

# Detection of Movement Profiles in Facility Management using Complex Event Processing

Ralph Holland-Moritz<sup>\*</sup>, Ralf Vandenhouten

Technical University of Applied Sciences Wildau, Wildau, 15745, Germany

---

**Abstract** This paper presents a solution for detecting movement and spreading profiles by analyzing sensor data in the facility management. First of all, a scenario is defined and described. On the basis of this scenario an approach is made using integration of events over time and location. For making the integration possible the required information, which the data objects have to provide, is identified. At the end a solution is presented using Complex Event Processing to implement the integration of events.

**Keywords** Facility Management, Complex Event Processing, Sensor, Detector, Movement Profiles, Spatial Dimension, Integration, Alarm Reliability, Event Driven Architecture, Evaluation, Security, Intrusion, Fire

---

## 1. Introduction

In the context of the research project named InSeM (Intelligent Security Management for facility management solutions) an existing security and information system was extended by an intelligent component to detect complex signal patterns and enabling the early detection of potential risk of danger. The underlying security and information system is a flexible, plugin-orientated, modular and scalable client-server system, which is for condition monitoring and controlling of different safety systems being able to integrate different types of connected sensors and safety systems. The research project InSeM dealt with the extension of this security and information system in the form of an intelligent evaluation system using the sensor data of the connected sensors and generating abstract information. In a first phase of the project a flexible architecture based on Complex Event Processing (CEP) was proposed in [1]. In the solution, which is proposed in the present paper, the Complex Event Processing is used as intelligent component for increasing the alarm reliability and detecting movement and spreading profiles to reach some of the main project objectives and supporting decision-making processes in facility management systems.

Prior the application of the used security and information system at the location in question the system has to be adapted to the local situation. Among other things layout plans of the local conditions are created and the modules for the different device types are configured. In the scope of

this proposed solution a fictive scenario is defined which can be seen in figure 1.

The scenario covers a sample building which consists of four rooms. The four rooms are attached to each other in pairs and connected by a door. In addition, every room has an outer door and two windows.

Every room is equipped with multiple sensors of different types. This includes glass-breakage, contact, fire/smoke detector sensors and motion detection sensors. Each window is equipped with a glass-breakage sensor for detecting a possible forced entry. Both, doors and windows are equipped with closing/opening contacts from which it can be inferred whether a door or window is opened or closed. Motion detectors in every room enable the possibility to detect movements.

For detecting possible fire in the rooms, they are equipped with different fire detection sensors. For detecting smoke, there are nine smoke detectors in every room, which are arranged in a 3x3 pattern. For supporting the detection of fire, there is a temperature sensor installed in every room which measures the room temperature.

The aim of the proposed solution is the increasing of the alarm reliability and the detection of movement and spreading profiles. The increase of the alarm reliability is achieved mainly by decreasing the number of false alarms. Hereby it must be considered that with a too strong desensitization, which means a lowering of the threshold, alarm relevant sensor events could be overlooked. The increase of the alarm reliability can be achieved by increasing the probability of detection and decreasing the probability of false alarms. With the detection of spatial dimensions in the form of movement and spreading profiles it can be detected how a fire spreads spatially, or how burglars move through the building.

---

\* Corresponding author:

rholland@th-wildau.de (Ralph Holland-Moritz)

