

# Self-organizing mobile surveillance security networks

Duco Ferro and Alfons Salden  
Almende BV  
Westerstraat 50  
3016 DJ Rotterdam, The Netherlands  
+31-10-4049444  
{duco,alfons}@almende.org

## ABSTRACT

We develop and evaluate ASK-ASSIST, a decision support system that enables automatically setting up communication networks and sharing knowledge amongst security personnel, in particular when they are confronted with a series of unexpected incidents while on patrol. Currently, team leaders of mobile security guards are still forced to assess the consequences of a series of incidents and to take decisions in a heuristic way. On the basis of incomplete and uncertain information they have to determine the security situation, who to bring in contact with whom, and which personnel or which information systems should be lined up in order to resolve the incidents as a whole. ASK-ASSIST, however, automates robust dynamic formation and coordination of the right coalitions of security personnel, such that they support human improvisation in determining the nature of incidents and taking decisions by the coalitions. These coalitions comprise security personnel, communication security infrastructures and knowledge management security infrastructures. ASK-ASSIST is based on a framework for self-organization that generates ranked lists of incident-specific critical (re)configurations of coalitions in which joint tasks are assigned to agents associated with roles, like guards or team leaders. Our main contributions are the grounding and evaluation of the framework underlying ASK-ASSIST. We instantiate, validate and test the system on the basis of real data. These data concern organizational structures reflected in the coalitions and implicit or explicit feedback provided by personnel from a private security company. The provided feedback relates to the daily operational mobile security surveillance processes including those that involve functionalities of ASK-ASSIST.

## Keywords

Coalition formation, Coordination, Robustness, Mobile Surveillance, Incident Management, Reconfiguration, Scale-Space.

## 1. INTRODUCTION

Organizations physically located in urban areas and isolated industrial zones generally hire private security companies to guard their property from theft and vandalism. There are many ways in which security at such locations can be realized. For example, security personnel may be stationed at the different sites of the organization. This is a standard way of surveillance of a site during daytime or in the evening. In addition or alternatively, patrol guards are sent to check multiple client premises during nighttime or week-ends by car. In this paper, we focus on the latter case of mobile surveillance of security objects and in

particular the management of incidents like the occurrence of alarms by offering the guards on patrol support for improvisation.

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. This planning consist of different routes, that is, sequences of locations for each guard and the tasks, such as closing down the building, that are associated with each of these location. In addition, personnel of the security company dispatch center may act upon alarms (e.g., burglar alarms), for example, by sending a guard for further inspection or, ultimately, by notifying the nearest fire department. The requirements with respect to these less routine actions (e.g., the maximum time to arrival at the location of the alarm) may be specified as part of the contractual agreement as well.

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 entail the acts of transportation and performing location specific tasks, but also frequent communication with, for example, a responder of the dispatch center or with a team leader. This communication need is particularly important if an incident occurs. Knowing what to do or who to contact is paramount when confronted with such unexpected events.

Handling incidents is commonly achieved by following documented security protocols and procedures. Due to a lack of transparency and the low quality of those protocols, such procedures are hardly internalized in the organization and offer little help [1] when faced with unexpected events or unanticipated incidents. Therefore, members in the private security organization are compelled to improvise.

In such cases, mobile security personnel, after assessing the situation in-situ, invariably contact team leaders or a responder at the dispatch center. As a consequence, the physical and cognitive work load of all security personnel increases. We are developing a mobile surveillance coalition formation support system, ASK-ASSIST that facilitates improvisation by security personnel themselves in case of incidents or emergencies by improving the efficiency of communication amongst them. It has been suggested that decision support systems (DSS) are helpful in handling incidents [8]. In the COMBINED systems project an architecture is presented for such systems [6]. Although the literature

describes ample work on decision support systems for incident management, research on decisions support systems is in the early stages. Approaches that aim at support the improvisation process have only been recently proposed [10]. To facilitate the decision making process, the use of intelligent software agents, as an intermediate layer, has been suggested [16]. Hybrid human-agent systems enabling such support require particular groupings of, for example, guards and systems.

We believe that given an incident - we consider simply parts of the actual and historic context - ASK-ASSIST could recommend appropriate ranked lists of robust coalitions of security personnel and communication and information systems to assist individual security guards or groups. This way ASK-ASSIST could arrange and, in addition, effectively establish a desired telephone connection or conference call amongst security personnel in case of, for example, a burglary alarm at a bank.

The problem of grouping together individuals in an effective way has been studied in a variety of different settings. For example, two-sided matchmaking tries to bring individuals together into couples [7, 9, 13]. Many situations occur in everyday life where one group of entities must be mapped onto another (e.g., employees/firms, patients/doctors, consumers/products). At first sight, it also seems logical to model the formation of these simple coalitions using bipartite graphs. However, a security guard, for example, needs to contact another security guard via a responder of the dispatch center and, possibly, a team leader for a particular reason (e.g., when certain expertise about a security object is required). Those extra organizational roles and context should be taken into account in the formation process, as the qualities of the coalitions heavily depend on them.

Coalition formation was originally studied in game theory. However, coalitions there are not formed to bring together individuals given a certain context, but rather to suggest possible cooperation between players in a non-cooperative game. Since players in a non-cooperative game can always improve on their collective payoff by cooperating, it was traditionally assumed that players will always form the largest possible coalition. The precise argument in traditional game theory is based on the observation that all games that are put in coalitional form are super-additive. The main concern of cooperative game theory is how to divide the collective gain over the members of the coalition taking into account all possible contexts that might occur. However, in crisis incident management particular contexts can predict the onset of an incident or a disaster, and trigger measures countering them. Thus averaging coalition formation over all contexts is certainly not a panacea for supporting human improvisation in case of incidents or disasters.

Recently, bounded rationality, incomplete information, or communication overhead in large coalitions of agents have been put forward as reasons against forming large coalitions (i.e., against the super-additive property of agent networks). 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. Here the value function reflects, for

example, an agent preference scheme and a critical scale the maximal possible or tolerable level of communication intensity between two agents given their limited storage, communication and processing resources [15]. Similar critical context-dependent temporal dynamic scales can be identified for larger coalitions needed to resolve more complex incidents. In earlier work, a framework was suggested that (i) generalizes existing conceptual or formal matchmaking and coalition formation models and (ii) is used for the implementation of a matchmaking system [17]. The effectiveness and efficiency of such groupings and, in particular, the value functions have to be accounted for and empirically modeled.

