtering and spam protection aim at integrating this technology into our environment. Information on the system architecture and usage of the installation are also presented.

1 Introduction

At the start of each semester much information is presented to the freshmen students. Ranging from basic questions like the location of the registrar's office up to course-specific data, each morsel of information must be looked up in a different part of the campus. Senior students look for office opening hours, announcements for specific courses or extracurricular events. Our concept of developing a campus information system seeks to answer these questions in a personalised way, at any time, at any location. The idea is to enable the user to find and access all information that is of relevance to her. All she needs is a Bluetooth enabled mobile device, like a PDA or mobile phone. All used techniques are in themselves not new or unique, but the combination of instant messaging, Bluetooth-centric transport and profile based information narrowcast is novel. Likewise the involved AI methods are well-known but combine to a complete and complex result. On top of that, the system based on a platform made purely for research is in the stage of evolving into a product and is even now being utilised as a public service on-campus.

This information network is only one piece of the puzzle of our view of a ambient intelligence information network. Previous steps done on the Koblenz campus: in a series of projects funded by the EU (Trial Solution) and BMBF (In2Math, [BGOM04]) we developed "Living Books" [BFGHS03], personalized, intelligent teaching material, which is also available for PDAs and smart mobile phones. There is also an approach to use mobile devices for interaction during classroom teaching[1]. Altogether we find a situation on Campus, where students use their mobile device for learning and interacting and for location-based, personalized information.

[1] `www.mobilearn.org`

Research done in the area of mobile information systems include the projects SmartWEB [Wah07] and SmartKOM [Wah06], which allow queries from anywhere, at any time, utilizing natural language and speech recognition. These technologies require state-of-the-art hardware and broadband mobile network connections. Other groups researching applications on mobile personal computers for ambient intelligence have come to the same conclusion as we have, that the main attention with pervasive applications has shifted from a "use anytime, anywhere" perspective to a location-based, personalized view [RMM05]. A lot of work is happening in this area at the moment. Using a Bluetooth mesh for positioning to send data over non-local wireless links like GSM or GPRS is one avenue to take [AGKO04]. In our approach we opted for positioning and transmission over the same channel. The local wireless link can also be skipped completely, which leads to different usage models [FV02]. A bit closer to our usage scenario of a intelligent university environment than these mentioned projects is the project "mobile cafeteria menu"[2], although there are neither location-based nor personalized aspects involved. Our approach is in a certain sense a reduction of all mentioned projects as we do not install software on the phone, which is a fragile process, nor do users incur additional costs.

2 Campus News – Concept

The Campus News System is based on results of the research project IASON[3], funded by the "Stiftung Rheinland-Pfalz für Innovation". Motivated by the development of powerful mobile devices and the semantic web, we defined a *Semantic Mobile Environment*. In such an environment, so-called service nodes are installed at chosen points of interest. These service nodes broadcast messages to nearby mobile users using Bluetooth wireless technology. For example a bookshop could send its latest offers, or the University restaurant could present its menu or a faculty presents the schedule of events to the students. Each message is annotated with a profile, so that end users will only receive messages that are of interest to them.

This semantic annotation is a logical concept in Description Logic (DL) [BCM$^+$03, BHS03]. We also gave the users the opportunity to build their individual interest profile. The first usable prototype of the project (see [Mar05]) was implemented in J2ME, such that the user profile and the inference engine for the personalization was stored in the mobile device.

During several tests in the University and in the City of Koblenz within the framework of the EU project Spatial Metro[4] it turned out that most mobile phones did not yet fulfill our system requirements. They could not access the Bluetooth wireless functions from Java. Apart from that we learned that the barrier to install software on mobile phones or PDAs is higher than with computers. As of today, users aren't used to application installation on

[2] `http://www.studentenwerk-dresden.de/mensen/handy.html`
[3] `www.uni-koblenz.de/~iason`
[4] `www.spatialmetro.org`

these devices and as such distrust the idea. To overcome both the technical shortcomings of mobile devices and the need for application installation, we chose to move the decision process (the ”reasoning engine”) from the mobile phone onto a server, thus eleminating the application. The profile of the user now needs to be entered centrally on a web page. The following describes this solution, which we call ”Campus News Information System”.

2.1 System architecture

The architecture of the Campus News Information System consists of three components (as shown in figure 1): a web application as the user frontend (blue), a server application (red) in the middle and a freely scalable number of service nodes (green) for delivering the information to the mobile devices. We implemented two different kinds of frontends, one for each group of users. We need an administration interface for the users that want to offer information to the public. We call this frontend the Management Console. We also need a user interface for the recipients of the information, in our case the students. This is called the Userweb; it is depicted in figure 2.

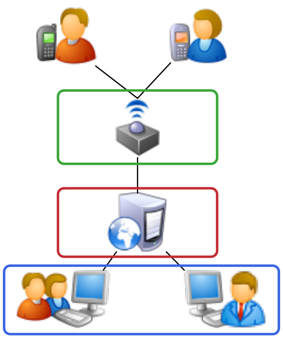


Figure 1: System architecture

Both Management Console and Userweb access the backend, consisting of a relational database and a server application. The database acts as a central storage for message data, profile data and service node information. Both web frontends store any user-made changes here. The server application also accesses the database, but uses this data to drive the service nodes. As soon as mobile devices are recognized by one of the service nodes, the server looks up the profiles of the corresponding users. This lookup uses a combination of several aspects of the mobile phone including its Bluetooth address to ensure that this matching between mobile phone and user is correct. Using a subsumption check on the annotation of the messages and the users' profile, the server decides on which information conforms to the users' interests. In the next step a history query is made to ensure that no mobile device receives the same information twice. All relevant messages are then transferred to the service node by either wireless or wire-bound networking.

