Police Interventions as a Context-aware System. A Case of a Contextual Data Modelling

Radosław Klimek
AGH University of Science and Technology
Krakow, Poland *rklimek@agh.edu.pl*

Abstract

Smart systems which operate in Intelligent Environments (IE) are complex. They analyse the large volumes of various contextual data on-line and often in real time to obtain, autonomously and reliably, the required pro-activeness of a system which operates pervasively. We proposed both a development framework for context-aware systems and a context-based decision making scheme for the system of managing police interventions, focusing on providing support for police patrols in life threatening situations. This system, owing to the symultaneous collection of rich contextual information from many police officers, which constitute the mobile network, as well as the complex processes of contextual reasoning, takes automatic decisions on supporting officers in emergency. We implemented the initial, yet not trivial, simulations of the system behaviour within the whole city. The results obtained prove the feasibility of the framework.

Keywords: Pervasive Sensing, Gathering and Modelling Contextual Data, Decision Making, Police Interventions

1. Introduction

Smart systems, especially in connection with IoT (Internet of Things), AmI (Ambient Intelligence) or AI (Artificial Intelligence), and operating in IE (Intelligent Environments), may be a source of misunderstandings both on the part of the designers (difficulties) and the users (threats) of such systems. The difficulties and threats mentioned may be overcome by introducing appropriate methodologies and frameworks that will allow for better development of context-aware systems and consequently making decisions in the complex multi-agent environment.

The most general idea of the system supporting police interventions proposed herein, colloquially referred to as a *smart gun*, consists in the system searching autonomously for support among police patrols located in the surroundings in the event of sudden and unplanned firing by a police officer during a regular intervention. Thus, a police officer is released from activities connected with communication and such a police officer can be focused on the firing situation and he/she can be certain that the support of colleagues will be soon provided automatically.

Our objective and contribution is to introduce an appropriate framework for the modelling and developments of context-aware and pervasively sensing systems which provide smart decisions as a result of the pervasively acquired contextual data, its online filtration and processing, including weighted MaxSAT solving. We will argue in favour of our approach showing the practical implementation of a non-trivial system supporting police services. The architecture of multi-agent system oriented at contextual data processing was proposed, where the agentification was carried out together with its behavioural model. We conducted the first few simulation experiments; these are the experiments on the scale of the entire city. The system increases the self-awareness of the environment and the processes of decision-making automation in IE.

2. Related Works

The concept of a context and contextual data is known and it was mentioned for the first time by Dey and Abowd [6]. Whereas, Zimmermann et al. [16] introduce the categories of a context, which is to cope with the complexity of contextual data and to facilitate understanding. We introduced a similar categorisation for contextual data but matching the reality of the system aiding police interventions, cf. [16, 8]. The article by Cheng et al. [4] also analyses the endeavours of research communities in understanding the context. Whereas, the article by Bettini et al. [2] discusses context modelling techniques, introduces abstraction levels and other requirements. Perera et al. [14] survey context awareness from an IoT perspective. It addresses a great variety of techniques, methods and solutions related to context awareness and IoT. The article helped us to understand the background when introducing IoT for context-aware basics.

Biegel and Cahill [3] proposed the development framework to gather data from separate sensors, to represent a context and to reason about a context. It was designed for ad-hoc wireless environments. Our approach takes the relationship between contextual data and its operational aspects into account. Ferreira et al. [7] presented a framework to gather reasons concerning a context on mobile devices. By means of the encapsulation of implementation sensor data details, it is exposed to the sensed context enabling abstractions. The approach results from the mobile data specificity and can be used partially in our approach. Bardram [1] proposed a Java-based framework for creating context-aware applications. It supports runtime architecture and its programming model. From our point of view, this approach is oriented excessively towards the implementation phase, whereas our approach is focused on the design stage. Perhaps in the future, it will be advisable to combine both approaches. The smart gun idea is not completely new [15, 9] but it is limited only to the identification of an authorised user. Such an approach is related to pattern recognition when biometric data is used. In article [9] the idea of a smart gun is discussed but it only presents the concept of a smart gun system on the example of use cases and scenarios expressed in a natural language. In this paper, a multi-agent system was proposed and a detailed analysis of contextual data was conducted. A prototype of a simulation environment was implemented.

3. Context Understanding and Modelling

According to a classic definition [5], a *context* is a collection of circumstances and facts surrounding a given object. The introduction of subsequent notions will be illustrated with references to our system through the following phases: category identification (CI) $\rightarrow$ operational relationships (OR) $\rightarrow$ attributing $\rightarrow$ assigning.