Published online at <http://journal.sapub.org/ijis>

Copyright © 2013 Scientific & Academic Publishing. All Rights Reserved

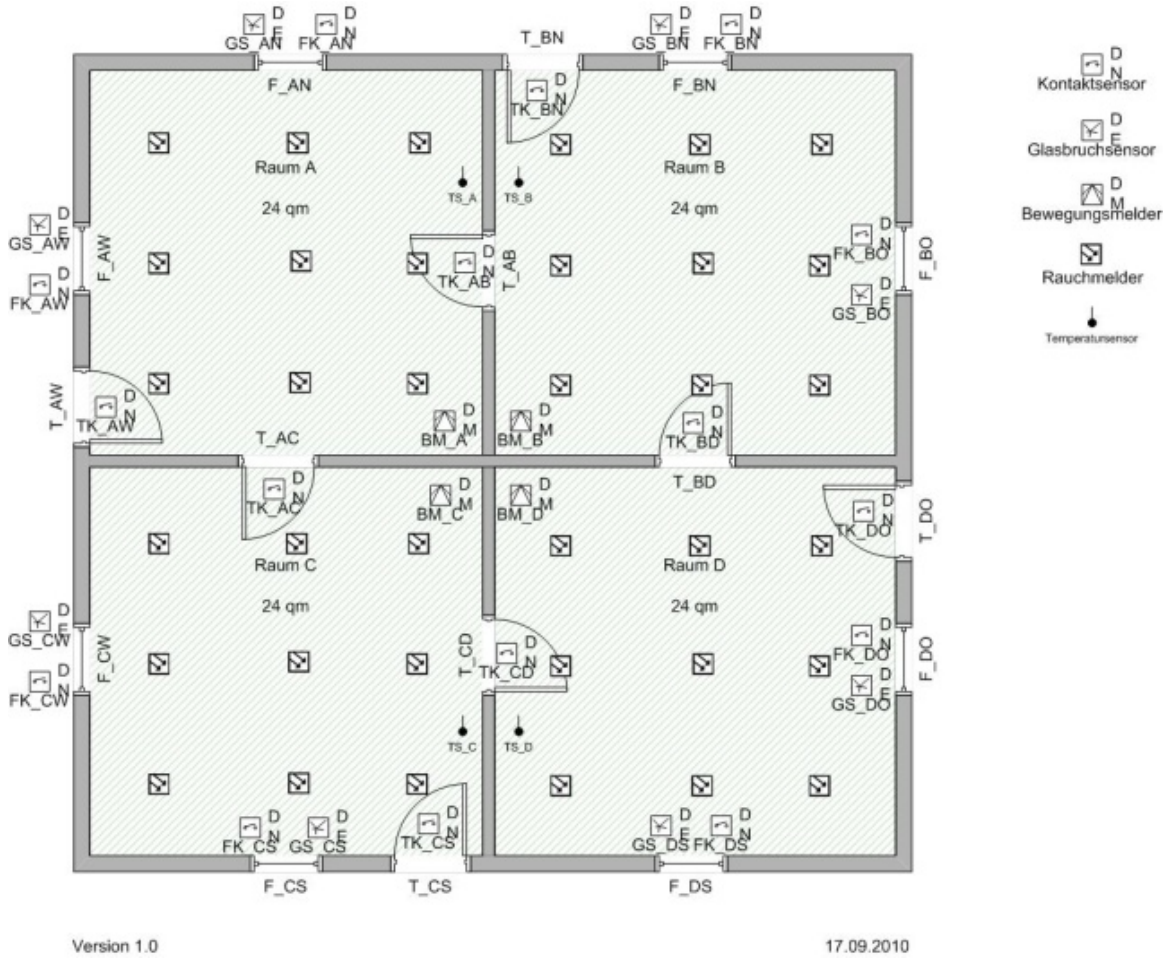


Figure 1. Fictitious ground plan

## 2. Approach

For increasing the alarm reliability and for detecting movement and spreading profiles there exist different methods dealing with sensor integration and sensor fusion. Beside the increase of the alarm reliability in special the potential advantages in integrating and/or fusing information from multiple sensors are that the information can be obtained more accurately, concerning features that are impossible to perceive with individual sensors, in less time, and at a lesser cost. Redundant information from a group of sensors is used, which refers to the same features in the surrounding. In place of a group of sensors a single sensor can be considered over a period of time for generating redundant information.[2] In this proposed solution the redundancy is reached by the methods integration over the time and integration over the location.

