Filtering Algorithm for Agent-Based Incident Communication Support in Mobile Human Surveillance

Duco N. Ferro[1,] , Catholijn M. Jonker[2]

[1] Almende B.V., Westerstraat 50,
3016 DJ Rotterdam, The Netherlands
duco@almende.org

[2] Man-Machine Interaction group, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
C.M.Jonker@tudelft.nl

Abstract. This paper presents an ontology and a filtering algorithm used in an agent-based system to support communication in case of incidents in the mobile human surveillance domain. In that domain reaching the right people as soon as possible is of the essence when incidents occur. The main goal of our efforts is to significantly reduce the response time in case of incidents by proposing and setting up the communication to the right people. Experimental results show that this can reduce the response time by more than 50%, e.g., from 40 to 20 minutes. To continuously improve the accuracy of the proposed communications, the agent-based system uses feedback mechanisms. An implementation of this system, ASK-ASSIST, has been deployed at a mobile human surveillance company.

Keywords: mobile human surveillance, incident management, communication support, collaborative filtering, agent-based decision support

1 Introduction

Efficient and effective communication is critical to timely align human and other resources capable of handling incidents. In domains such as security, crisis management, medical care, the military and traffic incident management, hundreds of individuals distributed over various groups and organizations perform several tasks at different places and move from one location to another over time. Such tasks are by nature urgent, localized and incident prone. As performing these tasks effectively typically depends on the activities and goals of other people, the activities and communication sessions need to be coordinated. Individuals that are not co-located need information and communication technology to coordinate their actions.

The leading case study of this paper is incident management during patrols in the domain of mobile human surveillance (MHS). In those domains patrols are planned in advance but may be disrupted by unforeseen events requiring immediate attention (e.g., in [9]). Managing incidents is complicated by a number of factors. The knowledge, information or support required for dealing with incidents is distributed across the organizations involved. The availability of resources changes over time, and so does the context. In addition, due to organizational and legal requirements, incidents need to be resolved within a certain time limit. Due to these problems, individuals need to initiate and, possibly, anticipate the needed communication on the basis of incomplete and uncertain information.

Currently, central communication points (e.g., dispatch centers) are used to deal with these problems. Human operators at the central communication point respond to requests for support from the field. They propagate such requests across the network to find proper assistance. While this approach is effective, it is not efficient. Due to communication bottlenecks the response time to incidents can be too high [1].

To improve the efficiency of information provision in mobile human surveillance networks, we present an agent-based communication management architecture that is sustained by real-time self-organization. Analogous to recommendation system techniques, which are usually applied to recommend, for instance, books, movies or music to users, we propose an approach that ranks and recommends particular communication pairs/groups to actors that need assistance. For this we use information filters that exploit similarities between actors reporting incidents with those that handled incidents in the past and similarities between the incidents themselves and the contexts in which they occurred.

In this approach all entities in the application domain are associated to their own personal software agent. The idea is that each agent is capable of exploring potential links with other agents in a peer-to-peer manner that exceeds the network of the represented entity itself. Using these links the system can induce, rank and recommend communication groups according to the probability that these groups are capable of handling the incident at hand. Once a recommendation is made to the requesting actor, the result is evaluated using implicit and explicit feedback mechanisms. Implicit feedback is obtained by evaluating the time to solve the incident. Explicit feedback consists of feedback grades provided by the security actors themselves, after the incident is handled. Depending on these evaluations, the strength of the links among the agents is adjusted or new links are created to further improve the support the system offers.

The filtering algorithms and the multi-agent architecture have been implemented in the ASK-ASSIST system to set up context- and incident-sensitive phone and/or conference calls amongst the personnel in our case study domain. Experiments on the data of Trigion, a mobile security company, show that our approach produces prediction schemes that effectively and timely set up the communication network taking into account the context of the individuals and other aspects of the security network (e.g., feedback). The implemented system is currently in use by Trigion.

The remainder of this paper is organized as follows. In Section 2, we discuss related work and its potential in leveraging the communication bottleneck in incident management. Section 3 introduces and formalizes the mobile surveillance domain. The ASK-ASSIST system implementing the filtering algorithms and hosting the agents is described in Section 4. The filtering techniques essential for recommending the right communication are presented in Section 5. We discuss our work and lay out our plans for future work in Section 6.