3.1. Categorisation

Category identification (CI) means the conceptual data division into respective categories, see Figure 1, in order to obtain the subsets of elements which are semantically disjunctive:

- *Individuality* – the circumstances in which an object is present, something in which it is involved, here: being on/off duty, also a patrol which is incomplete as a result of firing;
- *Time* – the circumstances related to the current and passing time, here: the time of an incident, i.e. a day or a night, current time, the prolonging duration of an incident;
- *Location* – the position of an object, its geolocation, as well as spatial relation, here: geolocation, also a bad/good (not safe/safe) district, i.e. with a higher/lower risk;
- *Activity* – information on activities which an object is involved in, here: the patrol statuses: observation, going for an intervention, intervention, going to provide support, firing;

- *Relations* – a relation towards other objects in the environment under consideration, here: searching radii both for intervention and support, different times of a day/night, etc.

Identified contextual data resulting from the categories proposed, see Figure 1, affect considerably the decision-making processes of MC agent, see Section 4.

3.2. Operational Relationships

Operational relationships OR, see also Figure 2, allow to plan activities on contextual data in the system under design using the following *R6 rule*:

- *Represent* – leaving the data unchanged, leaving the data as it is after its readout, for instance: current time readout, patrol geolocation;
- *Resolve* – data conversion, filtering or aggregating, to forms which are proper and readable in terms of the system needs;
- *Retain* – preserving unchanged data in a form obtained in a previous phase, data collection phase, here: e.g. current system time;
- *Reinforce* – data collection, perhaps merging and thus obtaining a new emphasising perspective, here: time/time stamp with patrol geolocation;
- *Remove* – data removal and not storing it, such data is not used until new values are read out, here: e.g. a current patrol distance from a location where it is to provide support;
- *Remain* – leaving data in the system until it is overwritten by the readout in a new data acquisition cycle, here: e.g. geolocation intervention locations.

Figure 3 presents the interrelations between the respective sets of contextual data. Furthermore, this allows to plan and track the processes of contextual data conversion. We assume that given subsets are pairwise non-intersecting.

3.3. Attributing

The next step is to attribute the information obtained concerning contextual data and its hierarchy. *Attributing* involves, firstly, assigning variables to individual leaves of the tree shown in Figure 4. The next step is *assigning* variables to the agents of the adopted MAS (the MAS architecture is proposed in Section 4). It means that individual agents operate on these variables.

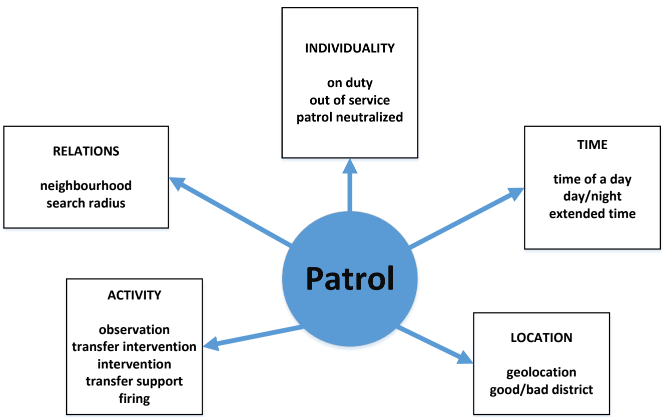

Fig. 1. The categorisation of the entire system contextual data in relation to a single police patrol

Both attributing and assigning allow us to build a rich picture of the MAS system's operations, in particular, taking contextual data into account.

To sum up the entire section: it allows for better understanding and management of contextual data, with consideration of said agent model. First, we identify the categories of contextual data, then we plan specific variables for each of them, however, multiple variables may be associated with each category. We evaluate how the OR (R6) model affects these variables (attributing), i.e. how particular variables will be processed in the system. In the end, we identify which variables will be required by particular agents (assigning). It gives us a good picture of contextual data processing in the entire context-aware system.

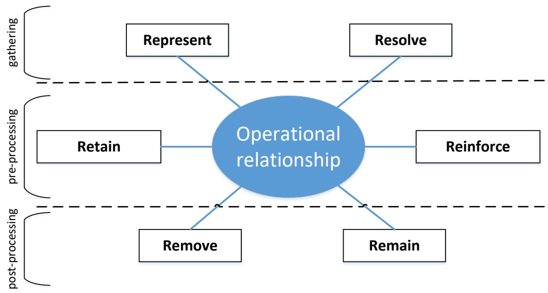

Fig. 2. Operational relationship as an operational model of contextual data conversion