With the integration of the measured parameter over the time it is assumed that a measured quantity, being relevant for the detectable dangers like fire or intruders, is present for a certain time span. In contrast, for a disturbance, which would cause a false alarm, it is assumed to occur for a short time span. With sensors which measure a parameter continuously the average over a defined time span can be determined and a serious alarm can be triggered on

exceeding this average.[3]

With simple detectors which internally evaluate measurands, indicate a danger and stay in an alarm state the generation of multiple alarms over a short period of time can be forced by resetting the internal state by a central system. When exceeding a specific number of simple alarms it can be assumed that a serious alarm occurred.

With the method of the integration of the event over the location multiple sensors with known relative positions and overlapping supervision area are taken into account. With detectable events, which are measurable over a larger surface or volume it is assumed that neighbored or grouped sensors almost simultaneously measure the same risk.[3]

When the integration over time and location are combined a location- and time-dependent detection of events is possible which forms the basis for detecting movement and spreading profiles.

For applying the presented procedures specific information about the sensors must be available. For the integration over the location there must be some sort of location information. A simple location information in the form of a coordinate is not sufficient for detecting neighbored triggerings. Therefore the information about the coverage of a sensor is defined as a property for a sensor object. It is defined as the area in which the sensor acquires

the measurand. For differentiating which types of risk, like fire or intrusion, can be detected by evaluating the measurement a sensor type is defined as property of a sensor object. As result a sensor object is defined as to be seen in figure 2.

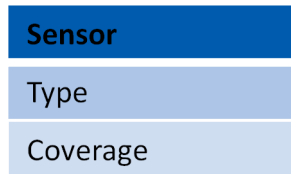


Figure 2. Properties of a sensor

The state of a sensor or the current measured value is projected as a sensor state event. When there is a state information for a sensor, a new sensor state event is generated. For enabling the integration over the time, time information is needed for an occurring event. For knowing for which sensor the event was generated and for accessing the information about the sensor a reference to the triggering sensor is needed. The property state is defined to project the specific state at a defined time. Depending on the kind of sensor, it can be represented different in nature beginning from continuous states, like a temperature, to abstract states like „idle“ or „alarm“. A sensorstate event is thus defined as seen in figure 3.

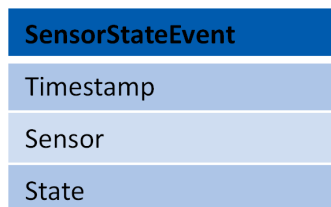


Figure 3. Properties of a sensor state event

The fire and smoke detectors being used are connected via a fire alarm system and stay, in case of a regular triggering, in the triggered state, unless they are manually reset. This means that only a single generation of a sensor event happens. For getting multiple triggerings in a short period of time and thus enabling the integration of the measurand over the time, a mechanism is needed which resets the detectors immediately after the triggering. This can be achieved with the help of a CEP-Engine, which is described below, by triggering an action in the management system, when a sensor state event with the state “triggered” occurs, resetting the detector to the state „idle“ immediately. Alternatively this effect can be achieved by internal mechanisms of the management system. As result one gets several triggerings over a longer time span when a serious danger is present.

### 3. Solution

Most facility and risk management systems that exist on the market, like GEMOS[4] or WotanEX[5], are built upon a Service Orientated Architecture (SOA) and are structured on the basis of services.[6] They implement rules for showing

and recording value exceedances or sensor triggerings programmatically in source code. Especially in the field of sensor evaluation, where sensor readings can be assumed as events and processes are event-driven, an Event Driven Architecture (EDA) is recommended.[6]

With the software-technical implementation of the integration over the time and location it is required that statements with temporal rules can be formulated. The Complex Event Processing, which is a software technology belonging to the architecture style EDA using events as central elements, supports such temporal rules.[7] In the Complex Event Processing events and the detection of known patterns take center stage. The intention is to detect the occurrence of interdependent events and make abstract statements by using complex events.[8] An event query language is used to process the events and evaluate them continuously.[9] With the help of these event query languages complex conditions can be defined, which represent the relations between the events. A specific type of event query language is the data stream query language, which is based on query languages of relational databases like SQL (Structured Query Language)[10]. In the following solution examples are presented in the form of EPL (Event Processing Language), which is the query language of the CEP solution Esper[11].