The service nodes scan for mobile devices with activated Bluetooth visibility. After handing this information to the server and receiving the messages, they attempt to transfer this data. After two successive rejections by the mobile device no further attempts will be made for a certain duration, to adapt to students that are not interested in the service.

Figure 2: Campus News Management Console (left) and Campus News Userweb (right)

2.2 Content entry

Up to now, the system allowed for content to be entered centrally by trusted administrators as described, creating a one-way flow of information. Content will only be received at the end of the flow, i.e. the mobile device, if it fits into the users' need according to a concept filter as outlined above and explained in more detail in 3.1. This has been realized for the Campus News system and is in daily use. Our requirements for an intelligent communication utility required also peer-to-peer or community-centric communication models. This necessitated a bi-directional and decentralized flow of information, which is now being implemented.

The first step is to enable the service nodes to receive messages in a fixed form that is understood by most mobile devices. Using known aspects of the mobile phone and the Bluetooth hardware, the sender can be identified within the network. There are two ways to route the message. It could be a broadcast message in the style of an announcement, which would then be injected into the central database. Example broadcast messages could be *"The car with license plate X is parked with lights on"* or *"The course Y at 1 AM was cancelled"*. Apart from transfer via Bluetooth, this information could then be displayed on public TV screens as a news ticker or on web pages. One problem area here is to filter out unappropriate or hurtful contents, which will be discussed in 3.2.

The other possibility would be to route this message only to a certain set of recipients, thereby enabling personal messaging systems. This platform could then be used for a whole range of applications in the area of ambient intelligence, utilizing the mobile device as a messaging center that is highly portable and ubiquitous. In the context of the intelligent university environment, personal messages with grades, examination appointments and schedule changes could be sent to individuals. In this scenario, as with most peer-to-peer communcation, information would not necessarily have to be scanned for malicious content.

3 AI methods

In this section we describe the methods of artifical intelligence which are used within the Campus News environment. As mentioned before we are developing an information system which allows bi-directional flow of information. Each direction requires its own methods of filtering and selection to make sure that only wanted messages pass the system and reach the user.

3.1 Concept filtering

As already mentioned we give the users the possibility to choose the topics they want to be informed about. To achieve this goal we developed a mechanism which is called concept filtering, based on DL profiles and ontologies. A similar server-side reasoning process was already used in the Living Book which is a tool for the management of personalized teaching material. For this the KRHyper [BFGH^{+}03], a full first order theorem prover based on the hyper tableau calculus was used. Inspired by the Pocket KRHyper, small version for mobile devices, of the theorem prover, we developed the algorithm for filtering the huge amount of messages which comes within the system.

The reasoning process itself consists of these steps: before transmitting a message to the user, the server has to decide whether the information fits to the users interest or not. This deduction process called matchmaking [KS05] is done by first order reasoning. The user's interests are called a profile. Each message is annotated by its author with a concept. Both consist of Description Logic concepts and are based upon the same terminology. We also built a small ontology for our semantic environment. The decision whether a message matches a user's profile is based on concept satisfiability and subsumption of the DL in use. The task is done by only two queries.

$$\begin{aligned} \mathit{profile} \sqcap \mathit{annotation} &\not\equiv \bot & (1)\\ \mathit{annotation} &\sqsubseteq \mathit{profile} & (2) \end{aligned}$$

If the annotation satisfies test (1) the annotation is *compatible* with the profile. Because an unsatisfiable annotation will be subsumed by every profile, the first test prevents any unsatisfiable annotation to be considered as a match. This test avoids spam. Test (2) will give a better *match degree* for those annotations that are subsumed by at least one of the $positive$ terms. We call these annotations a *match*. This second test is only performed after successfully testing satisfiability (1).

Example

The example shall illustrate the match decisions with respect to a user that is interested in lectures about philosophy and information about the vegetarian menu, but hates sports.

The profile contains the interests $\exists offer.(lectures \sqcap \forall hasTopic.philosophy) \sqcup \exists offer.(menu \sqcap \forall hasDiet.vegetarian)$ and the disinterest $\exists offer.sports$.
(**Note:** a user who is requesting something is interested in offering the same thing.)

On her walk through the campus, the mobile user passes different service nodes and receives the messages listed in Table 3.1.

Message Text **Annotation**
The menseria offers the delicious menu *Lasagne Bolognese* $\exists offers.(meal \sqcap noon)$
and for the vegatarians a salad $\exists offers.(meal \sqcap noon \sqcap \forall hasDiet.vegetarian)$
An extra curricula lecture about Descartes is offered $\exists offers.(lectures \sqcap \forall hasTopic.philosophy)$

Table 1: Example messages

With this profile the messages of the lecture and the vegetarian menu are matched, but the message related to the famous menu *Lasagne Bolognese* is rejected.

3.2 Message filtering

Broadcast messages are displayed on public TV screens or transmitted to all interested mobile users. They are entered by the admins into aforementioned "Management Console". The next step is to enable the users to enter their own messages. Appropriate channels would be either the Campus News web interface or sending messages directly from the mobile device via Bluetooth.

The content entered by the admins is trusted content, since it is a closed and well-known user group. But messages that originated in the user base have to be filtered to ensure that hurtful or malicious content is not displayed, since the admins do not control this content. This problem is exacerbated by the fact that users feel anonymous, although the operator can track identity by means of Bluetooth metadata. These messages therefore cannot be inherently trusted. Admins need tools to classify messages as "spam" or "ham". The system should preferably be able to automatically classify messages after setting a few basic settings and manually classifying a few messages.

Many proven approaches exist for classifying email into the two categories "spam" and "ham". Some of them filter messages according to a list of keywords [Heb01]. If one these keywords appear in the message, it is classified as spam, otherwise as ham. Other methods are based on a statistical approach. A well known method is to use Bayesian filters [Sch04]. Another popular approach is to use decision trees [Bla02]. Instance-based approaches are also in use [Bla02]. When used to filter emails, the layout and email headers can also be analysed. Examples for the use of this metadata would be to categorize emails based on attached figure files or to look into the route the email took according to information in the header.