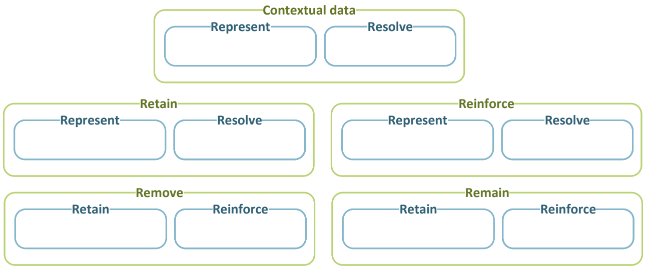

Fig. 3. The interrelation between individual sets and subsets of contextual data, or a schema for a top-down selecting of successive disjoint subsets, see also Fig. 4

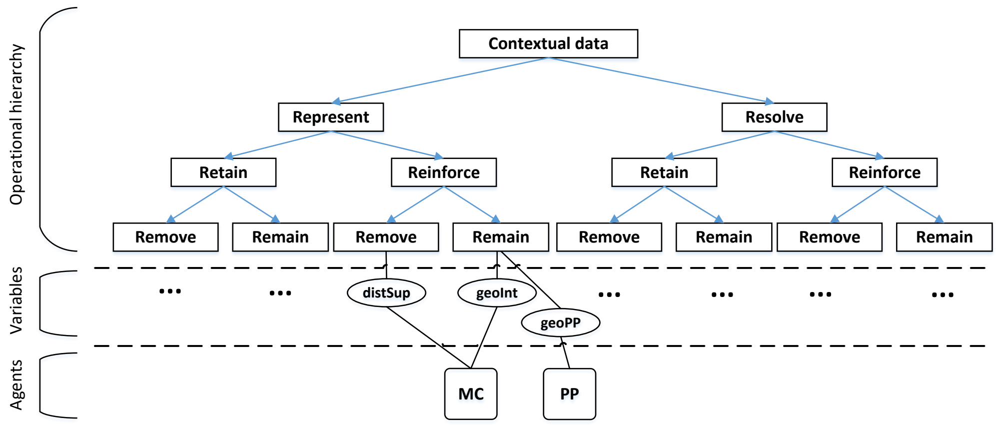

Fig. 4. Attributing contextual data and its hierarchy basing on the previous relations, see Fig. 3 and Fig. 4, and assigning individual variables to leaves. (For example, variables $geoInt \equiv$ geolocation of the intervention location; $distSup \equiv$ distance from the support location of a current patrol; $geoPP \equiv$ geolocation of the police patrol.) Then, assigning variables to agents. (The MAS architecture is shown in Fig. 5)

4. Multi-agent System

We propose the following division of agents constituting the entire multi-agent system (MAS) and its architecture. *Agentification* means the creation of specialised software objects which combines autonomy and specialisation of tasks. It allows us to disperse well and balance the system activities. This agentification issue is close to the IoT (Internet of Things), which is discussed in some articles, see the articles by Maamar et al. [13], and by Kwan et al. [11]. Our approach, when considering an urban ecosystem, covers partially the proposed methodology for the agentification of things. Figure 5 presents the architecture of the proposed agent system.

MC – management centre, an agent initiating the system and other agents, and subsequently implementing the ongoing management, appoints and dismisses PP agents, it collects data on interventions from all PP, it stores and models incoming contextual data, it submits the demands for dispatching ambulances to the firing site to AB,

PP – police patrol, or police officer's smartwatch, it establishes its geolocation and sends it cyclically, at the beginning of being on duty, it creates Gn and Nv agents, it receives signals from MC with the order for regular intervention.

Gn – police officer's gun, it sends signals to PP, informing on the fact of firing, which is its fundamental role,

Nv – navigation in the police officer's car, it receives a new, indicated geolocation from PP with the order for directing the patrol to the spot,

X – it collects messages on the monitored area incoming from all PP and it hands them over to MC, the only role of the agent is to mediate in message providing, In the future, the entire system may be optimised in terms of increasing its capacity through many X agents,

HQ – police headquarters, command system, is capable of affecting the current situation by changing the values of given parameters affecting MC decisions, e.g. increasing or decreasing the number of patrols when providing support, changing the number of ambulances reaching the place, etc.

AB – the centre of management and ambulances fulfilling the demand of MC.

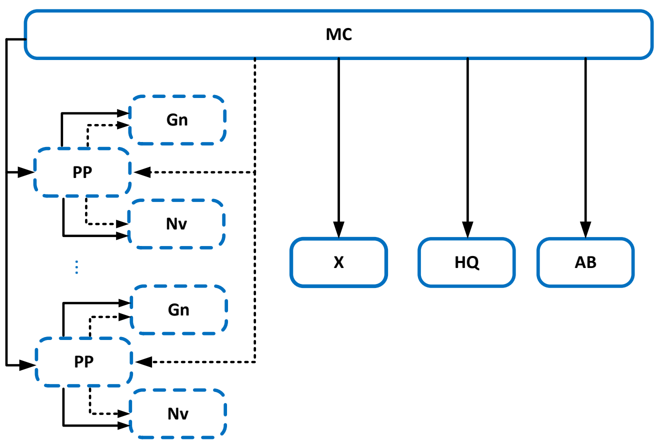

Fig. 5. Basic agent architecture, in the example of only two PP agents, and agent relationships (solid ovals show permanent agents and dashed ovals show agents which may exist temporarily. Solid and dashed lines show agents' constructions and destructions, respectively)