A multi-level evaluation is used to implement the scenarios with the help of complex event processing. On the lowest level there are elemental events, which directly represent the sensor readings. On the next level a pre-processing takes place, which determines whether there are alarm-relevant events by checking the sensor values. Alarm events are generated on this level in case of a relevant event. On the highest level, which is the most abstract, an evaluation of movement and spreading profiles takes place. The evaluation levels are shown in figure 4.

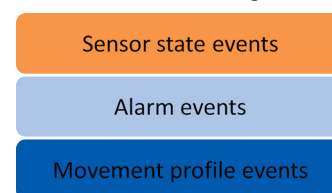


Figure 4. Evaluation levels for detecting profiles

#### 3.1. Integration of the Measurand over the Time

For validating whether a detector in the system was wrongly triggered and in effect generating a false alarm the elemental sensor events are evaluated in the CEP engine and filtered based on specific criteria. A class is registered at the CEP engine to be informed when two sensor events of the same sensor and the state “triggered” occur within ten seconds. This is reached by using the following statement:

```
every (a=SensorStateEvent(state=TRIGGERED) -> SensorStateEvent(sensor=a.sensor,state=TRIGGERED) where timer:within(10 sec))
```

The class is informed every time when two successive objects of kind “SensorStateEvent” with reference to the

same sensor („sensor=a.sensor“) and the state “triggered” („state=TRIGGERED“) occur within ten seconds („timer: within(10 sec“) as presented in figure 5. Subsequently an alarm event is created, which refers to the triggering detector and is provided to the CEP engine.

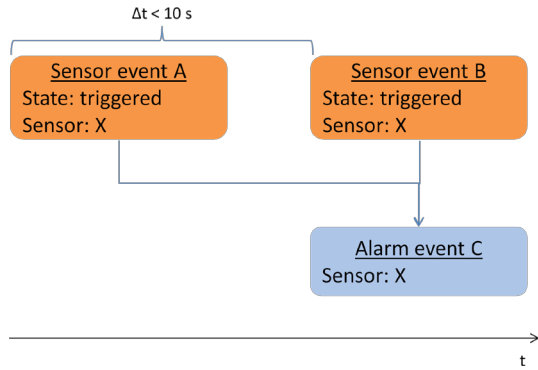


Figure 5. Connection between sensor event and alarm event

### 3.2. Integration of the Measurand over the Location

For the integration of sensor events over the location the spatial coverage information of the sensor is used. A class was implemented which represents the spatial coverage of a sensor. It is equipped with a function which enables the determination of overlapping with other sensors. In order to integrate sensor events over the location a class is registered to be informed when two events occur within a defined time having a reference to sensors of the same type and having overlapping coverage. This is done by using the following statement:

```
every (a=AlarmEvent -> b=AlarmEvent(sensor.covers(a.sensor),
sensor.type=a.sensor.type) where timer:within(10 sec))
```

This statement fires when two alarm events of the same alarm type („sensor.type = a.sensor.type“) occur within ten seconds („timer:within(10 sec“) and the referenced sensors with the same sensor type having overlapping coverage („sensor.covers(a.sensor“)“).

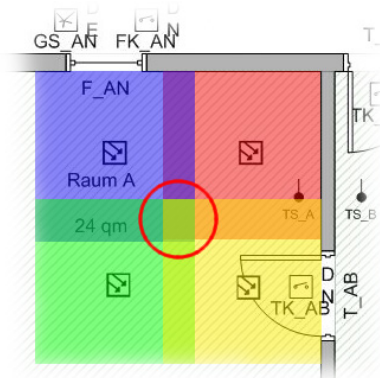


Figure 6. Overlapping supervision areas of smoke detectors

In order to make the integration over the location work the coverage of the underlying sensors which are installed in the surveillance building has to be designed redundantly. This means that the coverage of a sensor has to overlap with another sensor of the same type as it can exemplarily be seen

in figure 6.