2 Related Work

The potential of decision support systems (DSS) for incident management is shown in e.g., [11]. Although the literature describes ample work on decision support systems for incident management, research on decisions support systems is in the early stages. To facilitate the decision making process, the use of intelligent software agents, as an intermediate layer, is proposed in [8] and [2]. Hybrid human-agent systems enable such support, for instance, in health care [5].

The communication bottleneck is related to the problem of finding the right coalition of people. Robust matchmaking or coalition formation appears to occur at critical agent network scales. For example, in pair partnership matching an agent is satisfied with a coalition of itself and only one other agent as soon as a specific threshold of a value function is met or passed at a critical scale [7]. The effectiveness and efficiency of such groupings and, in particular, the value functions have to be accounted for and empirically tested.

In this paper we propose to use filtering techniques to find the most promising coalition to meet the requirements of the incident and the agent reporting the incident. In literature, different types of filtering techniques can be found:

- Content-based filtering [6] ,which allows the matching of an agent to an agent coalition. A corresponding task can be allocated to alleviate an incident on the basis of the similarity between an agent coalition given an incident and those of interest to one agent given the specific incident.
- Collaborative filtering, either memory-based [4] or model-based [10], which allows matching of an agent onto an agent coalition. A corresponding task can be allocated to alleviate an incident on the basis of the similarity between the coalition formation profile of an agent and those of other agents.
- Collaborative content-based filtering [3], which combines the above.

Although these techniques already allow filtering incident management data, to the best knowledge of the authors, no literature on collaborative content-based filtering techniques for this domain can be found. To solve the communication bottleneck we propose to use collaborative content-based filtering.

3 Mobile Human Surveillance and Its Formalization

In this section, we present a real world case study of the mobile human surveillance domain. First we illustrate our case study by showing some examples of the activities in the case domain. Then, we present a formalization of this domain, which will be used in the following section to describe the filtering algorithms. The main categories in the formalization refer to actors, shifts, communication, and incidents.

3.1 Introducing MHS security

In the case of mobile surveillance, the security company plans frequent visits (i.e., the number and the nature of the visits are specified in the contract that was agreed on) by security guards to their client premises to deter and, possibly, observe inappropriate actions. While on patrol a security guard has to move by car from one location to another. Once the guard arrives at a location, there may be one or more tasks to perform (possibly in a specific order). A typical course of events during a work shift of a security guard does not only include the acts of transportation and performing location specific tasks, but also frequent contacts over the phone with, for example, an operator of the dispatch center or with a team leader. This communication need is particularly important if an incident such as a fire alarm occurs.

Now, assume that a member of the dispatch centre confronted with an alarm occurring at object 643221, situated at route 240, initially tries to assign that alarm to route 240. Suppose that after 5 minutes route 240 still does not respond (or actually refuses to perform the alarm check). The assignment task is then delegated to a team leader. The team-leader calls the guard at route 245. That is the route closest to the alarm. The guard at this route does not respond. After 15 minutes, route 275 is requested and responds positively. The guard at route 275 goes to the object and arrives 20 minutes later. So after 40 minutes there is a guard present at the object. In simulated experiments, we show that the arrival time can be strongly reduced.

Having illustrated surveillance security, we formalize the entities, logistic network, communication network and incidents for the mobile human surveillance security case.

3.2 Formalization of MHS security

While modeling the entities in the domain of mobile human surveillance security, we make a distinction between two types of entities. The first type of entities we call actors (e.g., guards). Actors are ascribed individual preferences and are capable of exerting these preferences to when and which groups are formed.