Such accounts, and accounts for coalition formation, may come about after collaboratively filtering the logged agent network history of coalition formation patterns at critical scales. The formation of coalitions - given incidents - can be modeled by means of different types of filtering:

- Content-based filtering [12], 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 [18], 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 [2], which allows the matching of an agent to an agent coalition and allocates a corresponding task to alleviate an incident on the basis of a joint similarity of the above.

The kernels inducing incident-specific coalition formation help model improvisation by humans. Robust models can be distilled upon learning the performance of situational context-specific coordination schemes of dynamic multi-agent systems (MAS), including artificial and human agents, viz. [5]. Learning by MAS, in turn, can be corroborated by complying with an appropriate scale-space paradigm [14] that adequately captures system dynamics and evolution and self-organization at critical scales.

For lack of space we refrain from in-depth expositions on our coalition formation framework and multi-scale-based collaborative filtering techniques. Instead, we apply them to an future mobile surveillance scenario (see section 2) given a specific incident on a route of security guards, to find a suitable group of agents and tasks this group has to perform when additional, possibly artificial, agents and agent functionality become available (in this case a coalition formation support system). Next, in section 3 we use our framework to create a model of this dynamic mobile surveillance domain. In this section, we present a formal description of the domain entities in a coalition formation environment. We describe the configurations of coalitions and the evolution of these configurations. In addition, we sketch how to

retain robust context-specific ranked coalition formation models by collaborative filtering. In section 4, we describe the design of the ASK-ASSIST system for improvisation support. Based on a three-layered architecture, we focus on the layer of software agents that intermediates between human actors and the data/information layer. Finally, we will present and discuss in section 5 the results of testing our decision support approach on real data in the mobile surveillance domain.

## 2. FUTURE SURVEILLANCE SCENARIO

In the following scenes we sketch a mobile surveillance scenario in which the need for coalition formation support becomes more and more compelling with the consecutive scenes. After having discussed these scenes, we show how our coalition formation framework can support improvisation of humans resolving incidents by taking advantage of ASK-ASSIST functionalities.

### 2.1 Scene 1: Bob Goes on Security Patrol

Commonly, a security guard starts patrol from a specific site. In general, this site corresponds to the base of the company that covers a region and that hosts a diversity of assets that are used for surveillance in that region, such as vehicles and all sorts of communication devices. So, every regular route starts at the company's base (e.g., to pick up a vehicle) and ends there as well (e.g., to drop off a vehicle). On patrol a guard sequentially visits each location and performs the tasks relevant to this site in accordance with his or her schedule.

*Bob is based as a security guard in the northern part of the Netherlands. His work shift starts at 10 o'clock at night. It is a Monday, Bob and his team receive a weekly work shift update from a new team leader, Alice. She has assigned Bob to a specific route, namely 321. Having finished his cup of coffee, Bob steps into his patrol car. The scheduling application on his PDA indicates that he has to be at a remote complex of hangars in the harbor at 10.45 p.m. for a routine security check.*

*Bob arrives at about 11.00 p.m. He types his arrival-time in on his PDA. Then, he takes the right set of keys with him and checks all hangars. Every hangar is secured with an alarm system. To turn this off, he enters a code, which he can read out from his PDA. When he is done checking, he re-enters the code to reactivate the alarm. Just before midnight Bob has completed the surveillance of the whole complex. He has noticed nothing out of the ordinary. Back in the car, he updates his current status in the PDA and leaves.*

*His next mission is a routine security check of a bank in the urban area near his base. After a ride through the polder, he arrives at 12.25 a.m. Also here, after inspection, it appears that everything is in order. Bob visits four more objects that night. At 5.40 a.m., he returns to his base.*

This course of events reflects the most basic activities of mobile surveillance in the security domain. The next two scenes illustrate more complex events that require inter-human communication in order to deal well with the given situation.

### 2.2 Scene 2: David Assigns an Alarm

Besides performing the scheduled patrols, private security companies are also contracted to react on alarms occurring at a client's property. The security company, then, obliges itself to have a guard present at the alarm location within a certain time interval (say, 30 minutes) as specified in a contract. When an alarm is set off a typical course of events is the following:

*It is a Tuesday, 4 o'clock in the morning. David's working shift at the dispatch center has almost ended, when his attention is drawn to an alarm on his console from one of the many private alarm centers. Apparently, a burglary alarm has gone off at a bank in the service region of the security company. His task is to make sure that a guard is going there as soon as possible. David retrieves all active guards on route in that region. At 4.02 a.m. he decides to call the guard on route 321. It is Bob. Bob refuses this request, because he may not leave his actual location given the security protocol at hand.*

*To delegate the alarm assignment David calls Alice, the team leader for that region, and asks her to find the most appropriate guard(s) on route. It is 4.04 a.m. The first security guard that Alice calls is Carol on route 291 and appears to be available. At 4.06 a.m. she notifies the dispatch center of this. The location at which the alarm occurred is inserted into the schedule of Carol on route 291. Carol arrives at the bank at 4.23 a.m., just in time.*

In this procedure, every alarm that occurs requires the dispatch center to assign (and, possibly, to delegate the alarm assignment to the task leader). This procedure is effective, but not very efficient. The total communication delay increases as the number of alarms increases and as the number of assignments issued by the dispatcher center coordinator that are refused by the guards on route increases. The assignment of alarms to guards is a much recurring process that depletes the physical and cognitive resources of those involved [3]. It requires new planning activities, which may obstruct operational processes and can even aggravate incidents.