This figure shows a picture detail of figure 1 with four neighboring smoke detectors. The colored areas show the visualized coverage of each detector that is saved as property for the smoke detector instances. One can see that the supervision areas overlap each other especially in the center which is highlighted by the red circle.

### 3.3. Detection of Spatial Dimensions

The detection of spatial dimensions is based on the information about spatial coverage of the sensors. Events are considered to be linked in a temporal and spatial context. Two cases, which are to be detected, are differentiated here - fire spreading and movements.

With the detection of fire spreading it is assumed that a fire spreads like a cloud from a starting point. This means that for a fire profile all alarm events are considered which occur in a temporal and spatial context of previous fire events. All fire events together form the current fire spreading. A fire spreading profile is created if at least two fire events occur in a temporal and spatial context. An existing fire profile is updated if a new fire event is triggered by a sensor, which is not part of the profile but in the spatial vicinity of a sensor, which is already part of the profile. In order to determine whether an alarm event occurs in the vicinity of a fire profile the class representing a fire profile has a method “borders(AlarmEvent)” which returns whether a passed alarm event is in the near of sensors belonging to the profile. For updating and creating fire spreading profiles the following pattern is used:

```
every ((ae=AlarmEvent or fpe=FireProfileEvent) ->
(ae2=AlarmEvent(sensor.type=FIRE,sensor.covers(ae.sensor)) or
ae2=AlarmEvent(fpe.borders(ae2),sensor.type=FIRE)) where
timer:within(10 min))
```

In the first place a fire profile (“FireProfileEvent”) is searched, which is followed by a fire event within ten seconds („timer:within(10 min“) being in the vicinity of the fire profile („fpe.borders(ae2“)“). When there is an according fire profile found the new alarm event is added and the updated fire profile is passed to the CEP engine as new event in order to be used for a new check of the pattern. When there is no fire profile in the vicinity of a fire event (“ae2”) but a previous fire event (“ae”) a new fire profile is created and passed to the CEP engine.

For defining the time between two events a minimal fire speed of 0.15 m/min according to DIN 18232 is assumed. Therefore, a fire needs approximately 7 minutes to spread one meter. With a sensor coverage of 2 meters and an overlapping of 0.5 meters it takes 10 minutes max. between two sensor triggerings.

When detecting a movement profile of a burglar it is about associating subsequent intrusion events, which follow an initial intrusion event, to a movement profile. This profile has information about the initial event, the last event and the events in-between in the form of a chain of events. Thus it is a special case of the fire spreading because every new

movement can only be made from the last triggered sensor to a neighbored one having overlapping coverage. Since the burglar can pass a sensor multiple times, in contrast to fire spreading, the chain of events can include multiple events referring the same sensor.

For creating a movement profile two neighbored sensor triggerings are searched for, which occurred one after the other with in a defined time. The movement profile is updated when an alarm event occurs in temporal and spatial relation to the last event of the chain of events. The time span depends on the time of the retriggering and the time a burglar needs for moving from one sensor to another. The pattern for creating and updating the movement profile is as follows:

every ((me=MovementEvent or ae=AlarmEvent) -> (ae2=AlarmEvent(sensor.covers(ae.sensor), sensor.type=HOUSEBREAKING) or ae2=AlarmEvent(sensor.covers(me.last.sensor),sensor.type = HOUSEBREAKING)) where timer:within(30 sec))

In the first place it is searched for a movement profile (“MovementEvent”) where the last alarm event of the chain is in the vicinity (“sensor.covers(me.last.sensor)”) of a following intrusion event (“AlarmEvent(sensor.type = HOUSEBREAKING)”), which occurs within 30 seconds

(“timer:within(30 sec”)”). When there is an according movement profile, the alarm event found is added as last event to the chain of events of the movement event as it can be seen in figure 7. Subsequently the updated movement profile is passed as new event to the CEP engine for allowing the pattern to match again. If there wasn't a previous movement profile the pattern matches for two subsequent events within 30 seconds, which occur in the vicinity of each other. With this alternative case a new movement profile is created having the two alarm events as beginning and end event.