Format and mode of display for broadcast messages differ significantly from emails. Compared to the well-known form of email spam, our messages are much shorter. Our tests showed that messages should not be longer than 200 characters, otherwise they will not be readable or intelligible, spread out over many screen widths. Our message format does not include headers or layout instructions, so we cannot use this metadata to aid classification. We have a format that makes classification more difficult, but on the other hand there are less possibilities to trick the filter mechanism. All techniques that rely on header information (as with email) are not an option.

Only a few of these methods are of relevance to our project, like the search for keywords, or rather more "key phrases". The admin can edit this list of keywords. For categorization Bayesian filtering, decision trees and instance-based methods are interesting too. All these approaches have to be tested with respect to our specialized situation.

When users or certain mobile devices abuse the system, the admins can blacklist [Heb01] these entities. There is also a whitelist [Heb01], to specify senders that never need to be filtered.

4 Results and outlook

Now, ten weeks after introducing the Campus News System at the University of Koblenz, we are pleased to say that the usage and acceptance by the students is very high for this short time frame. It will be interesting to see if acceptance will climb even higher in the future. The ratio of found devices to devices that received information was at 12.8% in April 2007. This ratio climbed to 47.1% in June. We consider this to be the number of Bluetooth capable devices owned by users willing to activate Bluetooth functionality, divided by the number of Campus News adopters. We detected over 2200 different mobile devices with Bluetooth activated and served 675 of them. These 675 are comprised of 464 unregistered users that received the cafeteria menu and urgent public announcements, and 211 registered users that got news according to the profile they set. All in all we transmitted over 4078 different messages in this short time frame (see table 2). To put the numbers into perspective, the campus Koblenz has around 5000 students. Taking into account occasional visitors, more than one third of all people on-campus have actived Bluetooth and more than ten percent have received Campus News information.

We also did a questionnaire on user wishes and opinion regarding Campus News. We sent out a code and asked the students to enter that code on the answer sheet. On top of that we asked about mobile phone brand and model, opinion of the system in general and wishes or suggestions for future work. Of the 97 students that replied, 12 could not receive the code. Using the stated information about the mobile phone brand we got insight into the workings of Samsung and Motorola brand phones and could increase the compatibility in this area. The opinions varied from general vague acceptance of the concept up to enthusiasm. The most wanted feature was a higher density of service nodes and up-to-date

	April	May	June	since Roll-out (16/04)
found devices	1079	785	1103	2286
served devices	139	163	520	675
transmitted data	828	903	2347	4078

Table 2: Usage of the Campus News System

information in the system for course schedule changes.

The next step is building a pervasive community by extending the system for reception of messages. Combined with the concept of intelligent categorization and thus personalization, this community enables ubiquitous mobile devices to became intelligent information centers.

Acknowledgments

Campus News has been carried out by the cooperative work between the University of Koblenz, Studierendenwerk Koblenz, Arbeitsgruppe Künstliche Intelligenz and wizAI Solutions GmbH and is based on the research work within the IASON project, which is funded by the "Stiftung Rheinland-Pfalz für Innovation".

References

[AGKO04] Lauri Aalto, Nicklas Göthlin, Jani Korhonen, and Timo Ojala. Bluetooth and WAP push based location-aware mobile advertising system. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 49–58, New York, NY, USA, 2004. ACM Press.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BFGH+03] Peter Baumgartner, Ulrich Furbach, Margret Gross-Hardt, Thomas Kleemann, and Christoph Wernhard. KRHyper Inside — Model Based Deduction in Applications. In *Proc. CADE-19 Workshop on Novel Applications of Deduction Systems*, 2003.

[BFGHS03] Peter Baumgartner, Ulrich Furbach, Margret Groß-Hardt, and Alex Sinner. 'Living Book': -'Deduction', 'Slicing', 'Interaction'. In *CADE*, pages 284–288, 2003.

[BGOM04] Peter Baumgartner, Barbara Grabowski, Walter Oevel, and Erica Melis. In2Math - Interaktive Mathematik- und Informatikgrundausbildung. *Softwaretechnik-Trends*, 24(1):36–45, 2004.

[BHS03] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description Logics as Ontology Languages for the Semantic Web, 2003.

[Bla02] Kai Blankenhorn. Spam-Filterung mittels Maschinellem Lernen. Master's thesis, Fachhochschule Furtwangen, 2002.

[FV02] Alois Ferscha and Simon Vogl. Pervasive Web Access via Public Communication Walls. In *Pervasive '02: Proceedings of the First International Conference on Pervasive Computing*, pages 84–97, London, UK, 2002. Springer-Verlag.

[Heb01] Wolf-Guenter Hebel. Spam mail classification system. Master's thesis, University of Georgia, 2001.

[KS05] Thomas Kleemann and Alex Sinner. Description logic based matchmaking on mobile devices. In Joachim Baumeister and Dietmar Seipel, editors, *1st Workshop on Knowledge Engineering and Software Engineering - KESE2005*, pages 37–48, 2005.

[Mar05] Markus Maron. IASON Mobile Application - Konzept und Realisierung einer mobilen Anwendung für profilbasiertes Matchmaking von Nachrichten. Master's thesis, Universität Koblenz-Landau, 2005.

[RMM05] G. Roussos, S. Maglavera, and A. Marsh. Enabling Pervasive Computing with Smart Phones. *IEEE Pervasive Computing*, 4(2), 2005.

[Sch04] Bernhard Schüler. Aggregator - A Tool to Bootstrap Text Classification. Master's thesis, Universität Koblenz-Landau, 2004.