Definition 3.1 (MHS Security Actors). *The set of actors A is the union of mobile surveillance security guards G, team-leaders TL and dispatch centre operators D.*

The second type concerns passive entities (e.g. tasks). Typically, passive elements represent contextual information such as a location where an actor is at or a task that the actor is performing. We define the passive entities as follows:

Definition 3.2 (MHS Security Passive Entities). *The set of passive entities P is the union of:*

1) *$\mathcal{T}$, the set of tasks:* {*opening_round, closing_round, regular_surveillance*}
2) *R, the set of route identifiers such that $R \subset \mathbb{N}$,*
3) *O, the set of identifiers for a security object (e.g., for a bank or a supermarket),*
4) *DEVICE, the set of communication device identifiers (e.g., for a PDA device),*
5) *INCIDENT_TYPE:=*{*burglary_alarm, fire_alarm, assault, medical_alarm*},
6) *S, the set of work shifts identifiers, where $s \in S$ is a tuple $<r, t_{start}, t_{end}>$. such that $r \in R$, identifying the route number of the shift, and $t_{start} \in \mathbb{R}$, denoting the start the time window of the shift, and $t_{end} \in \mathbb{R}$, denoting the end of the time window of a shift,*
7) *I, the set of incident identifiers such that $i \in I$* is a tuple *$<incident\_type, o, t>$,*

such that incident_type $\in$ *INCIDENT_TYPE, identifying the type of incident, and o* $\in$ *O, identifying the object related to the incident, and t* $\in \mathbb{R}$*, denoting the start of the incident,*

8) *C, the set of communication meetings(i.e. calls and conference calls), where c* $\in$ *C is a tuple* <n, t_{start},t_{end}, *REASON*>, *such that n* $\in \mathbb{N}$, *a set of unique identifier for each call,* $t_{start} \in \mathbb{R}$, *denoting the start the time window of the call, and* $t_{end} \in \mathbb{R}$, *denoting the end of the time window of the call, and REASON* $\in$ *I* $\cup$ {regular}, *denoting whether the call is considered regular communication or related to incident I,*
9) *GRADE := {1,..,9}, the set of feedback grades on incident support provided.*

Among the different elements, either active or passive, different relations exist. In section 3.1, we have illustrated the MHS security domain. Essential to mobile surveillance are the security patrols on each work shift. A work shift is a sequence of visits preceded by a login and followed by a logout. Formally we define a work shift as follows:

Definition 3.3 (MHS Security Work Shifts). *A mobile human surveillance security work shift w is described by a member of the union set W containing planned shifts, active shifts and finished shifts, such that:*
1) *planned_shift* $\subseteq$ *S × login × (planned_visit)* × logout,*
2) *finished_shift* $\subseteq$ *login × (finished_visit)* × logout*
3) *active_shift* $\subseteq$ *login × (finished_visit)* × [active_visit] × (planned_visit)*× logout*

Generally, we will be interested in the state of the work shifts at some time point. We define a function *state: S ×$\mathbb{R}$ → W,* which allows us to retrieve the current state of a work shift. The mean travel time between two objects is defined in Definition 3.4:

Definition 3.4 (MHS Security Travel Time). *The mean travel time* $\overline{\delta}^{travel}$ *from one object* o_1 *to another* o_2 *is defined by the following function:*

$$\overline{\delta}^{travel}(o_1, o_2) = \frac{\sum_{v \in V_{finished}} t_{arrival} - t_{goto}}{|\, finished\_visit \,|} \qquad (1)$$

In the next section, we show what specific problem we provide a solution for in this paper.

3.3 Communication support for alarm handling

An important problem in mobile surveillance is the assignment of route to alarms (see previous section). This process can be very time consuming. The average of the amount of time it takes for a team-leader to assign an alarm can be estimated using log data. We determined the alarm assignment delays using the log data from 9 months. On a total of 12694 alarm assignments in that period, the average assignment time is about 19 minutes. The amount of alarms that is assigned after 10 minutes is 75.6%. About 12.7 % of the alarms are assigned after one hour.

To speed up this process, we developed a dynamic conference call that allows the personnel to setup a multi-party phone call. Given a request for support request_support($a_{incident}$,<*incident_type*,$o_{incident}$,$t_{incident}$>,t_0), the problem is how to setup a call setup_incident_call($a_{incident}$, <*incident_type*,$o_{incident}$,$t_{incident}$>,{a_1,…,a_n}, t_{setup}) most suitable for handling an incident of type incident_type at object $o_{incident}$ for this specific actor $a_{incident}$.

Having defined and exemplified different element of the mobile human surveillance security environment and the problem focus, in the next section we discuss the MHS incident communication support problem in more detail and show how multi-agent support can improve the performance in this environment.