### 2.3 Scene 3: Bob Assists Carol

In reality, security guards are confronted with many more contingencies that make it difficult for them to perform their tasks as planned. There is wide range of unexpected events that may occur on patrol. Such events may be illustrated by means of the following slightly more problematic course of events:

*It is 11.25 p.m. on a Wednesday, Carol, on route 291, has just arrived at the hangar complex near the harbor. After updating her PDA, she steps out of the surveillance vehicle. When she looks around, she realizes that, being there for the first time, she has little knowledge of the alarm system on-site. She does not know exact locations of the consoles that are used to operate the alarm. She reasons: "When I enter a hangar, the alarm will go off. If I don't know exactly where the consoles are, I might not be able to put it off or I might be too late." Since the resulting (false) alarm creates all kinds of extra work for the organization, Carol decides*

to contact Alice, her team leader, for finding somebody to help sorting out these issues for this location.

Alice, in turn, is not able to give a specific answer, but realizes that Bob should know. She contacts Bob on route 321 and asks him to immediately call Alice.

Likewise, a security guard may be confronted with incidents that ask for communicating with a colleague who recently visited the same security object (e.g., in case of recent damage to the object like broken windows). Furthermore, she or he may have to handle incidents in which it is necessary to figure out the location of a set of keys or the access code to the premises before he can complete the core security tasks. In all these cases it is essential that the right information is made available, once the right person or support system is found.

## 2.4 Scene 4: Bob Transferred to Carol

The two previous scenes illustrate the claim that a sudden sequence of incidents may lead to an increase in the workload of dispatch center employees and team leaders to undesirable levels. Actors of the latter type manage dynamic mobile surveillance, also in case of incidents. They malfunction and underperform as they are subject to communication overload. It is not surprising that there is an interest in automating delegation and escalation processes from security guards towards dispatch center employees or team leaders as soon as they are threatened by physical or cognitive overload. Another reason for automation is to resolve incidents more rapidly and to prevent disasters. Before we discuss in more detail how automation could be attained, let us briefly illustrate this automated process:

*A Saturday at noon, Bob arrives at a department store. He updates his PDA and he enters the building. He notices that the alarm has been shut off. Since this is not what Bob expected, he likes to sort this out. Instead of calling the dispatch center or team leader - which is still possible -, he calls a support system, ASK-ASSIST. An IVR menu is presented and Bob chooses option 4: last security guard at the current location. ASK-ASSIST uses operational data as context (e.g., Bob's location, choice he made in the IVR menu, history of earlier logged data) to determine the right person to help him in case of this incident. ASK-ASSIST calls out to Carol (a suitable match) and when she answers, Bob and Carol can speak to each other.*

A guard may also be in need of certain keys of or expertise about a particular security object (as in scene 3). The idea is that in such cases ASK-ASSIST uses information about the actual and historic states and contexts of the relevant entities (e.g., keys, vehicles, guards) to determine the right coalition of (human) agents and assigned tasks in order to resolve the incident or at least to provide the information crucial to solving the problem by human improvisation. Thereto, we propose ASK-ASSIST to recommend given situational contexts ranked lists of potential current or future agent coalitions for either retrieving the requested information or setting up communication or audio phone conference amongst security personnel. Such recommendations by ASK-ASSIST are established after properly grounding the mobile

surveillance entities involved and empirically modeling their actual and historic situational contexts.

## 3. MODELING SURVEILLANCE DOMAIN

In this section, we use our framework to model the dynamic mobile surveillance domain as illustrated in the previous section. First, we describe the domain entities in terms of a coalition formation environment. Next we describe the configurations of coalitions of these entities and the evolution thereof. In the last section, we show how collaborative filtering historic, current and future context data in the mobile surveillance domain provides robust context and incident specific rankings of coalition formation models.

### 3.1 Coalition Formation Environment

For any particular coalition formation problem, we say that the coalition formation elements belong to a coalition formation environment associated with the problem domain. In earlier work we have given a formal definition of the concept of a coalition formation environment [17] and its application to mobile surveillance in security.

In a coalition formation environment we make a distinction between two types of elements. The first type of elements we call active elements, (e.g. a buyer, a provider, a patient or an employee). Active elements are capable of exerting particular preferences to when and which groups are formed. The second type concerns elements that are passive (e.g. tasks or real-estate objects). Passive elements are indifferent to what group they belong. Typically, past, current and future contexts are considered passive elements. In many applications, matchmaking or coalition formation aligns and groups individuals (e.g., users) and items (e.g., access code information) such that their configurations meet the preferences and requirements of all the separate entities, being perhaps all possible individuals or any subgroup thereof. The general idea is that active elements, in contrast to passive elements, have and impose preferences and requirements during the coalition formation process. In the domain of mobile security, for example, coalition formation could concern a security guard requesting expertise about a security object, preferably, by the most experienced security guard available. Clearly, this process depends on the contexts, i.e., aspects of the collective of passive elements involved.

If we look to the scenario of the previous section in more detail, we can identify the following active elements (roles) in this security domain:

- Mobile surveillance security guards(MS), and
- Team-leaders(TL), and
- Dispatch center responders (DPC), and
- Dispatch center responders (DPC), and
- Private alarm centers (PAC). and

- Coalition formation support system, ASK-ASSIST (ASK).

Active elements represent the roles that agents can fulfill in a domain. One can imagine that individual agents fulfill different roles over time. Furthermore, new active and passive elements can be introduced in the domain. Last but not least, relations between agents and active elements can be explicated and spelled out [17].

As mentioned earlier, not all elements in the environment need to be active. In our mobile security scenario, there following entities have no notion of preferences:

- The set of all tasks(T),
- The set of all security objects (Obj),
- The set of patrol vehicles (V) ,
- The set of keys for accessing or locking security objects (K),
- The set of security object specific alarms (S) ,
- The set of routes (R), and
- The set of discrete times (t)..

Note that the set of tasks also includes interactive tasks, such as communication. In Scene 4 of the previous section, for example, communication is set up between two security guards by ASK-ASSIST. This means that, in reality, this communication task is labeled with the relevant member of the set of active elements A. These sorts of relations could, for example, also be relevant to the set of security alarms, because of their object specificity. In this work, we do not explicitly model these labels.

### 3.2 Evolution of a Configuration of Coalitions

Having described a coalition formation environment, the goal is not only to generate a configuration of elements, so-called coalitions, that can handle common mobile surveillance circumstances, e.g. daily patrol in Scene 1 of the previous section, but that can also handle unexpected security incidents by providing improvisation support to humans with the advent of newly added elements, e.g. ASK-ASSIST, in Scene 4.