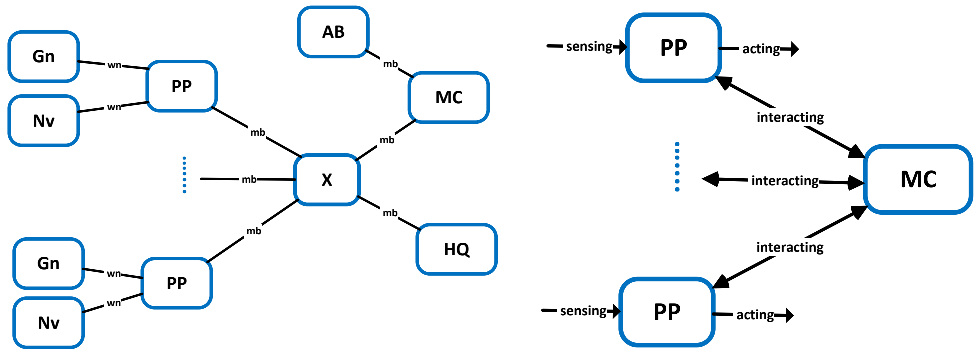

Fig. 6. The agent environment, left: topology and types of communications ("wn" means a local wireless network and "mb" means use of a message broker), right: model of agents' interactions

The architecture presented may be developed in the future, for instance MC may be extended and decomposed by the separation of tasks. As already mentioned, Figure 4 illustrates the assignment of context variables to individual agents.

Figure 6 shows the agent environments. Communication takes place either via the wireless local network (short-range radio, Bluetooth, etc.) or with the use of a message broker (e.g. Kafka, Rabbit, etc.). The logical model includes both the context sensing and action, but also the agents' communication.

We applied one simplification, although not a significant one. Usually, a patrol consists of two police officers – however, at this moment, we assume, for one patrol, the existence of one PP agent, patrol commander, one Nv agent, patrol car equipment, and Gn agent. In fact, police officers may have more than one gun.

5. Behavioural Model

5.1. Police Patrol

Let us present the behavioural model of the system based on the state diagrams in respect of police patrols, see Figure 7. Each patrol can have one of the following states:

- **Observation** or **patrolling** – A patrol moving around a neighbourhood, usually moving

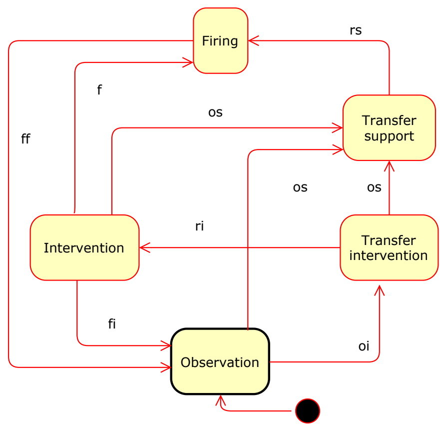

Fig. 7. A state machine for a single police patrol PP, basic states only. (Observation is both initial and final state. Signals: oi – order intervention, os – order support, ri – reach intervention, rs – reach support, f – on fire, fi – finish intervention, ff – finish firing)

slowly, stopping occasionally, observing the area patrolled until it is called for intervention or a firing incident;

- **Transfer intervention** – A patrol during going to an (regular) intervention – upon the MC call, a patrol goes to an intervention to an indicated location; such a drive may be interrupted only by a call for support, if this takes place, there is switching to the support state, and the intended intervention still remains in the MC in the pool of interventions requiring solving;

- **Intervention** – A patrol during an (regular) intervention – it may be interrupted only by a call for support in a firing incident, and the intended intervention goes back to the MC to the pool of interventions requiring solving;

- **Transfer support** – A patrol going to support – upon the MC call, a patrol goes to support to the indicated geolocation; such a patrol drive may be interrupted only by calling off, in the meantime, the assumed number of patrols in the support, if such calling off take place, observing is started, but if a patrol going to support reaches its intended geolocation, it switches automatically to the during firing state;

- **Firing** – Patrol during firing or chasing – only during a regular intervention, it may change into a firing incident; a trigger here is firing a shot by one of the police officers; a patrol intervening originally becomes a firing host then;

A proper city map of the system operation procedure, with the patrol positions marked, is available at the headquarters from the HQ agent; whereas, Figure 7 presents a state machine for each patrol on a patrol duty in the monitored area.