4 The Ask-Assist system

To increase the speed of handling an incident, we propose an agent-based incident communication support system: Ask-Assist. The system consists of multiple agents. Each agent is mapped to either an actor or passive entities as defined in the previous section, see figure 1.

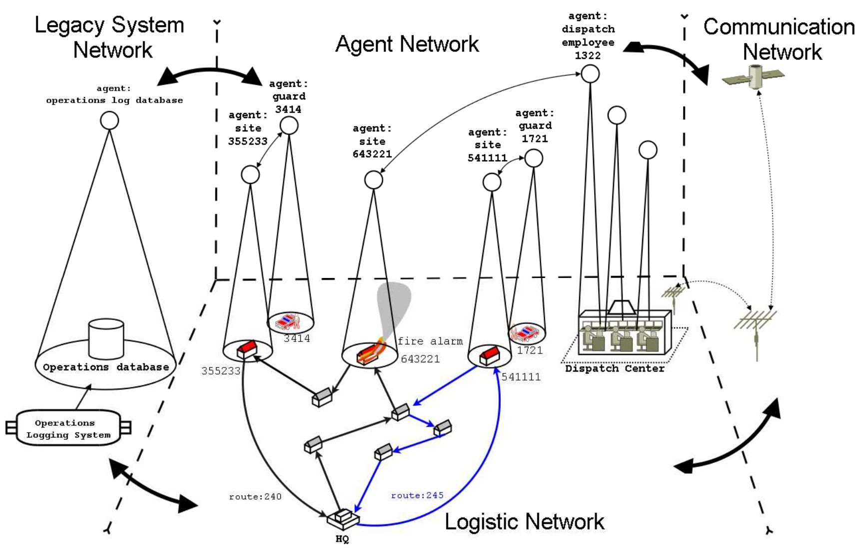


Fig 1. A schematic overview of the mobile security environment and its mapping to the agent layer

Definition 4.1 (MHS Security Agents). *Let $\mathcal{A}$ be the set of agents and represent: $\mathcal{A} \rightarrow (A \cup P)$ a function that maps each agent $\alpha \in \mathcal{A}$ to an entity in the environment. For example, we can each of the following type of entities to a corresponding agent:*

- *Operations log database agent α_{log}, where represent(α_{log}) = operations_log_system, and*

- *Guard agents α_g, where represent* $(\alpha_g) = g \mid g \in$ G

These agents represent the entities enabling the system as a whole to explore and extend potential communication connections in the real world. Agents recommend connections that may not exist in the network on the basis of their analyses of the data available on real world phenomena.

Each agent has different capabilities with respect to processing data that is retrieved on real world processes. In the next section, we will discuss the functionality of setting up a multi-party phone call for incident (i.e., alarms) handling by matchmaking based on logged data.

5 Algorithm for self-organizing incident communication support

We propose a collaborative content-based filtering approach to recommend communication when mobile human surveillance personnel are confronted with incidents. More specifically, we focus on the handling of alarms as described in section 3.3.

Both collaborative and content-based filtering are typically used in e-commerce recommendation systems to elicit user preferences on things like books, movies or music. But also in research there has been a growing interest in filtering techniques as an intelligent mechanism to deal with large amounts of data.

The idea is that in order to predict the preferences (e.g., by user ratings) of a set of users U over a set of items I, the filtering mechanism exploits either similarities between the preference schemes of users over items or the similarities between the items. If we define a utility function v measuring the value of an item i to a user u, i.e., v: $U \times I \rightarrow \mathbb{R}$, then to each user $u \in U$, we would like to recommend item $i \in I$ that maximizes the user's utility:

$$\forall u \in U, i = \arg\max_{i \in I} v(u,i) \qquad \textbf{(2)}$$

Similarly, we define the problem of finding the right person(s) $a \in 2^A$ to contact in case an alarm incident, $<\text{alarm\_type}_{\text{incident}}, \text{o}_{\text{incident}}, \text{t}_{\text{incident}}> \in \mathit{INCIDENT}$ is notified by a dispatch operator $d \in D$ and requests support, as follows:

$$\forall incident \in INCIDENT = \arg\max_{a \in 2^A} v(incident, a) \qquad \textbf{(3)}$$

where v is utility function, INCIDENT $\times\ 2^A \rightarrow \mathbb{R}$ that measure the utility of setting up communication with one or more persons a for an incident. While commonly most filtering algorithms are based on ratings provided by the users, we identify some additional factors that determine the utility of the support that is recommended.