Our research interest lies in showing how the introduction of ASK-ASSIST, taking advantage of a domain-specific collaborative filtering paradigm, facilitates run-time reconfiguration either to tackle security incidents directly or to enhance improvisation by humans. The latter can be accomplished by ASK-ASSIST, for example, by recommending to place the right security personnel in a conference call or to provide the most appropriate security context information or support systems. Before we discuss this paradigm, let us make the concept of a configuration explicit in terms of our coalition formation environment and additional structures.

In the literature, the concept of a coalition configuration appears frequently, although under different names. In combinatorics the coalition configuration is known under the name matching or perfect matching. Cooperative game theory uses the term coalition structure. In earlier work, we defined a configuration as a grouping of agent roles performing a set of (dependable)

compositions of tasks in specific contexts. A configuration can consist of elements of the environment, such that each element only occurs exactly once at the same time in a configuration.

In order to further structure a configuration we model role groups. A role group is a subset of active elements in A that performs a collection of tasks in T in a specific group context, i.e., a subset of *passive elements* B. This implies that an agent coalition consists of group roles in a given group context assigned to a corresponding collection of tasks. We represent this assignment in terms of a task assignment function  $\gamma: 2^A \times 2^B \times t \rightarrow 2^T$ . Thus, a coalition of agents in context at discrete time performs a collection of tasks.

For our mobile surveillance domain we formally denote a configuration as:

**Definition 1** Given a coalition formation environment M with agents in A, where the roles agents can fulfill in A and passive elements B as contextual elements, a configuration is defined by a collection of coalitions, in which each coalition is represented by the time-ordered composite task assignment to roles in a context,

labeled by a route number, as  $\gamma^{r \in R} = \gamma_{t_n}^{A_n B_n} \circ \dots \circ \gamma_{t_0}^{A_0 B_0} |r$ . Here  $\circ$  is a composition rule consistent with a non-commutative operator or task algebra.

Particular configurations may not be effective or possible at all. Different types of constraints, i.e., those spelled out by the task algebra, limit the set of all possible combinations of coalitions, ergo non-commutative. We define the skill set of configurations  $\Gamma$  in the environment as the collection of possible combinations of coalitions for a specific route in the environment. For the routes  $r \in R$ , a configuration  $\Gamma_0: \gamma_i^r \cup \dots \cup \gamma_j^r$  is a member of the skill set. Furthermore, not all configurations of the skill set perform equally well considering future security requirement levels. In general, there may be various security performance measures associated with a configuration. In order to formalize, let us define a so-called M-value for each coalition in a configuration:

**Definition 2** Let M be an environment. An M-value is a pair  $(\mathcal{V}, v)$  where  $\mathcal{V} \subseteq \Gamma$  and v a value function over each member of  $\mathcal{V}$ .

Now, to identify the different value schemes that individual agents may apply to coalitions, we define an M-evaluation as a family of M-values where the index set, represented by the set of agents, of the family is used to identify the different schemes

**Definition 3** Let M be a coalition formation environment. An M-evaluation is a finite family  $V = \{(\mathcal{V}_\omega, v_\omega)\}$  of M-values for each agent a.

Analogously, we have to perform an M-Evaluation of the possible reconfiguration functions  $f: \Gamma \times t \rightarrow \Gamma$ , when security incidents, like alarms, occur at a certain point in time that require assistance and gathering of security guards active on other routes, or that require a coalition formation support system, like ASK-ASSIST, to be introduced or lined up.

This ranking feature of an  $M$ -Evaluation operator with respect to context-dependent reconfiguration functions provides the desired means to recommend the right coalition formation support for improvisation by humans. In this case the  $M$ -Evaluation may also cover the actual and plausible future operational contexts (including evaluations, e.g. involved costs) of more than one coalition. For example, whenever a security guard is instructed to leave his or her route to assist a colleague on another, the required time needed for providing assistance and the delay caused by traveling forth and back may hamper his or her overall route performance more than that of another guard just in close vicinity of the guard in need.  $M$ -Evaluation will shed a light on various types of differences as well as on communication delay reduction and other operational performance advantages of adopting ASK-ASSIST. This ranking can be formally captured by inducing context-dependent hierarchies of reconfigurations following similar ideas presented in the literature [19] for quality of context-based optimization of end-to-end mobile-health data delivery services:

**Definition 4** Let  $M$  be a coalition formation environment and  $V$  an  $M$ -evaluation in terms of  $M$ -value pairs  $(\mathcal{V}, \nu)$  where  $\mathcal{V} \subseteq \Gamma$  and generalized value function  $\nu: \mathcal{V} \times \dots \times \mathcal{V} \rightarrow \mathcal{R}$  describing the value for consecutive reconfigurations coinciding with (unexpected) context changes over time. Assuming that reconfigurations occur at  $t_0 \dots t_n$ , corresponding to time-points at which incidents occur, a CMH of reconfiguration function  $f$  is defined by:

$$CMH(f) = \{(\Gamma_q^{opt}(t_0^r), \dots, \Gamma_q^{opt}(t_n^r)) \mid q, n, r \in \mathbb{N}\}$$

where  $\Gamma_q^{opt}(t_i^r)$  is valid from  $t_i$  to  $t_{i+1}$  and where  $r$  is denotes the route number and  $q$  denotes the index set of the ordering on the set of possible (re)configurations. Note that the time points carry route-specific annotations. Labeling  $\Gamma$  by *opt* expresses the optimality of the (re)configuration in terms of the  $M$ -evaluation associated with it, such that the following holds:

$$(\Gamma_q^{opt}(t_0), \dots, \Gamma_q^{opt}(t_n)) = \text{arg}(\Gamma, \dots, \Gamma) / (\Gamma_{q-1}^{opt}(t_0), \dots, \Gamma_{q-1}^{opt}(t_n)) [\text{ext } \nu(\Gamma(t_0), \dots, \Gamma(t_n))]$$

where  $\text{ext } \nu(\Gamma^*)$  denotes an extreme value configurations, this could be either a maximum(for profits) or a minimum(for costs) depending on the type of value function used.