The other rules binding in the system are as follows:

- only one patrol indicated by the MC may go to an intervention;

- support in a firing incident may be provided by a certain number of patrols; nonetheless, during a drive, a specific excess number of patrols may be called for support, and this redundancy is an effect of various calculations, including the distance of the patrols called from a target place, anticipated increased traffic hindering the drive, etc. – it is better to call more patrols, and the prospective excess may be called off when the intended number is reached on the spot. Our context consideration extends [10] and Algorithm 1 shows how to get the list of supporting agents. Regarding line 1: P_t is a set of negotiating agents, here $P_t = \{MC, PP_1, PP_2, \ldots\}$, Y_t is a set of *public* pieces of context data (variables), e.g. $Y_t = \{districtSafety, timeOfDay, dayOfWeek\}$, and X_t is a set of *private* pieces of context data (variables), e.g. $X_t = \{geoPP, currentState, serviceTime\}$. Lines 2–6 show public variables processing to determine the size of support required ($sizeSup$). Lines 7–18 show private variables processing to designate a specific set of agents ($supList$). Each variable in line 4 is mapped to corresponding numeric values, which in turn is set by HQ before or while the system is running. For example, for variable $districtSafety$, the values of $(NotSafe, RatherSafe, Safe)$ are mapped to numbers (2.5,1.5,1.0), respectively. Other public variables are processed in a similar way. Once the final value of $sizeSup$ is determined, see line 6, through the public contextual data analysis, it remains to designate the specific PP agents. This is solved as a weighted MaxSAT problem, see lines 7–18, where private pieces of contextual data are weights in the task. For example, a short distance to the shooting site (i.e. small difference between $geoInt$ and $geoPP$) has a high weight, $currentState = Observation$ is high, with $currentState = TransferIntervention$ rather low and $currentState = Intervention$ very low. Because the size of the designated optimal set of PP may be

Algorithm 1 Negotiating fire support

Input: $geoInt$ ▷ geolocation of the intervention location, see also Fig. 4
Output: $supList$ ▷ list of agents to provide support
1: $Cx_t = P_t \cup X_t \cup Y_t$; ▷ establish context Cx for the time t
2: $sizeSup$:= 1; ▷ public variables processed in lines 2–6
3: **for** every $var \in Y_t$ **do** ▷ the size of support required
4: $sizeSup$:= $sizeSup \cdot (var.map)$ ▷ each variable is mapped
5: **end for**
6: $sizeSup$:= $round(rdF \cdot C \cdot sizeSup)$; ▷ redundancy factor rdF, const C set by HQ
7: $formula$:= $\emptyset$; ▷ private variables processed in lines 7–18
8: **for** every agent $\in P_t$ **do**
9: **for** every $var \in X_t$ **do**
10: Add var to the weighted MaxSAT problem in $formula$
11: **end for**
12: **end for**
13: $supList$:= $weightedMaxSAT(formula)$;
14: **if** $supList.len > sizeSup$ **then** ▷ to obtain $supList.len = sizeSup$ in both cases
15: Remove from $supList$ agents farthest from $geoInt$
16: **else**
17: Add agents from $(P_t \setminus supList)$ closest to $geoInt$
18: **end if**

different than $sizeSup$, it is corrected, see lines 14–18, according to a simple distance criterion. We have already shown the effectiveness of the SAT solvers usage in article [8].

5.2. Outline of the Remaining Rules

The other behavioural rules are outlined briefly below:

- Selection for an ordinary intervention: the MC selects an observing patrol which is closest to the location of the planned intervention and calls it for going to an intervention. If in the assumed radius/area, there is no observing/available patrol, we wait until patrols are released from the current interventions so that they could undertake a new/subsequent intervention. A selection radius may be conditioned on a district (good/bad) and day time (day/night). The HQ may increase/decrease additionally these set searching parameters.

- during a firing incident, a police officer from the patrol may get injured, then such a patrol does not come back to its service;

The following, see Figure 8, is marked on the city map in different colours: patrols in different states, patrols excluded after the loss of a police officer, ambulances, drones.