5.1 Factors for predicting the success of alarm handling

We define a number of factors that are used to determine the expected success of communication aimed at handling alarms in mobile human surveillance:

- Incident similarity

The similarity between two incidents i_1 and i_2 is dependent on the nature of incident. This can be considered as a simple content-based similarity measure. We provide a function δ^i: INCIDENT × INCIDENT → [0,1] that allows us to propagate the utilities of handling different incident types. The function δ^i is described by the following table based on an expert opinion:

Table 1. Similarity table of the different incident types in mobile human surveillance

incident_types	*burglary_alarm*	*fire_alarm*	*assault*	*medical_alarm*
burglary_alarm	1.0	0.5	0.8	0.2
fire_alarm	0.5	1.0	0.6	0.3
assault	0.8	0.6	1.0	0.7
medical_alarm	0.2	0.3	0.7	1.0

- Explicit feedback

Each individual that has acted on a particular recommendation is contacted at a later time point to provide feedback on the systems recommendation, i.e., a number in the range of 1 to 9. This is a rating that measures whether the right people where suggested for handling the incident at hand., i.e., r:D × *INCIDENT* × *A* → *GRADE*, where a dispatch operator $d \in D$ evaluates the utility of actor $a \in A$ in handling an *incident* ∈ *INCIDENT*. To enrich the utility space of the feedback provided, we define a normalized feedback function φ: D× *INCIDENT* × *A* × 2^D→ [0,1] taking into account the ratings of a set of dispatch operators(i.e., the neighborhood), as follows:

$$\varphi(d,<q,o,t>,a,D') = \forall d \in D' \frac{\sum_{<q',o',t'>\in INCIDENT} r(d',<q',o',t'>,a)\times\delta^a(d,d')\times\delta^i(q,q')}{\sum_{<q',o',t'>\in INCIDENT} |\,\delta^a(d,d')\times\delta^i(q,q')\,|} \quad \textbf{(4)}$$

where δ^a: *D* × *D* → [0,1] is a similarity function between two guards and $q \in$ INCIDENT_TYPE the type of the incident. We implemented δ^a by the standard Pearson correlation ρ [4]. The ratings of all guards in the guard agents neighborhood D' are weighted according to δ^a and δ^i.

- Experience

Some security objects are visited by particular guards more than other guards. We argue that these guards are therefore more experienced with these objects. We define the experience of an actor $a \in A$ in dealing with a particular type of incidents $q \in$ *INCIDENT_TYPE* with respect to a particular security object $o \in O$ and a $decay \in \mathbb{R}$ as weight for the recentness of the alarm handling as a function $\boldsymbol{\varepsilon}: A \times INCIDENT\_TYPE \times O \times \mathbb{R} \times 2^O \rightarrow [0,1]$, taking into account the experience of a at similar object in the set 2^O, described by the following algorithm:

Algorithm 1: $\boldsymbol{\varepsilon}\,(a,q,o,decay,O') \rightarrow [0,1]$

```
t0 = currentTime()
total = 0
result = 0
for ∀o' ∈ O' do
  for ∀incident ∈ INCIDENT do
    if leave(a,<r',t_start,t_end>,<q',o',t_i>,t_j) ∈ incident
    then
          result += (2^(-(tj-t0)/decay)) × δ^ε(o,o') × δ^i(q,q')
    endif
    total+=1
  endfor
endfor
return result/total
```

where $\delta^{\varepsilon}: O \times O \rightarrow [0,1]$ is similarity function between two objects in terms of the guards by which they are visited. We implemented this by a kendall correlation τ on by ordering the guards by visiting frequencies to a particular location weighted by the recentness of the visit.