Let us give an example to illustrate a course of events where a simple reconfiguration is suggested to handle an incident:

Today security guard Bob is scheduled for route 326. Bob needs to visit four different sites that night. A glimpse on his schedule tells him that, first, he is to visit a local bank and, then, a hangar complex near the harbor. He leaves the base for a visit to the first security object planned on his route. At  $t_1$ , he arrives at the site. After the usual preparation for each visit (with the PDA), Bob discovers he does not possess the set of keys to enter the bank. Because of this incident, he decides to call the support system, ASK-ASSIST. This is at  $t_i$ .

To support the decision, ASK-ASSIST will suggest a possible reconfiguration (or several suitable reconfigurations) to those involved in the process. We can give a more formal representation of the original configuration (partially) given in the scene sketched so far as. The configuration consists of different coalitions  $\gamma^f$ .

ASK-ASSIST, by means of function  $f$ , may generate multiple possible reconfigurations of the current configuration. Considering the time-points  $t$  and, in specific, the time  $t_i$  at which the incident occurs, a reconfiguration by function  $f$  can be denoted as follows:

$$f(\Gamma, t) = \begin{cases} \Gamma(t) = \Gamma_s(t), & t_0 \leq t < t^i \\ \Gamma(t) = \Gamma_s^i(t), & t^i \leq t < t_n \end{cases}$$

with

$$\begin{aligned} \Gamma_s &= \cup_r \gamma^r \\ \Gamma_s^i &= \cup_r \bar{\gamma}_{t_n}^{A_n, B_n} \circ \dots \circ \bar{\gamma}_{t_{i+1}}^{A_{i+1}, B_{i+1}} \circ \bar{\gamma}_s^i \circ \gamma_{t_i}^{A_i, B_i} \circ \dots \circ \gamma_{t_0}^{A_0, B_0} \Big|_r \end{aligned}$$

From the time point on at which the incident occurs, the configuration is altered in order to deal with the unexpected event. For example, given Bob's problem, ASK-ASSIST might suggest that Carol or Bob's team leader Alice, who are actually both in possession of a set of the right keys, should be aligned. Carol is in the vicinity. So, her travel-time is less than that of Alice.

At  $t_2$ , Bob contacts Carol through the system. She agrees to come by, which, she estimates, will take about 10 minutes. At  $t_3$ , Carol arrives at the local bank and hands over the keys to Bob. She leaves, at  $t_4$  to continue her daily routine as originally planned. Bob, too, continues his schedule with a slight, though acceptable, delay.

Other possible configurations would have been possible. For example, the key could have been brought to Bob by the team leader. Also Bob could have driven to one of the others as well. ASK-ASSIST can decide who to ask upon the time-delay order of all possible reconfigurations. This order is given by the context-dependent  $M$ -evaluation hierarchy on  $f$ . The resulting ranking can be taken into account by the user.

Herewith our conceptual model of our mobile surveillance domain is complete but not yet readily made operational. Grounding or empirical modeling of context-dependent CMH-hierarchies of reconfiguration function  $f$  and ASK-ASSIST functionalities are needed in order to really support improvisation by humans in setting up communications or taking decisions. Methods are needed that are capable to propagate information or knowledge about surveillance operations throughout the evolving security network of multi-agent systems. For example, when

introducing ASK-ASSIST for aligning systems and security personnel in case of an incident, we would like to take advantage of the already stored and accessible historic contexts to do so. Thereto, the network should self-organize this given the potential reduction of communication chains and delays provided by ASK-ASSIST compared to standard way of working, and the observed and processed historic contexts of e.g. who wants to contact whom when such an incident happens. This self-organization of the evolving security network culminates in a prediction model for reconfiguration given unexpected events or the introduction of novel coalition formation support systems.

### 3.3 Robust Reconfigurations

Collaborative content-based filtering [2] can readily be applied to incident-specific coalition formation problems in the security domain. Joint kernels weighing past reconfigurations (explicitly by measuring feedback or implicitly by measuring occurrence) can be used to predict and recommend ranked lists of reconfigurations, i.e. context-dependent *CMH*-hierarchies on the reconfiguration function, given not anticipated incidents or the introduction of ASK-ASSIST. Besides collaborative filtering also machine learning of e.g. Bayesian network classifiers can be applied in order to obtain such hierarchies [11]. Although these approaches provide means to ground those hierarchies in actual and historic contexts, they do not generate robust coalition formation solutions, whenever those contexts or *M*-evaluations are uncertain. The context values, e.g., locations of security guards and road traffic states, can only be known up to a certain level of resolution. Furthermore, specific types of *M*-evaluations, such as surveillance expertise needed in case of alarms, require exploring and analysis of the exhibited skills and knowledge level of security personnel at typical context-dependent critical temporal scales. Thus to find a lost key it makes sense to ask the most experienced only when there is evidence that the last guard who visited the security object claims he has deposited the key where it should.

In order to arrive at robust grounded coalition formation alternatives we elaborate on how to apply our dynamic scale-space paradigm [14]. Thereto, it is necessary first to define notions of reconfiguration image, gauge and dynamic scale-space of reconfiguration image.

**Definition 5 (Reconfiguration Image)** *A reconfiguration image  $I: M \rightarrow N$  is a representation of the external vector-valued energy-density field  $M$  of an actual reconfiguration as a vector-valued density field of the induced one  $N$ .*

Note that the vector-valued density field of the induced reconfiguration is a member of all potential and existing context-dependent reconfigurations that can be made operational or are stored in a database, e.g. ASK-ASSIST solutions for putting the right security people in a mobile conference call when an alarm occurs.

To properly analyze such an image in terms of a complete and irreducible set of equivalences, it is mandatory to know how the

reconfiguration image changes whenever they are subjected to a particular class of so-called gauge groups:

**Definition 6 (Gauge)** *A gauge group consistent with reconfiguration image (Definition 5) is a group or set of transformations leaving (Definition 5) or some of its properties invariant.*

Such gauge groups could cover spatio-temporal deformations and even morphological transformations including spatio-temporal reordering, cutting, pasting, insertion and deletion of reconfiguration (image) objects. For example, introducing ASK-ASSIST in a security company next to the standard way of working causes non-trivial image reconfigurations in case of unexpected alarms and given incentive to reduce the workload or to increase operational performance.