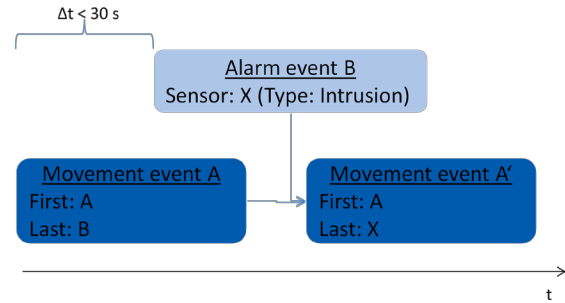


Figure 7. Update of a movement profile

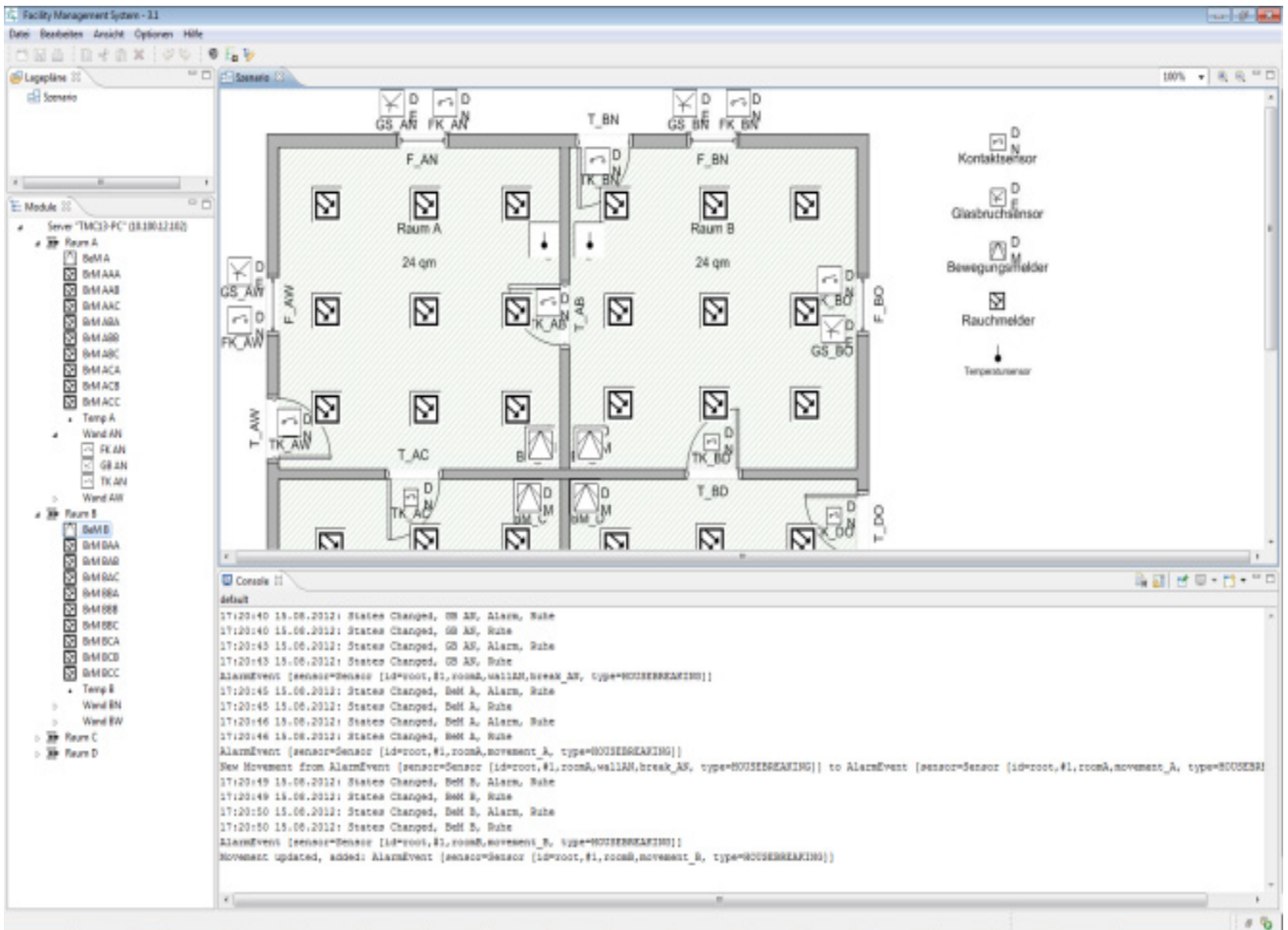


Figure 8. Movement profile of a burglar

The movement event provides information about the point a burglar entered the building, what route he had taken and at which point he left it. Because the movement profile is updated closely to real-time, depending on the chosen parameters, the information of the movement profile can be used to support the arresting of a burglar.

Figure 8 shows exemplarily the progress of a movement profile where the burglar moves from the north wall of the room A via room A to room B. In the right lower console window one can see the creation of a new movement profile and the addition of a new event to the existing movement profile.

## 4. Conclusions

A solution for detecting movement and spreading profiles on the basis of sensor data in the facility management was presented. It was used and prototypically implemented to detect spreading profiles in the research project InSeM. Because of a similar approach the solution is suitable for the seamless integration into an intelligent management system based on the proposed solution for software architecture in[1].

With the help of the additionally gained information about the spreading of fire and movements of burglars the security staff can be supported in better assessing the situation and take suitable measures. By reducing false alarms the desensitization of the staff can be counteracted leading to an increment of security in facility management.

## ACKNOWLEDGEMENTS

The project InSeM has been sponsored by the German Federal Ministry of Education and Research (BMBF) in the funding program FHprofUnt.