- Local rerouting costs

The rerouting of a actor $a \in A$ to an incident location $o_{incident} \in O$ is determined locally by a function $\gamma\ A \times O \rightarrow [0,1]$:

Algorithm 2 : $\gamma(a,o) \rightarrow [0,1]$

```
t0 = currentTime()
r ← onRoute(a)
active_shift_r ← state(<r,t_begin,t_end>,t0) | t_begin < t0 < t_end
o_from ← current_position(active_shift_r)
return 1/δ^travel(o_from,o_incident)
```

5.2 Recommendation and adaptation cycle

Setting up communication support involves a number of steps. First, the system needs to be initialized in to order directly provide the support in the field. This is realized by bootstrapping some part of the already available past data and initializing the neighborhood D' of each dispatch operator agent and the neighborhood O' of each object agent.

After having bootstrapped the agents, the following algorithm is used to produce a recommendation. This happens when the system receives a request for support (request_support).

Algorithm 3: SUPPORTALARMINCIDENT (request_support) → setup_incident_call

description*: this algorithm determines what call should be setup up, when an request for support for handling an alarm incident comes in.*

input: request_support($a_{incident}$,<$q_{incident}$,$o_{incident}$,$t_{incident}$>,t_0):

output: setup_incident_call($a_{incident}$, <$q_{incident}$,$o_{incident}$,$t_{incident}$> ,{a_1,…,a_n}, t_{setup})

candidate[] <- ∅
for $\forall a' \in A \mid a' \neq a_{incident}$
  candidate[a'] = φ($a_{incident}$, <$q_{incident}$,$o_{incident}$,$t_{incident}$>,a',D')
  candidate[a'] *= ε (a', $q_{incident}$,$o_{incident}$, *decay, O'*)
  candidate[a'] *= γ(a', $o_{incident}$)
end for
$candidate_{sorted}$ ← **sort**(candidate) **by** value
group_to_call ← ∅
for j = 0; j < **n-best**; j++ **do**
  group_to_call ← group_to_call ∪ candidate[j]
end for
return <$a_{incident}$,<$q_{incident}$,$o_{incident}$,$t_{incident}$>,group_to_call,t_{now}>

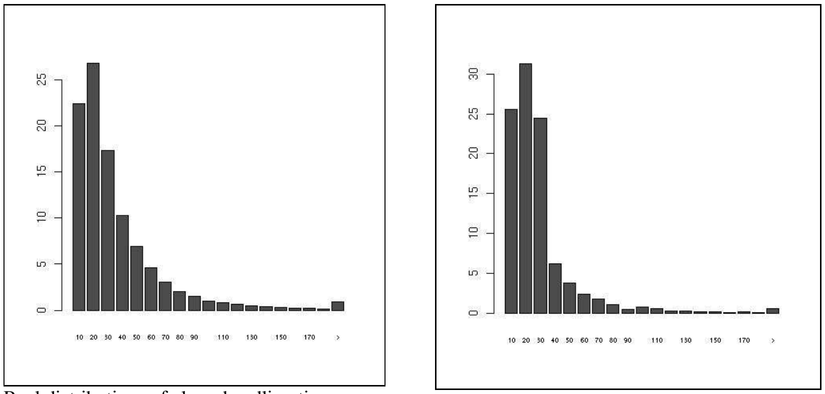
where for a particular incident the utility for each possible actor is determined and the *n-best* are picked for a conference meeting.

5.2 Learning by iteration

Recommendations will result in particular behavior. First, implicitly, the new log events will show whether the recommendation was justified or not. Those new log events are processed and then utilized for future recommendation. In addition, explicit feedback may be received. These are processed as well, such that the agent links can be updated by replacing peers with low correlation or adding new peers that have a higher correlation. By allowing constant adaptation and a growth of incident related data, sparse events will be better handled as the amount of support request grows.

5.3 Experimental Results

We evaluated the potential of our filtering approach, which gave us some promising preliminary results. In the setting without any support, 62.45% of the alarm incidents is handled within the time limit of 30 minutes. By simulating the security domain based on a part of the logged data (n-best=3, decay=90 days), we found that for a bootstrapping period of 6 months, 72.2% could arrive within 30 minutes. Training the system on 9 months of log data gives a simulated result of 78.78%(see figure 2)



Real distributions of alarm handling times

Simulated result when using Ask-Assist

Fig 2. Distributions of the times to alarm handling.