6. Simulation Results

6.1. Basic Assumptions

Figure 9 shows the environment in which the experiments were performed. Said environment consists of two basic components, namely, a system supporting the work of the police and an environment imitating the urban environment in the sense of generating interventions, turning some of them into shootings, etc. The basic guidelines concerning simulation rules are given below. Patrols move slowly in the observation mode, stopping occasionally. When a patrol is called to intervene, then a drive takes place based on the familiarity with parameters for the

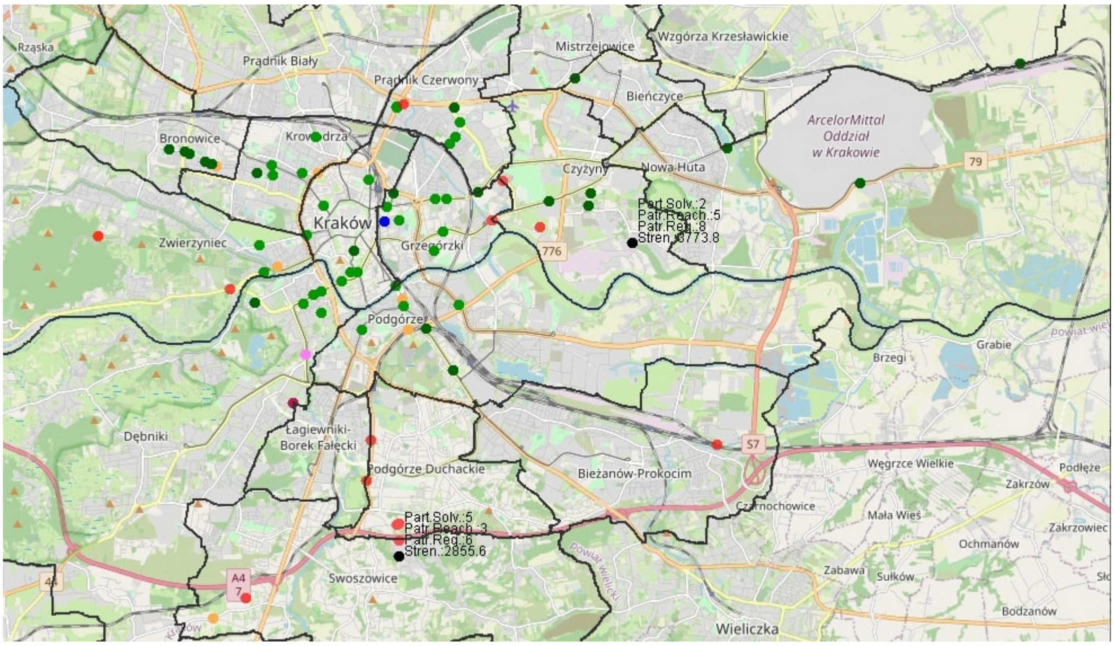

Fig. 8. System simulation screenshot. (Coloured points: dark blue – HQ; green – observing patrols; dark green – patrols returning to the HQ after finishing their service; orange – transfer to intervention; red – transfer to firing; blue – patrols after the termination of intervention, absent on the screen; violet – patrols after termination of firing; white – neutralized, absent on the screen; pink – patrols which choose the route of further movement; red – a simple intervention that did not turn into a shootout; black – firing)

navigation subsystem, indicating a driving route, from the patrol starting point to the target point of intervention, the drive time is randomised concerning the distance. The intervention duration is chosen at random. The probability of a regular intervention changing into a firing incident is chosen at random; moreover, the time is selected at random, after which a standard intervention is likely to change, if applicable, into a firing incident; firing or chasing duration is randomised as well. All the above parameters are predefined but can be changed within their ranges before starting the simulation.

6.2. Course of the Simulation

The system was subject to simulation in order to verify the model created and its assumptions [12]. The simulation maps the real 24 hours in a shorter period, i.e. events during the simulation happen faster (speed-up) than it would be in the real world. Figure 8 presents an exemplary panel of the simulation observer; it is similar to the panel found in the command centre, which is available for the HQ agent. The coloured points denote an event generated; the black colour refers to the most dangerous event connected with a firing incident, that is with the highest priority; the green colour indicates regular observations, and red points represent regular interventions.

The simulator enables the creation of many diagrams depicting a situation on the monitored