[Wah06] Wolfgang Wahlster. *SmartKom: Foundations of Multimodal Dialogue Systems*, volume XVIII of *Cognitive Technologies*, chapter Dialogue Systems Go Multimodal: The SmartKom Experience, pages 3–27. Springer, Heidelberg, Germany, 2006.

[Wah07] Wolfgang Wahlster. SmartWeb - Ein multimodales Dialogsystem für das semantische Web. In B. Reuse and R. Vollmar, editors, *40 Jahre Informatikforschung in Deutschland*, Heidelberg, Berlin, 2007. Springer.

Searching for Temporal Patterns in AmI Sensor Data

Romain Tavenard[1], Albert A. Salah[2], and Eric J. Pauwels[2]

[1] IRISA/ENS de Cachan, Av. du Gén. Leclerc, 35042 Rennes Cedex
[2] CWI, Kruislaan 413, 1098 SJ Amsterdam, NL

Abstract. Anticipation is a key property of human-human communication, and it is highly desirable for ambient environments to have the means of anticipating events to create a feeling of responsiveness and intelligence in the user. In a home or work environment, a great number of low-cost sensors can be deployed to detect simple events: the passing of a person, the usage of an object, the opening of a door. The methods that try to discover re-usable and interpretable patterns in temporal event data have several shortcomings. Using a testbed that we have developed for this purpose, we first contrast current approaches to the problem. We then extend the best of these approaches, the T-Pattern algorithm, with Gaussian Mixture Models, to obtain a fast and robust algorithm to find patterns in temporal data. Our algorithm can be used to anticipate future events, as well as to detect unexpected events as they occur.

1 Introduction

The success of Ambient Intelligence (AmI) depends on observing the activities of humans and responding to their behaviour patterns intelligently. In ubiquitous environments, where a wealth of sensory data is produced, mining the data for temporal patterns serves this need by discovering associations and structure, either in an offline manner to pave the way for new designs and applications, or in an online manner to ensure adaptation to the user of the AmI environment.

Two things make this task especially challenging. First of all, in a real environment, action patterns that are composed of separate events are interleaved, either by the presence of multiple users, or simply by our habit of doing multiple actions at the same time. Thus, taking an event window to predict the next event in the system will simply not work. Secondly, these patterns exist in different time scales, and the time difference between related events of a single action can have a large variation. Consequently, detecting associations with these patterns becomes a very challenging task, and many traditional pattern analysis methods are not directly applicable as we show in the next section.

In Section 2, a brief survey of the relevant literature is presented, with an emphasis on the more prominent compression-based approaches. The T-pattern method and our proposed modifications to it are presented in Section 3 and Section 4, respectively, followed by our experimental results.

2 Description of the Problem and Related Work

The temporal data we would like to analyze is in form of a sequence of point events, derived from a dense network of non-intrusive and low-resolution sensors. The patterns that we hope to detect are in the form of short event sequences, with additional information about the expected time of each event in the sequence, relative to the previous event. These patterns can then serve for semantic analysis or prediction of events for responsive environments, for instance in scheduling of maintenance jobs or in arming home security systems. The challenge of the problem is the existence of multiple causes (e.g. multiple users of the environment), triggering unrelated events one after the other.

The most straightforward way to detect temporal events is by representing them spatially, where portions of the input feature are associated with increasing time indices. This approach does not work except for the simplest cases, as the absolute positions in a feature vector are not relevant at all.

A more appropriate way of representing time is to make it a part of the model. For instance in recurrent neural networks, the temporal dimension is taken into account with the help of context units [3]. However, recurrent neural networks and related approaches cannot deal with overlapping patterns, they quickly become cumbersome for larger input intervals, and they require lots of training samples.

Markov models have been recently employed to tackle simplified versions of this problem, where there are no action overlaps, and events are generated as one long sequence [6]. These models have three main disadvantages for the problem at hand. First and foremost, the first order Markovian assumption does not hold, as action patterns are construed as sequences of events, and the complete sequence is relevant for the prediction of the next event. Secondly, the estimation algorithms assume that the topology of the HMM-structure is known, which is not the case. Finally, they cannot predict patterns that have long event intervals.

A recent approach involves PCA-based methods to uncover daily human behaviour routines [2]. The data for each subject are stored in an activity matrix, whose most prominent eigenvectors (dubbed *eigenbehaviors*) are then interpreted. One obvious drawback with this method is that it requires a fixed sized activity vector. Additionally, there is no hierarchical decomposition of activities.

Finding a *dictionary* of patterns is possible with **compression-based algorithms** that treat events as "words" in a stream, and seek the patterns that lead to the best compression of the stream. These methods use the Lempel-Ziv compression algorithm, which is known to achieve Markov entropic compression, or a variant of it (e.g. Lempel-Ziv-Welch and Active Lempel-Ziv algorithms) [1].

The basic Lempel-Ziv algorithm (**LZ78**) uses an automatically updated dictionary to extract recurring "words" (patterns) in a string. The Lempel-Ziv-Welch (**LZW**) variant starts off with a pre-defined basic dictionary (in the case of sensor networks these are single sensor events) to avoid ill-detected patterns at the beginning of the stream and to introduce some continuity. The **Active LeZi** uses a sliding window of length l (length of the longest phrase in LZ table) on the stream to extract all possible sequences of size l.

LZW and Active LeZi both aim at adding continuity to LZ pattern extraction, yet they still have linear complexity, which is a beneficial feature for a real-time event detection system. On the other hand, none of the compression based methods take into account the temporal structure of the patterns, as the time delays are not modeled, and subsequently overlapping events may escape detection. For a dense, low-cost sensor network without the identification of event source, this is a major drawback as is clearly borne out by the experimental results reported below. This is the main reason why we turn our attention to *T-patterns* as discussed in the next section.