6 Discussion

In this paper we present a filtering approach for handling incidents in mobile human surveillance. Ask-Assist, the system that implements these algorithms, offers support to personnel of the Trigion security company. For this paper we concentrate on alarm incidents. Information on other incidents can be obtained from the authors. In simulated experiments on logged data made available to us by Trigion, we find that we can increase the amount of incidents that is handled in time (30 minutes) from 62.45% to 72.2% with an acceptable amount of training data. This means that roughly an extra 10% of the incidents, currently heavily delayed can be handled within 30 minutes. The results will be analyzed more thoroughly in future work, especially the dependency on different values for parameters such as *n-best, decay.*

Currently, a prototype of the system is being used by a security company in the Netherlands. The amount of human actors involved is around 60 guards and 5 team-leaders concerning 1318 sites. Ask-Assist has been developed by Almende B.V on a specific branch of its Common Hybrid Agent Platform(CHAP).[1]

In the future, we intend to evaluate the field study that is currently ongoing. In addition, we have an interest for the evolution of communication among the personnel and, specifically, for informal communication networks and their influence on performance. When confronted with incidents people tend to rely on their social

[1] see http://sourceforge.chap.net and http://www.groovyactors.com

network. Supporting this could be an interesting way to enhance incident management.

Acknowledgment

The authors would like to thank Trigion, especially, Ron Knaap, Rick Fenne, Harry Visser, the team-leaders of the region of Amsterdam, the guards in the region of Amsterdam and the alarm service centre. Great appreciation also goes out to Alfons Salden and Marian Verhaegh. Furthermore, the authors would like to thank ICIS and the Ministry of Economic Affairs in The Netherlands for funding this research.

References

1. Boy, G. A.(1998), Cognitive function analysis for human-centered automation of safety-critical systems, in Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '98), pp 265–272, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.
2. Boy, G.A.: Theories of Human Cognition: To Better Understand the Co-Adaptation of People and Technology, in Knowledge Management, Organizational Intelligence and Learning, and Complexity. In: Kiel, L.D. (ed.) Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford, UK (2002)
3. Basilico, J. and Hofmann, T. (2004) Unifying collaborative and content-based filtering, *Proceedings Twenty-first International Conference on Machine Learning (ICML'04)*
4. Breese, J., Heckerman, D., and Kadie, C.(1998) Empirical analysis of predictive algorithms for collaborative filtering, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp 43–52
5. Niemann,C. and Eymann,T. "Beyond Automation: Hybrid Human-Computer Decision Support Systems in Hospitals," *ease*, pp. 46-53, Fifth IEEE Workshop on Engineering of Autonomic and Autonomous Systems (ease 2008), 2008
6. Rocchio, J.(1971), Relevance feedback in information retrieval, in The SMART Retrieval System: Experiments *in Automatic document processing*, pp 313–323, NJ: Prentice-Hall, Englewood Cliffs
7. Sarne, D. and Kraus, S.(2004), Time-Variant Distributed Agent Matching Applications, *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pp 168–175, IEEE Computer Society, Washington, DC, USA
8. Storms,P.(2004) System of Systems Architecture. In *Proceedings of the First International Conference on Information Systems for Crisis Response and Management (ISCRAM 2004), Brussels, Belgium*, May 2004.
9. Streefkerk, J.W., van Esch-Bussemakers, M.P. and Neerincx, M.A. (2007). Context-aware notification for mobile police officers. *In: D. Harris (Ed.): Engin. Psychol. and Cog. Ergonomics, HCII 2007, LNAI 4562* , pp. 436–445. Berlin: Springer-Verlag.
10. Wettig, H., Lahtinen, J., Lepola, T., Myllyämki, P., and Tirri, H.(2003), Bayesian Analysis of Online Newspaper Log Data*, in Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT-W '03)*, p. 282, IEEE Computer Society, Washington, DC, USA
11. Zografos, K.G., Androutsopoulos, K.N., Vasilakis, G.M. (2002), "A real-time decision support system for roadway network incident response logistics", *Transportation Research Part C*, Vol. 10 pp.1-18.