## REFERENCES

- [1] Ralph Holland-Moritz, Ralf Vandenhouten "A flexible architecture for intelligent management systems", LINDI 2011 3<sup>rd</sup> IEEE International Symposium on Logistics and Industrial Informatics, pp.83-86, 2011.
- [2] Ren C. Luo, Michael G. Key, „Multisensor Integration and Fusion in Intelligent Systems“, IEEE Transactions on Systems, Man, and Cybernetics, vol. 19, no.5, pp.901-931, 1989.
- [3] Hans-Rolf Tränkler, Ernst Obermeier, Sensortechnik: Handbuch für Praxis und Wissenschaft, Springer, Berlin, 1998.
- [4] Online available: <http://www.ela-soft.com/gemos.php>
- [5] Online available: <http://www.wotanex.de/>
- [6] Jürgen Dunkel, Andreas Eberhart, Stefan Fischer, Carsten Kleiner, Arne Koschel, Systemarchitekturen für Verteilte Anwendungen: Client-Server // Multi-Tier // SOA // Event Driven Architectures // P2P // Grid // Web 2.0, Hanser, München, 2008.
- [7] Ralf Bruns, Jürgen Dunkel, Event-Driven Architecture: Softwarearchitektur für ereignis gesteuerte Geschäftsprozesse, Springer, Berlin, Heidelberg, 2010.
- [8] David Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison-Wesley, Amsterdam, 2002.
- [9] Michael Eckert, Francois Bry, "Complex Event Processing (CEP)", Springer, Informatik Spektrum, vol. 32, pp.163-167, 2009
- [10] F. Burger, P. Debicki, F. Kötter, Vergleich von Complex Event Processing-Ansätzen für Business Activity Monitoring, [http://elib.uni-stuttgart.de/opus/volltexte/2010/5258/pdf/FA\\_CH\\_0112.pdf](http://elib.uni-stuttgart.de/opus/volltexte/2010/5258/pdf/FA_CH_0112.pdf), 2010.
- [11] Online available: <http://www.espertech.com/products/esper.php>.