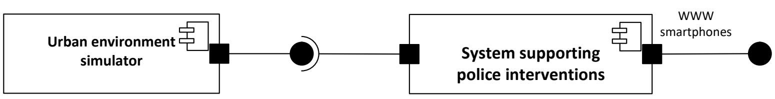

Fig. 9. Police supporting system and the urban simulator as separate components

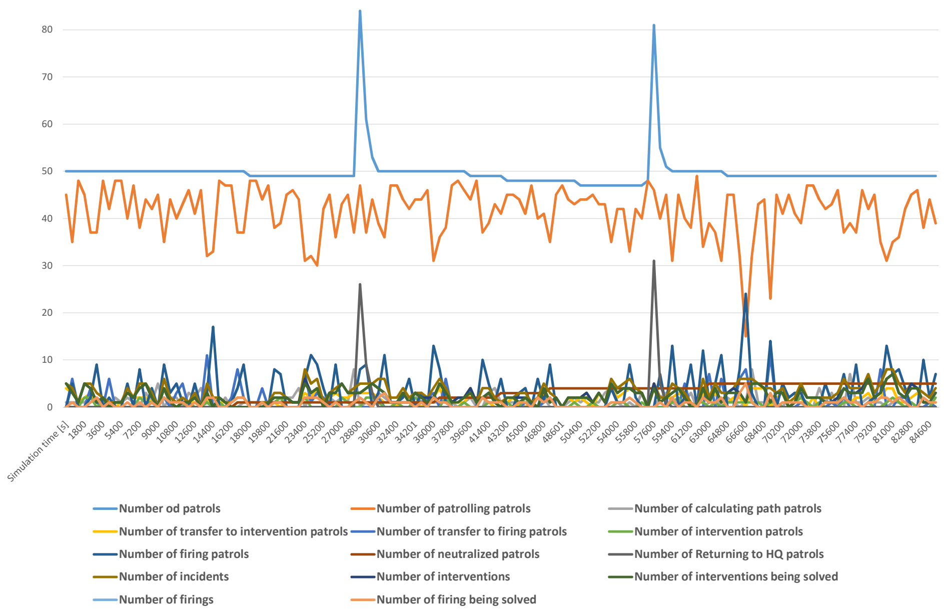

Fig. 10. The analysis of the police officers' involvement in the simulation. (Vertical axis – the current number of police officers, horizontal axis – subsequent simulation steps)

area. An example here is Figure 10 which presents police officers on duty on the monitored area and their involvement in the entire simulation process. Yet this figure shows many different events, see the caption in the bottom. The two peaks observed are due to a change of patrols, that is, some of them are already leaving the shift, and a new group is taking up the service. Figure 11 shows the course of the simulation with consideration with different city districts. The results show that the simulation is reliable and reflects reality well, that is in bad districts we have a greater number of shootings but also there are more patrols in such locations. Figure 12 shows the average values of the total spatial proximity for the different categories of city districts. *Spatial proximity* means proximity and approaching by objects on the space under consideration. This shows the influence of contextual data on the system and decision making. On the other hand, the cumulative value of spatial proximity in bad districts is lower, which should come as no surprise as there are more patrols there, that is, more patrols are directed there.

Figure 13 shows how the patrol states changed during the course of the simulation, which gives a good image of the simulation assumption implementations. After the intervention, pa-

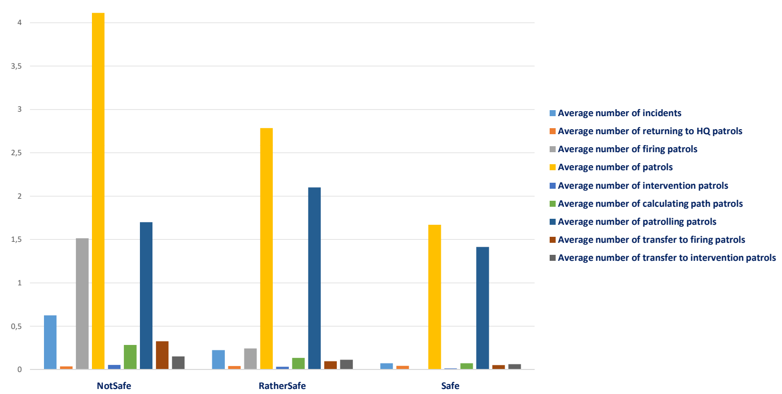

Fig. 11. Interventions in different city districts, worse districts are from left to right

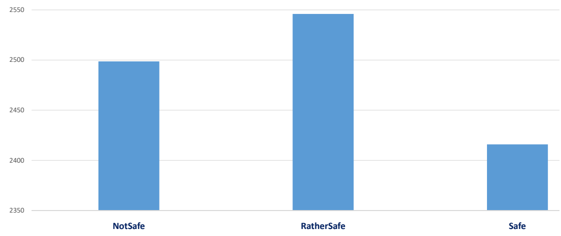

Fig. 12. Average values of the cumulative spatial proximity for districts

trols most often undergo observing/patrolling. From patrolling, we most often turn to interventions. An intervention always transfer to firing, unless the call was redundant. The transfer to intervention is sometimes interrupted by the transfer to firing due to a sudden need for support.