3 T-patterns

The temporal pattern detection methods mentioned in the related work section ignore the time information, and cast the problem into a simpler representation by retaining only the order of events. In neural network, HMM, and compression based approaches, the emphasis is on predicting the next event, which is not a suitable perspective for an environment with multiple overlapping event sequences.

In the *T-pattern* approach, as introduced and explored by Magnusson, symbolic time series are investigated, where each symbol represents the onset of a particular event or activity, with the principal goal of elucidating possible relationships between pairs of symbols and then building trees of temporal dependencies in a hierarchical fashion [5]. A thorough search is conducted on the training sequence for symbols of an ever-growing dictionary. As the algorithm proceeds, pairs of strongly correlated events joined into new events, and the search is resumed with the expanded dictionary.

Magnusson introduced the notion of a *critical interval* (CI): $[d_1, d_2]$ is considered to be a CI for the pair of symbols (events) (A, B) if the occurrence of A at time t entails that B is more likely to occur in the time interval $[t+d_1, t+d_2]$ than in a random interval of the same size. He then suggested to use the standard p-value to gauge how exceptional the observed frequency of the combination under scrutiny is.

More precisely, suppose the total data stream has length T with N_A and N_B occurrences of A and B, respectively. If we assume (following Magnusson [5]) as null-hypothesis that A and B are independent Poisson processes with intensity (i.e. the average number of events per unit time interval) $\lambda_A = N_A/T$, and $\lambda_B = N_B/T$, respectively. Now, assume in addition that there are N_{AB} occurrences of B in a predefined CI (of length d) trailing each A-event. Notice that under the null-hypothesis the expected number of B-events in a time interval of length d equals $\mu_B = \lambda_B d$. In particular, the probability of not observing a B-event in this CI is therefore equal to $\pi_0 = e^{-\mu_B} = e^{-\lambda_B d}$. The above-mentioned p-value is then computed as the probability of observing at least N_{AB} B-events in the CI, if we assume that A and B are independent. Hence,

$$p = P(N_{AB} \text{ B-events or more} \mid \text{A, B indep.})$$

$$= 1 - \sum_{k=0}^{N_{AB}-1} \binom{N_A}{k} (1-\pi_0)^k \pi_0^{(N_A-k)}.$$

Magnusson suggests as a T-pattern detection scheme to test for every possible pair of symbols of the form (A, B), every possible CI, from the largest to the smallest one, until the p-value is sufficiently small, indicating significance (.05 is a typical upper bound). Note that p will be high for high values of d, which means that short intervals will be favored.

4 The Modified T-Pattern Algorithm

We propose two modifications to the T-pattern algorithm to make it more resilient to spurious patterns, and to make the search for patterns more robust.

4.1 Testing independence between two temporal point processes

The repeated significance testing of the basic T-pattern approach substantially increases the risk of false positives (suggesting spurious dependencies). Applying a Bonferroni correction would be one way to mitigate this adverse effect. In this paper, however, we put forward a more efficient way of testing this independence between A and B which is based on the following proposition.

Proposition 1 *If A and B are independent temporal point process, then*

$$T_{AB} \sim U(0, \tilde{T}_B).$$

In plain language this proposition asserts that if the A and B processes are independent, then whenever an A-event occurs between two successive B-events, it will be uniformly distributed in that interval. Due to lack of space we will not attempt to give a rigorous proof, but it is intuitively clear that non-uniformity of A within the B-interval, would allow a keen observer to improve his or her prediction of the next B-event, thus contradicting independence (for a graphical illustration of this proposition we refer to Fig.2. This therefore allows us to formulate a statistical procedure to test whether A and B are dependent: using the notation established above we compare for each event A_k the time till the next B-event to the current B-interval length:

$$U(k) = \frac{T_{AB}(k)}{\tilde{T}_B(k)} = \frac{B_{k^*} - A_k}{B_{k^*} - B_{k^*-1}}$$

which, under the assumption of independence, should be uniformly distributed between 0 and 1: $U \sim U(0, 1)$. This can be easily checked by any number of standard statistical test (e.g. Kolmogorov-Smirnov). If the null hypothesis (independence) is rejected then it makes sense to start looking for inter-event time intervals (i.e. CI's). This is taken up in the next section.

4.2 Modelling T_{AB} times

The CI detection scheme as proposed in [5] has the drawback that only the first occurrence of B following A is considered. However, if the average occurrence rate of A is relatively high, or if the inter-event time for B is long, this could lead to fallacious associations.

For this reason, we propose to proceed differently. If the above-discussed uniformity test has rejected independence, then we look for the characteristic period by modelling the conditional probability $P(B \text{ at } t + \Delta t \mid A \text{ at } t)$ using Gaussian Mixture Models (GMM). More precisely, all the A-events are aligned at time zero, whereupon all subsequent B-events are plotted. If an A-event tends to induce a B-event after a delay of t time-units, this will show up in this plot as a significant peak. All the non-related B-events will contribute to a very diffuse background. For that reason, we model the B-events as a 2-component GMM. One sharp and localized peak sits on top of the critical interval, while all the other B-events give rise to a flat and broad second component. The standard variation of the sharp peak immediately suggest a value for the width of the CI.

5 Experimental Results

In order to have a simple and realistic experimental setup, we simulate simple interruption sensors in a home or office environment (See Fig.1). We have one or two users of the system generating simultaneous interruption events from a pre-defined event dictionary, which serves as a catalogue of prominent behaviours.