A set of equivalences  $F$  of a reconfiguration image (Definition 5) comes about after setting up a (co)-frame field, metric and/or connection invariant under a gauge group (Definition 6), see [14]. As alluded some of the gauge groups can generate active transformations, such as introducing ASK-ASSIST. However, these transformations can be undone by means of similarity operations inducing robust deliberation schemes on the reconfiguration image. Thus a categorization problem with respect to reconfiguration images emerges that involves besides the problem of invariance under the gauge group also the problem of demonstrating its robustness under similarity operations. The latter implies that a reconfiguration image acquired upon multi-scale collaborative filtering - required in case of a sudden alarm - need not cause a dramatic drop in the security level compared to the daily patrol shifts executed.

In order to ensure robust reconfiguration a gauge group consistent dynamic scale-space of reconfiguration images must be generated [14]. Such a dynamic scale-space is obtained by a gauge group consistent context dependent coarse-graining of reconfiguration images. Such a coarse-graining or self-similarity operation removes microscopic aspects of the images and yields similar reconfiguration image properties above critical context-dependent length scales.

**Definition 7 (Dynamic Scale-Space)** *A dynamic scale space of the context-dependent free energy  $F$  of a reconfiguration images, that is invariant under gauge group  $G$ , is governed by*

$$\delta_\tau F = -j^F,$$

where

$$j^F = -\frac{\nabla_{v_s}^\Pi F}{\kappa^2(\sqrt{g(\nabla_{v_s}^\Pi F, \nabla_{v_s}^\Pi F)}}$$

with

$$Z = \exp[-F[V_i(x)]]$$

and

$$F[V_i(x)] = \sum_{i,k,p} dv^p \left( \tilde{V}_{i;\pi_\kappa(g_1 \dots g_k)}(x, \tau_{i;\pi_\kappa(g_1 \dots g_k)}) \right)$$

with  $\kappa$  a monotonic increasing function,  $(g, \Gamma)$  a metric and connection, suitable initial-boundary conditions,  $v_s$  connecting free equivalence states,  $F(p_i)$  and  $F(p_j)$  and free energy  $F$  is related to statistical partition function  $Z$ . Here  $x$  labels any context complex and  $\tau$ 's label dynamic scales consistent with the gauge group  $G$  and equivalences.

Note that the evolution of contexts, agents and associated joint tasks, and, thus, reconfiguration images are captured by the dynamic scale-space. Robust prediction models for recommendations of context-dependent reconfigurations naturally emerge over time: evolutionary symmetry breaking is covered by the dynamic scale-space of the reconfiguration images, as the renormalization-group coarse-graining is intrinsically coupled to them; a necessary prerequisite for self-organization.

This does not mean that the introduction of ASK-ASSIST may have such an impact. It merely causes a renormalization of the performance of the security organization even if security objects and procedures become more complex: normally humans would solve incidents for example in an awkward way with many people involved. Adding ASK-ASSIST removes unnecessary overhead from the required new configurations, but does not reduce the level of complexity of the security problem introduced by an incident - law of conservation of complexity or rather free energy  $F$  prevails.

In the mobile surveillance security domain, historic data is available in the database that stores all operational data as past security operations in the case of incidents (e.g., David calls Bob when an alarm occurs). Knowledge on past operations is propagated to initialize the parameters in the system. Bootstrapping ASK-ASSIST in case of such incidents, allows proper analysis on how operational performances may increase. In case of, for example, the occurrence of an alarm, the probability of the old working procedures is also spread over ASK-ASSIST; this boils down to that next the standard way also the use of ASK-ASSIST is recommended!

Critical solutions manifest themselves on context-specific temporal scales, for example, finding a suitable guard with expertise about a security object. This can be determined taking into account the right time scale. The time scales of determining the most *recent* possessor of a particular set of keys and the time scales of determining some guard's expertise (*longer period*) are different. In fact, recommendation in the case of incidents entails scrutinizing contextual segments of a security guard that has found itself in a related context. To determine the different types of time scales, greedy methods could e.g. be iteratively applied to the run-length encoding of the primal context data stored and acquired till corresponding query-specific thresholds are surpassed. These thresholds are manifest in the time-periods that appeared statistically to be predominant and lead to incident resolution. Note that for finding keys or the person that latest

visited a security object corresponding to those keys might be the same and implies to use another contextual scale instead of the temporal one.

#### 4. DESIGN IMPROVISATION SUPPORT

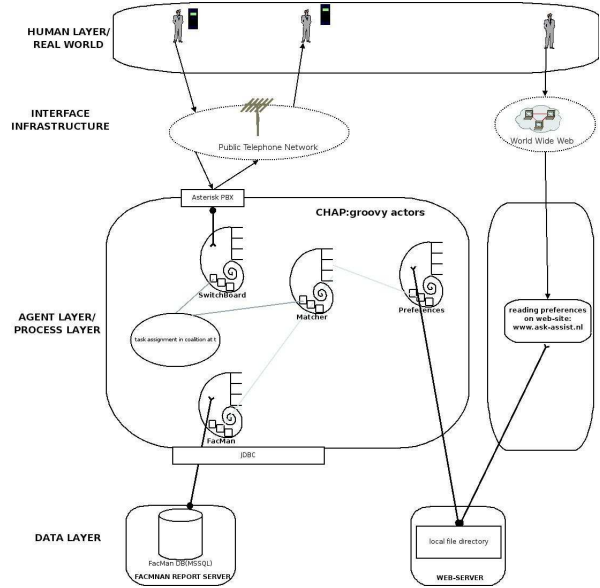


Figure 1: ASK-ASSIST architecture.

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. In this section, we give a description of the ASK-ASSIST system architecture. ASK-ASSIST has been developed by Almende B.V. on its Common Hybrid Agent Platform (CHAP).<sup>1</sup> This platform enables hybridization of ICT systems, organizations and agent systems, allowing the run-time addition and reorganization of networks without undue impairment of quality of service.

The design of the ASK-ASSIST system for improvisation support is based on a three-layered architecture (see Figure 1). We focus on the layer of software agents that intermediates between human actors and the data/information layer. While implementing in *groovy actors* there are a number of small adaptations that allow the introduction of an agent-based layer. Most notably, the introduction of an agent(actor) that is dedicated to processing information from a remote database containing operational data of the mobile surveillance process. This database is known as the FacMan Database (FMDB). In general, for every information source, a dedicated agent exists.