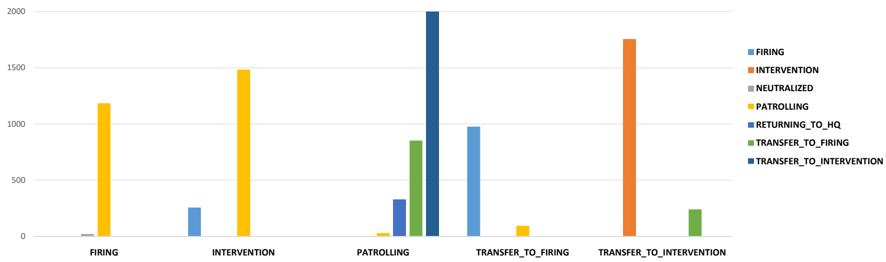

Fig. 13. Context transition, or state changes during the simulation. (Vertical axis – the current state, horizontal axis – number of state changes)

7. Conclusions

In order to ensure pro-activeness, the system processes complex contextual data; we proposed its categorisation corresponding to the field under design, and the entire framework with the data identified, so that it would be possible to obtain the clear image of its meaning in the system under design, to differentiate the individual roles, to plan the effective and efficient processes of contextual reasoning. We proposed a multi-agent system as the best environment for processing such contextual data.

The initially implemented simulation environment proves the feasibility of our proposal to context modelling and decision making in an environment with a rich set of contextual variables. We have not come across any significant efficiency limitations for average parameter values perhaps the planned stress tests will show some significant limitations of the system.

Further works will also consist in developing this environment to supply the full system analytics, to be able to affect the simulation process by any means, and also to research the phenomena characteristic for smart systems which operate in a smart city.

References

1. Bardram, J.E.: The java context awareness framework (jcaf) – a service infrastructure and programming framework for context-aware applications. In: Gellersen, H.W., Want, R., Schmidt, A. (eds.) Pervasive Computing. pp. 98–115. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
2. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing **6**(2), 161–180 (Apr 2010). https://doi.org/10.1016/j.pmcj.2009.06.002,

http://dx.doi.org/10.1016/j.pmcj.2009.06.002

3. Biegel, G., Cahill, V.: A framework for developing mobile, context-aware applications. In: Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications, 2004. pp. 361–365 (2004). https://doi.org/10.1109/PERCOM.2004.1276875
4. Cheng, Z.A., Dimoka, A., Pavlou, P.A.: Context may be king, but generalizability is the emperor! Journal of Information Technology **31**(3), 257–264 (Sep 2016). https://doi.org/10.1057/s41265-016-0005-7, https://doi.org/10.1057/s41265-016-0005-7
5. Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**(1), 4–7 (Jan 2001). https://doi.org/10.1007/s007790170019
6. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. In: Workshop on The What, Who, Where, When, and How of Context-Awareness (CHI 2000) (2000), http://www.cc.gatech.edu/fce/contexttoolkit/
7. Ferreira, D., Kostakos, V., Dey, A.K.: AWARE: Mobile context instrumentation framework. Frontiers in ICT **2**, 6 (2015). https://doi.org/10.3389/fict.2015.00006
8. Klimek, R.: Exploration of human activities using message streaming brokers and automated logical reasoning for ambient-assisted services. IEEE Access **6**, 27127–27155 (2018). https://doi.org/10.1109/ACCESS.2018.2834532
9. Klimek, R., Drwiła, Z., Dzienisik, P.: Proposal of a smart gun system supporting police interventions. In: Proceedings of 17th International Conference on Arificial Intelligence and Soft Computing (ICAISC 2018), 3–7 June, 2018, Zakopane, Poland. Lecture Notes in Artificial Intelligence, vol. 10842, pp. 677–688. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-91262-2_59
10. Kröhling, D.E., Chiotti, O., Martínez, E.C.: A context-aware approach to automated negotiation using reinforcement learning. Adv. Eng. Informatics **47** (2021). https://doi.org/10.1016/j.aei.2020.101229
11. Kwan, J., Gangat, Y., Payet, D., Courdier, R.: An agentified use of the internet of things. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 311–316 (Dec 2016). https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.76
12. Lepianka, S.: System for simulating police interventions in a smart city, Engineering diploma thesis, supervisor: Radosław Klimek, AGH University of Science and Technology (2022)
13. Maamar, Z., Faci, N., Boukadi, K., Ugljanin, E., Sellami, M., Baker, T., Angarita, R.: How to agentify the internet-of-things? In: 12th International Conference on Research Challenges in Information Science (RCIS). pp. 1–6 (May 2018). https://doi.org/10.1109/RCIS.2018.8406665
14. Perera, C., Zaslavsky, A.B., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: A survey. IEEE Communications Surveys and Tutorials **16**(1), 414–454 (2014)
15. Weiss, D.R.: Smart gun technologies: one method of eliminating unauthorized firearm use. In: 1994 Proceedings of IEEE International Carnahan Conference on Security Technology. pp. 169–172 (Oct 1994)
16. Zimmermann, A., Lorenz, A., Oppermann, R.: An operational definition of context. In: Proceedings of the 6th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT'07, Roskilde, Denmark. Lecture Notes in Artificial Intelligence, vol. 4635, pp. 558–571. Springer-Verlag (2007)