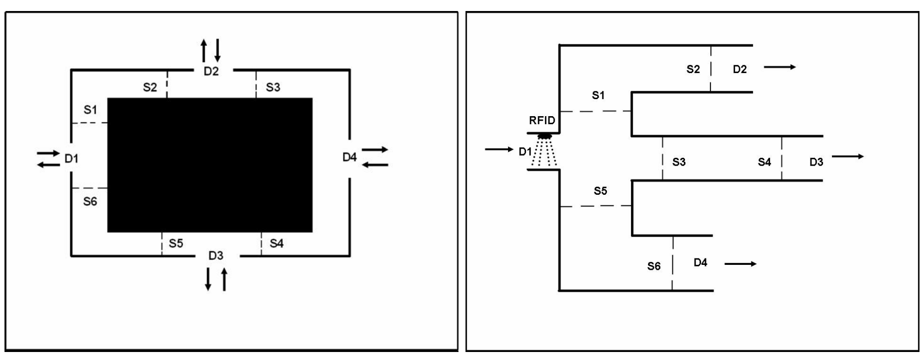


Fig. 1. Ground plan for two corridor layouts used in experiments. Left: Layout 1 shows 4 doors and 6 interruption sensors. Right: Layout 2 shows an entrance door and 3 exit doors, as well as an RFID reader and 6 binary interruption sensors.

For each configuration, we generated training and test sequences of $1,000$ symbols by simulating one or two persons walking in the corridors. We have investigated to what degree we could use the patterns discovered in the training phase as predictors for events in the second stream. The prediction is made for each discrete time slot, which is more granular than predicting the next event. We have contrasted the compression based methods, T-patterns, and our modified T-pattern approach. As the first symbol emitted by each new pattern is random and therefore completely unpredictable, and as individual patterns are short, the prediction rate will have an inherent upperbound.

Fig.3 summarizes the experimental results for the different compression algorithms. In each case the x-axis represents the minimal confidence in the prediction. Confidence is high (in fact 100%) whenever the current pattern unambiguously predicts a unique symbol. If there more potential outcomes, confidence drops accordingly. The dotted line indicates the percentage of cases for which prediction is possible with the confidence specified on the x-axis. The left column show results for scenarios in which only one person is present, the right column shows results for the case when two person intermingle. The two horizontal lines indicate the upper bound for the achievable accuracy (recall that the first symbol in each pattern is unpredictable) and the accuracy of a random prediction.

The results displayed in Fig.4 show contrasts the original CI-extraction (as detailed in [5]) with T-Patterns that use the GMM modelling expounded above. It transpires that Magnusson's original scheme produces too many (spurious) T-patterns making high-confidence prediction impossible as is clear from the way the curves quickly drop to zero. This is most apparent in the 2-person scenario where the intermingling of 1-person patterns generates a large number of new combinations, a fair bit of which are erroneously identified as T-patterns. The GMM approach fares much better, even in the more difficult 2-person scenario.

6 Conclusion

Detecting temporal patterns in sensor data is useful for semantic analysis and event prediction in AmI environments. In this paper we have reviewed two methodologies for the discovery of temporal patterns. The first one collapses the sequence into a string and then uses compression-based techniques to extract repetitive "words". The second one (so-called T-patterns) takes advantage of the time dimension to find the typical delay between related events. We have proposed some improvements to the basic T-pattern methodology (referred to in this text as GMM T-patterns) that significantly improve the performance. Experiments show that T-patterns outperform the compression-based techniques, which is not really surprising as the compression discards most of the temporal information. The experiments also show that the proposed T-pattern improvements (independence testing and GMM-modelling of correlation times) yield more reliable results.

To conclude we summarize the experimental results in Table 1. It was obtained by computing for each experiment the correct prediction rate for a confidence level of 20% (this amounts to constructing a vertical line at the x-value 0.20 in each of the figures and reading of the intersection with the solid curve). The significance of the proposed improvements is obvious.

	Layout 1		Layout 2	
	1 person	2 persons	1 person	2 persons
LZ	29.8	17.7	56.5	13.2
ALZ	21.1	18.8	66.4	19.6
LZW	28.9	22.0	60.5	15.1
T-patterns	28.8	17.1	61.5	24.2
GMM T-patterns	**34.8**	**29.3**	**61.9**	**48.3**

Table 1. Percentage correct predictions at the 20% confidence level.

While we focus on detecting behaviour patterns, a complementary problem would be to track multiple people using low-cost sensors, for which Bayesian filtering techniques are proposed in the literature [4]. The patterns that we aim to detect can serve the tracking problem in constructing a Voronoi graph of the environment. This application is currently inspected by our group.

Acknowledgments

This work was done while R.T. was at CWI. The authors gratefully acknowledge partial support by EC's NOE MUSCLE (FP6-507752), and the Dutch BSIK/BRICKS project. We are grateful to M.S. Magnusson for providing us with additional information on T-patterns.

References

1. Diane J. Cook. Prediction algorithms for smart environments. In Diane J. Cook and Sajal K. Das, editors, *Smart Environments: Technologies, Protocols, and Applications*, page 175. Wiley Series on Parallel and Distributed Computing, 2005.
2. N. Eagle and A. Pentland. Eigenbehaviors: Identifying structure in routine. In *Proc. Roy. Soc. A (in submission)*, 2006.
3. J.L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990.
4. D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian Filtering for Location Estimation. *IEEE Pervasive Computing*, pages 24–33, 2003.
5. M. S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behav. Res. Methods Instrum. Comput.*, 32(1):93–110, February 2000.
6. S.P. Rao and D.J. Cook. Predicting inhabitant action using action and task models with application to smart homes. *International Journal on Artificial Intelligence Tools*, 13(1):81–99, 2004.