<sup>1</sup> <http://sourceforge.chap.net>



## 4.1 FacMan Agent

The FacMan Agent is the agent that has access to the remote FacMan Report Database. In this database a wide variety of operational data on the security surveillance process is stored:

- data on personnel
- data on routes (route numbers, scheduled objects to visit, tasks to perform, ..)
- data on tasks (operational times)
- data on alarm handling
- data on keys
- data on phones and PDA devices
- data on objects
- data on phone calls from and to dispatch centre
- detailed log on 19 different log levels

## 4.2 Matcher Agent

In the mobile surveillance security domain, historic data is available in the database that stores all operational data as past security operations in the case of incidents such as alarms. Knowledge on past operations is propagated to initialize the parameters in the system. Bootstrapping ASK-ASSIST in case of such incidents, allows proper analysis on how operational performances may increase. The matcher agent is responsible for processing raw data into context-sensitive coalition formation schemes. These schemes can be used for recommendation to the human actors, if an incident is spotted.

To ensure the system robustness, formation schemes are not merely used in offering recommendation on to other agents, but the actual schemes are distributed to all agents that are hosted on communication devices such as PDA's. In this way, by means of redundancy, each individual can use basic recommendation even if the main matching agent is not available.

## 4.3 Switchboard Agent

When a call comes in from the public telephone network, the switchboard agent (actor) creates a telephone agent that identified by the caller-id of the incoming call. The switchboard is based on Asterisk functionality. The switchboard agent is also used to set up calls.

## 4.4 Preferences Agent

The preference agent is able to retrieve the personal preference on, for example, speed dial configurations that are filled in by the security personnel on the ASK-ASSIST web-site.

## 4.5 Entity Agents

In the mobile surveillance domain, several entities such as guards, phones and security sites are connected in a network that is subject to continuous change. ASK-ASSIST manages this evolution by representing entities as agents that are interconnected as a heterogeneous network (Dorogovtsev and Mendes, 2003). The realizations of such a network at specific time points correspond with the coalition formation processes.

- **Telephone agents** typically implement the dialling plan based on the interactive voice response menu design of ASK.

The three main functions in the IVR flow consist of: logging in, setting up outgoing call and logging out. In illustration 5, we can see that once the telephone agent has been created, it can communicate with several other agents to retrieve and store (operational) information.

- **PDA agents** representing PDA devices
- **Guard agents** representing guards.
- **Site agents** representing security sites.
- **Task agents** representing tasks.
- **Key agents** representing keys.
- Etc...

## 4.6 Matchingmaking functionality

While on duty, an MS often requires information about a specific key to gain access to an object, or information about a specific object or alarm system. Using its matchmaking capabilities, ASK-ASSIST can easily find the right person (availability, lowest possible escalation level) to provide the requested information. The following options are implemented in the IVR menu flow.

### *Finding a key of a security object:*

An MS may arrive at an object and discover the proper key for accessing the object is not in the key-safe in the vehicle. Using this option, the MS is automatically connected to a person that is most likely to have information about the whereabouts of the key of the object at which the MS is present.

### *Finding MS having last visited this security object:*

If an MS arrives at an object and needs to contact a colleague that has last or very recently visited this security object, then the MS can use Ask-Assist to retrieve that person. Choosing this option, Ask-Assist can setup communication very easily selecting an available person/route.

### *Finding MS having experience with this security object:*

Sometimes an MS wants to contact a colleague having some significant experience with some object. It is interesting to look, not only at the object history, but also at the particular task that were performed by the candidate colleagues. For this option, we need to calibrate the time-scales, we query on by analyzing (overall) performance of the organizational reconfigurations. This means we use context-specific incidents to retrieve, list and rank possible candidates on the basis of the current incident.

### *Dynamic Conference call for alarm assignment:*

ASK-ASSIST should allow conference calls with TL dispatch employee and MS. This means that whenever a TL or dispatch employee is talking to an MS it must be possible to add another TL/dispatch employee into a telephone conference. One can imagine a key that allows the owner of the conference call to add the best suitable available candidate for an alarm assignment. Also here we can calibrate the time-scales we query on by analyzing (overall) performance of the organizational reconfigurations. This means we use an analysis of context-specific incidents to retrieve, list and rank possible candidates on the basis of the current incident.

## 5. EXPERIMENTAL RESULTS

An important problem in mobile surveillance is the assignment of route to alarms (see scenario 2). 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 historic 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 45.6%. About 8.7 % of the alarms are assigned after one hour (Figure 2)!

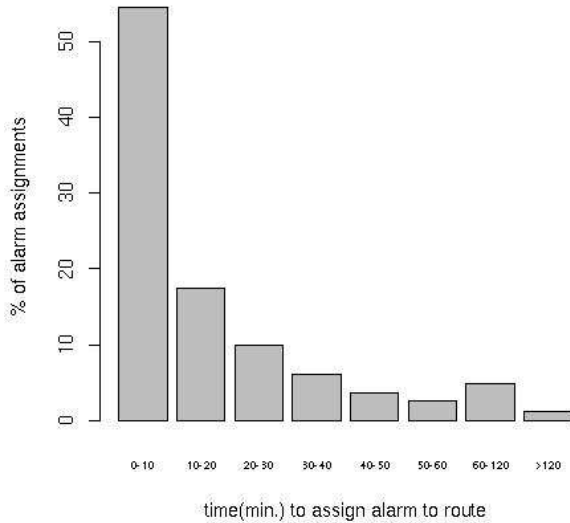


Figure 2: Alarm assignment rate versus time.

To speed up this process, we developed a dynamic conference call that allows a team-leader to set up a multi-party phone call. In Figure 3, we present the result of our recommender component. We varied the amount of months used for training, on the x-axis and looked at the resulting accuracy of our recommender in terms of 3 alternatives against the real decision made by the team-leader. We repeated these experiments for three different size of context taken into account. For example, when trained on a period of 9 months with a context of only the route, in 81.7 % of the alarms occurring in the test set, the real decision of the team-leader was among the best three recommendation made by the system. If we increase the number of context elements that is included in the recommendation process, the accuracy with respect to the real decisions made by the team leader decreases.