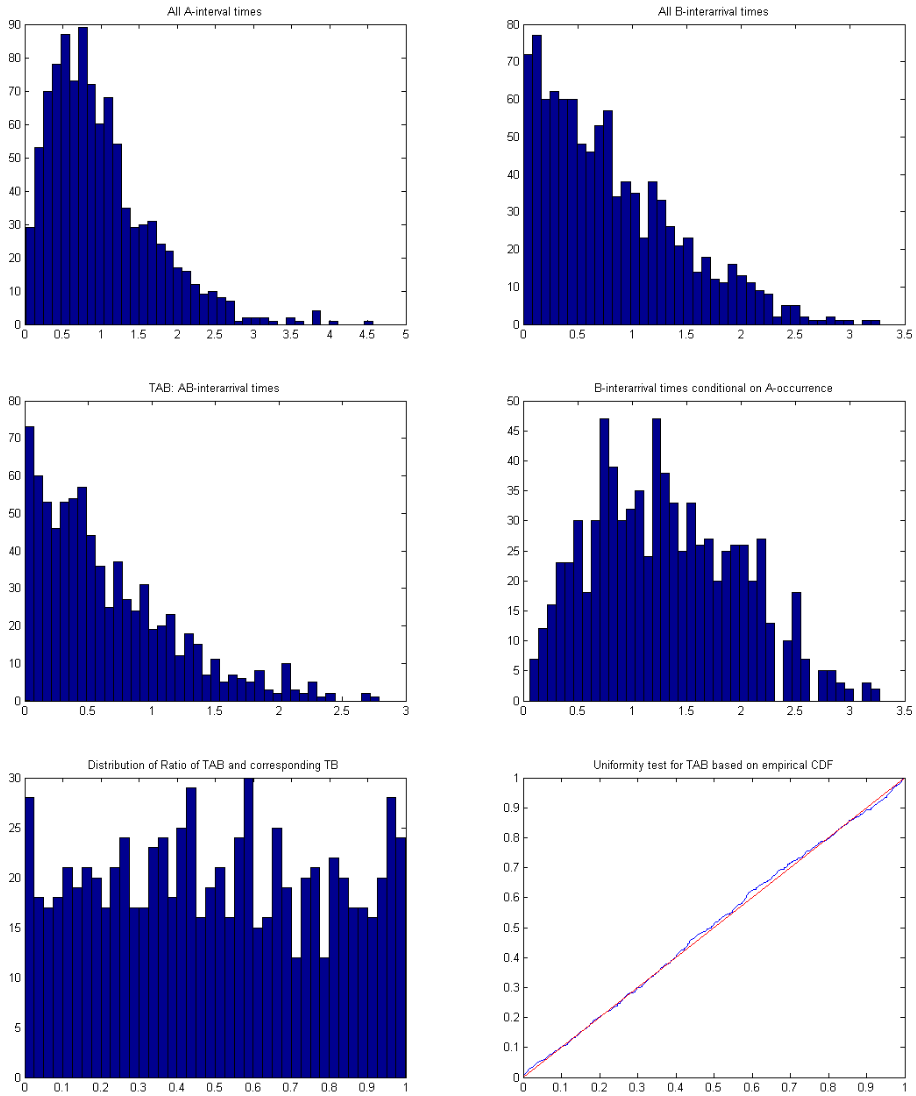
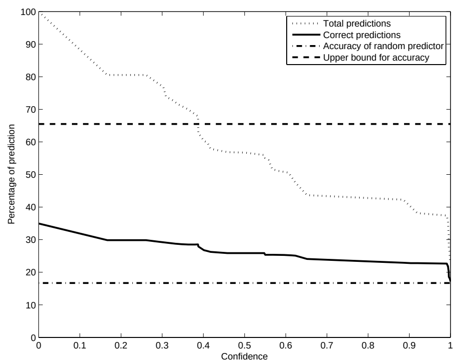
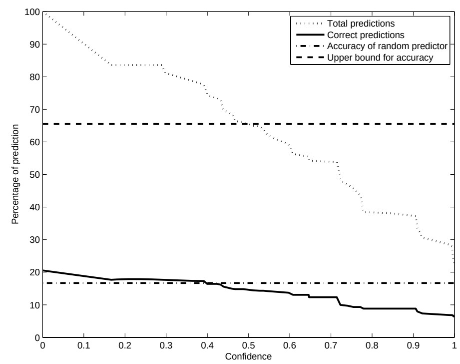


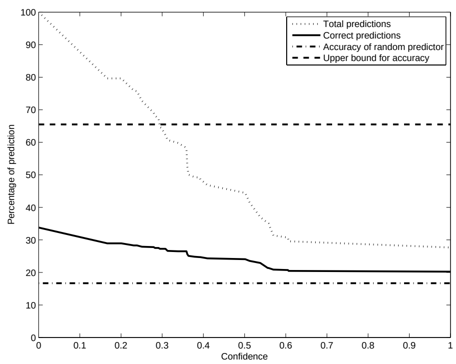
Fig. 2. *Top row:* Histogram for interevent times for the A (*left*) and B (*right*) process; *Middle row: left* T_{AB} distribution: time intervals between the occurrence of A and *next* B-event; *right:* Lengths of B-intervals in which a A-event occurred; notice the bias towards longer intervals (compared to histogram of all B interevent times above). *Bottom row: left:* Histogram of ratio $T_{AB}/\tilde{T}_B$, if A and B are independent, this ratio should be uniformly distributed between 0 and 1, a fact which is even more clearly borne out by its cumulative density function to the theoretically predicted one. (The p-value in this case was 0.61 which means that the null-hypothesis of independence is accepted.)



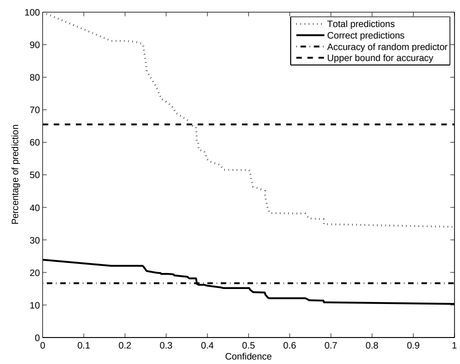
(a) LZ - 1 Person



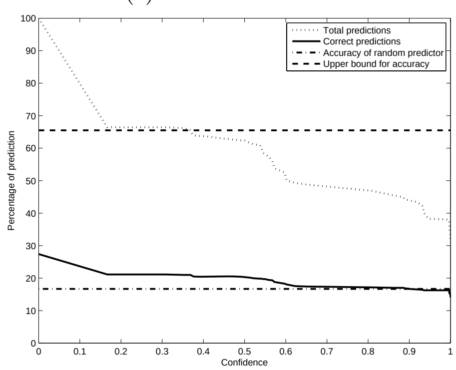
(b) LZ- 2 Persons



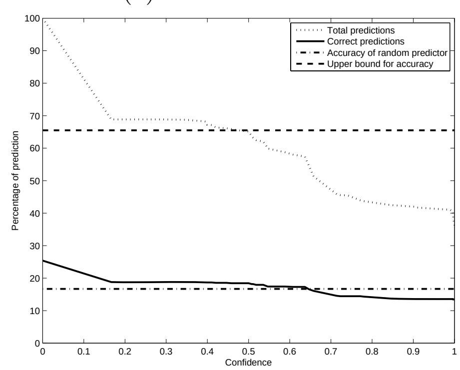
(c) LZW - 1 Person



(d) LZW - 2 Persons

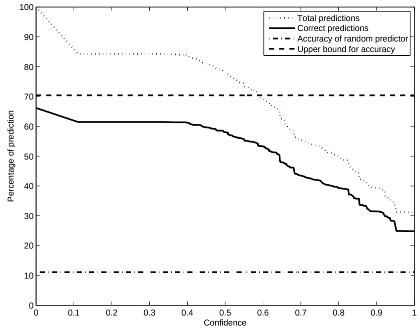


(c) ALZ - 1 Person

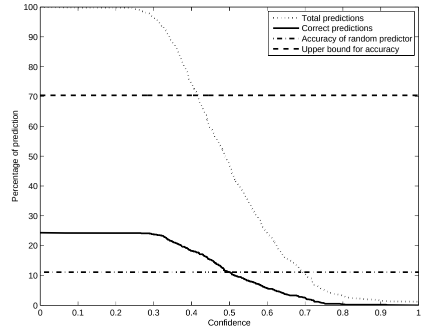


(d) ALZ - 2 Persons

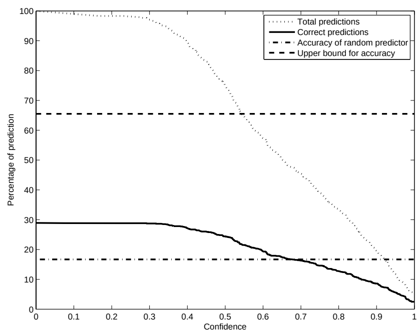
Fig. 3. Predition results for the compression based algorithms for the first layout. Detailed results for the second layout omitted for brevity.



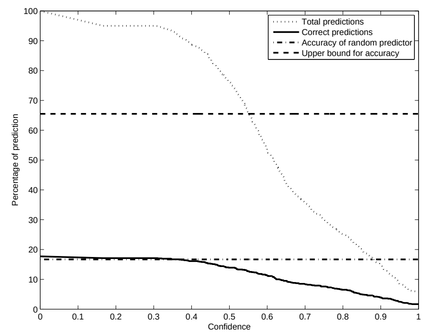
(a) Layout 1 - 1 Person



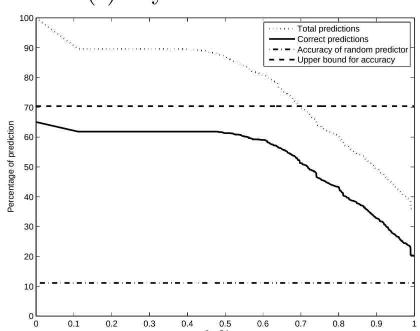
(b) Layout 1 - 2 Persons



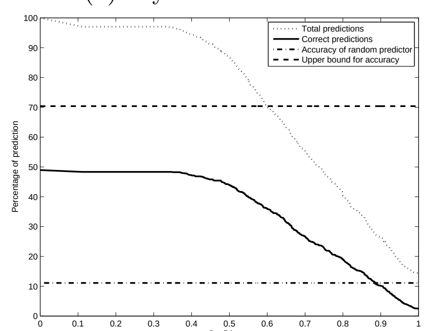
(c) Layout 2 - 1 Person



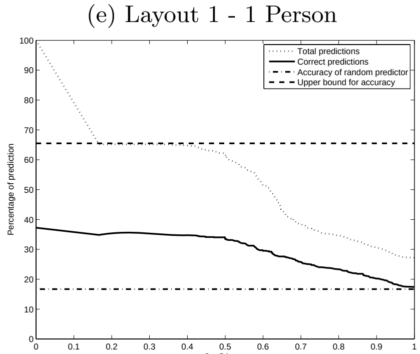
(d) Layout 2 - 2 Persons



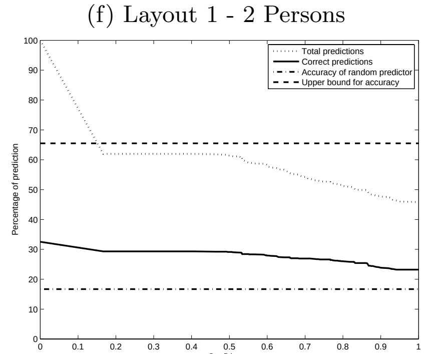
(e) Layout 1 - 1 Person



(f) Layout 1 - 2 Persons



(g) Layout 2 - 1 Person



(h) Layout 2 - 2 Persons

Fig. 4. Predition results for (a)-(d) the T-Pattern and (e)-(h) the modified algorithm