The training set concerns real data from a database storing operational data for the mobile surveillance process. The total amount is about 1 gigabyte of raw data. The test set concerned the alarms occurring during one month (1187 alarms).

Normally, in 59.45% of the alarm occurrences, the guard arrives in time (i.e., within 30 minutes). It is interesting to see that given a training period of 6 months or more and given a context that

includes the travel time from a specific object to the object where the alarm occurs, the amount of top three recommendations that includes solutions in which the guard arrives within 30 minutes is higher than 59.45%. That is, 68.2% for 6 months as training period and 70.78% for 9 months as training period taken from the database.

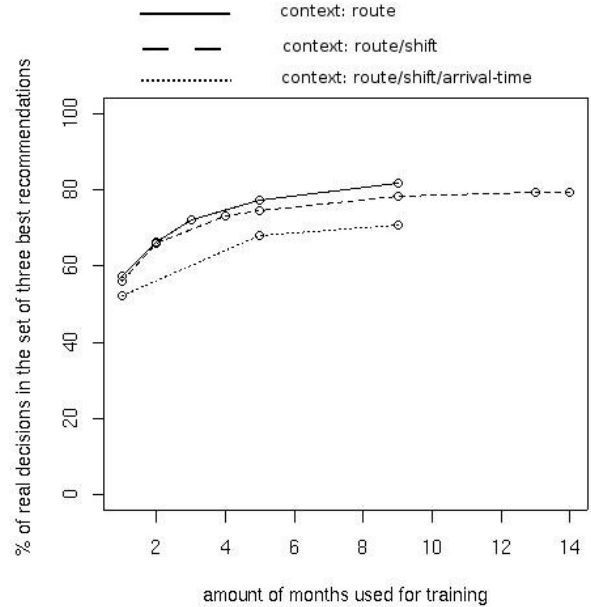


Figure 3: Recommender performance for different training periods and taking different contexts into account

## 6. Conclusion

We applied a dynamic coalition formation framework to the mobile security domain that enables communication and decision support to security personnel when faced with unexpected incidents. Our analysis of real data of the domain revealed that assigning alarms to guards on routes can be a time-consuming activity. Decision making by a team-leader without support for improvisation is a cumbersome task, because an exploration of alternative assignments tends to increase the assignment time even beyond the average of 19 minutes.

Through experiments in which learned the probabilities on a team-leader successfully assigning an alarm at a specific security object in a particular context to available routes, we showed that we can drastically the assignment time with a higher probability of success by providing alternatives.

We have performed these experiments for different training sets of different sizes and taking in account different number of context elements. The performance of the improvisation support saturates at particular time-scales and particular number of context elements.

## 7. REFERENCES

- [1] Abbink, H., Dijk, R. V., Dobos, T., Hoogendoorn, M., Jonker, C., Konur, S., Maanen, P. V., Popova, V., Sharpanskykh, A., Tooren, P. V., Treur, J., Valk, J., Xu, L., and Yolum, P. (2004), Automated Support for Adaptive Incident Management, *Proceedings of the first International Workshop on Information Systems for Crisis Response and Management (ISCRAM '04)*, Brussels, Belgium, pp 69–74.
- [2] Basilico, J. and Hofmann, T. (2004) Unifying collaborative and content-based filtering, *Proceedings Twenty-first International Conference on Machine Learning (ICML'04)*.
- [3] , 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.
- [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] Buşoniu, L., De Schutter, B., and Babuška, R. (2005), Learning and coordination in dynamic multiagent systems, Technical Report 05-019, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.
- [6] Burghardt, P. (2004), Combined Systems: The combined systems point of view, *Proceedings of the 1st International Workshop on Information Systems for Crisis Response and Management (ISCRAM '04)*.
- [7] Gale, D. and Shapley, L. (1962), College Admissions and the stability of marriage, *American Mathematical Monthly* 69(1), 9.
- [8] Johnson, C. W. (2003) Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting, University of Glasgow Press, Glasgow, Scotland.
- [9] Knuth, D. (1997), Stable Marriage and its Relation to Other Combinatorial Problems, *CRM Proceedings and Lecture Notes*, Vol. 10, American Mathematical Society.
- [10] Mendonça, D. and Fiedrich, F. (2004), Design for improvisation in computer-based emergency response systems. In *Proceeding of the first international workshop on information systems for crisis response and management*.
- [11] Nurmi, P., Salden, A., Lau, S. L., Suomela, J., Sutterer, M., Millerat, J., Martin, M., Lagerspetz, E., and Poortinga, R. (2006), A system for context-dependent user modeling, in Proceedings OTM Federated Workshops, *Lecture Notes in Computer Science*, Vol. 4278, pp 1894–1903, Springer-Verlag.
- [12] 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.
- [13] Roth, A. and Sotomayor, M. (1990), Two-sided matching: a study in game-theoretic modeling and analysis, *No. 18 in Econometric Society Monographs*, Cambridge University Press, Cambridge.
- [14] Salden, A. H., Ter Haar Romeny, B. M., and Viergever, M. A. (2001), A Dynamic Scale Space Paradigm, *Journal of Mathematical Imaging and Vision* 15(3), 127-168.
- [15] , 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.
- [16] Storms, P. P. A. (2004), COMBINED systems: a system of systems architecture. In *Proceeding of the first international workshop on information systems for crisis response and management*.
- [17] Valk, J. and Ferro, D. (2006), A generalized matching framework, combining matchmaking and coalition formation, in *Proceedings of the 18th Belgian-Netherlands Conference on Artificial Intelligence (BNAIC '06)*, pp 315–322, Koninklijke Vlaamse Academie van België voor Wetenschappen en Kunsten, Brussels, Belgium.
- [18] 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.
- [19] Widya, I. A., van Beijnum, B. J. F., and Salden, A. (2006), QoC-based Optimization of End-to-End M-Health Data Delivery Services, in *14th IEEE International Workshop on Quality of Service (IWQoS'06)*, New Haven (CT), USA, pp 252–260, IEEE Computer Society Press.