



HAL
open science

TiTiMaKe: A Distributed Service Architecture for Security Applications

Mauno Rönkkö, Markus Stocker, Markku Huttunen, Ville Kotovirta, Mikko Kolehmainen

► **To cite this version:**

Mauno Rönkkö, Markus Stocker, Markku Huttunen, Ville Kotovirta, Mikko Kolehmainen. TiTiMaKe: A Distributed Service Architecture for Security Applications. 10th International Symposium on Environmental Software Systems (ISESS), Oct 2013, Neusiedl am See, Austria. pp.381-390, 10.1007/978-3-642-41151-9_36 . hal-01457468

HAL Id: hal-01457468

<https://inria.hal.science/hal-01457468v1>

Submitted on 6 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

TiTiMaKe: A Distributed Service Architecture for Security Applications

Mauno Rönkkö¹, Markus Stocker¹, Markku Huttunen², Ville Kotovirta², and Mikko Kolehmainen¹

¹ Department of Environmental Science, University of Eastern Finland,
PO Box 1627, 70211 Kuopio, Finland,

{mauno.ronkko,markus.stocker,mikko.kolehmainen}@uef.fi

² VTT Technical research Centre of Finland,

PO Box 1000, 02044 VTT, Finland,

{markku.huttunen,ville.kotovirta}@vtt.fi

Abstract. In this article, we discuss integration of distributed information services, to support the implementation of applications for fields of safety and security. Although there are standard interfacing technologies, such as SOAP and REST, there is very little architectural support for the development of personalized and case specific applications, where the same data sources and computational services are integrated in different ways. As the main contribution, we present here a distributed service architecture, called TiTiMaKe, that supports the development of scenario based applications. We also illustrate TiTiMaKe by describing its use in applications in fields of surveillance and emergency rescue.

Keywords: Distributed Systems, Internet Computing, Service Networks, Safety and Security

1 Introduction

In this article, we discuss information system solutions to crisis prevention and incident management. These phases appear in various models for crisis management, such as in the crisis management chain by Carter [1] and in the relational model by Jaques [2]. In crisis prevention and incident management, enhanced perception by using state-of-the-art sensory technology coupled with computational methods can significantly improve situational awareness [3], that is, the comprehension and projection capabilities for the actors involved in the action. Also, through specifically tailored systems architecture, the enhanced communication capabilities can significantly improve the shared situational picture, leading to faster and more accurate responses.

Applications for crisis prevention and incident management are classified as safety critical systems. According to Knight [4] "safety-critical systems are those systems whose failure could result in loss of life, significant property damage, or damage to the environment". Safety-critical systems require significant effort

in specification, architecture design, implementation verification, and processing. In particular, reliability, efficiency, and resilience are key factors for the design. In this respect, Internet computing and cloud computing offer solutions for improving not only robustness and performance, but also extensibility and adaptation.

When developing service-based applications, however, there is virtually no support for features like adaptation, personalization, and automated reasoning at runtime. The same applies, if there is a need for data manipulation and transformation on-the-fly. Currently, the support is found at the technical level of integration. For instance, modern programming environments do support code generation for service integration, especially for standard interfacing techniques like SOAP and REST. Not surprisingly, some work on identifying integration patterns [5] has been done recently.

In this article, as the main contribution, we present an extensible, distributed service architecture, called TiTiMaKe, that supports the development of scenario based safety and security applications. The TiTiMaKe architecture is based on core logic design, where design patterns [6], such as factory, bridge, facade, chain-of-responsibility, mediator, observer, thread pool, and messaging patterns, are applied. The architecture not only supports the integration of new data sources and computational services, but also their concurrent use in multiple scenarios. Concurrency issues are resolved with an internal event queue and attachable event listeners running in multiple threads. Because of this, the architecture supports also the reuse of components throughout different scenario implementations. This is also how the TiTiMaKe architecture differs from some existing architectures for safety and security applications, such as UHHA [7]. In this article, we also illustrate the use of the TiTiMaKe architecture in two scenarios that use the same sensor network as the data source, but integrate it with very different kinds of computational services. Both scenarios were implemented and tested for functioning.

The rest of the article is organized as follows. We start by briefly describing the used information services in Section 2. In Section 3, we describe the TiTiMaKe architecture. In Section 4, we illustrate the use of TiTiMaKe in implementing two security applications. Finally, in Section 5, follows the conclusion.

2 Distributed Service Network

The service network considered in this article consists of multiple, heterogeneous services. These are the ToMoVaKe sensor network, Vehicle identification service, Dispersion modeling service, Reachability analysis service, and Spatio-temporal population modeling. We shall now briefly introduce each of these services.

The ToMoVaKe sensor network operated by Lappeenranta University of Technology, LUT Savo Laboratory of Green Chemistry, comprises of three zones: two control centers in two cities (Mikkeli and Kuopio) and an exercise field at the training site of Emergency Services College (PEO). There are many kinds of sensors at the field connected to the sensor network. The sensors include, among

others, vibration sensors, acoustic sensors, weather stations, surveillance camera systems, and chemical detectors. The network supports versatile communication technologies, including wireless communications protocols, fiber optics, and Zigbee. The ToMoVaKe sensor network provides both an interface for accessing raw measurement data as well as a user interface framework for applications.

The University of Eastern Finland (UEF) and the VTT Technical Research Centre of Finland (VTT) have developed a non-intrusive vehicle classification service [8] based on vibration measurements. The service is accessed on-line and it uses trained neural networks to classify vehicles based on preprocessed vibration measurement data. The service runs in real-time. It can be trained for different environmental conditions, including annual weather changes, temporal climate changes, and sustained changes in ambient noise.

Aalto University (Aalto) has developed an on-line reachability analysis service. It computes the reachability analysis for both on-road and off-road vehicles. Therefore, it uses not only road network data, but also forest and landscape vegetation data. It computes a reachability map based on location data and vehicle type classification. The map is provided as raster data.

The Finnish Meteorological Institute (FMI) has developed a on-line chemical dispersion modeling service, called ESCAPE [9, 10]. It computes the dispersion forecast based on weather data, type of the chemical, and other properties, such as emission location and altitude. The computed dispersion map contains also estimated concentration values. The map is provided as vector data.

Aalto University (Aalto) has also developed an on-line spatio-temporal population modeling service [11]. The service computes an estimation of inhabitants in a region at a given time of a day. The computation is, thus, dynamic taking into account variations in population dynamics during different times of the year and day. The estimation is based on building and road data. It differentiates between building types (e.g. office or residential). The service provides population estimation in XML format based on given location parameters.

3 TiTiMaKe Architecture

The TiTiMaKe architecture is depicted in Figure 1. It consists of the TiTiMaKe server, ToMoVaKe system, FMI server, Aalto server, and PEO sensor network. ToMoVaKe stores the measurements. It also provides messaging and query services that are used by the TiTiMaKe server to fetch the measurements. The TiTiMaKe server contains the core logic, which is responsible for running the application workflow. The core logic is used to detect anomalies in the measurements and to send alarms to ToMoVaKe through the Sensor ML interface. The core logic is also used to initiate relevant computations on FMI and Aalto servers over HTTP interfaces. The TiTiMaKe server relays computational results to ToMoVaKe for visualization. This logical decomposition is also depicted in Figure 1.

TiTiMaKe uses design patterns [6], such as factory, bridge, facade, chain-of-responsibility, mediator, observer, thread pool, and messaging. The architecture

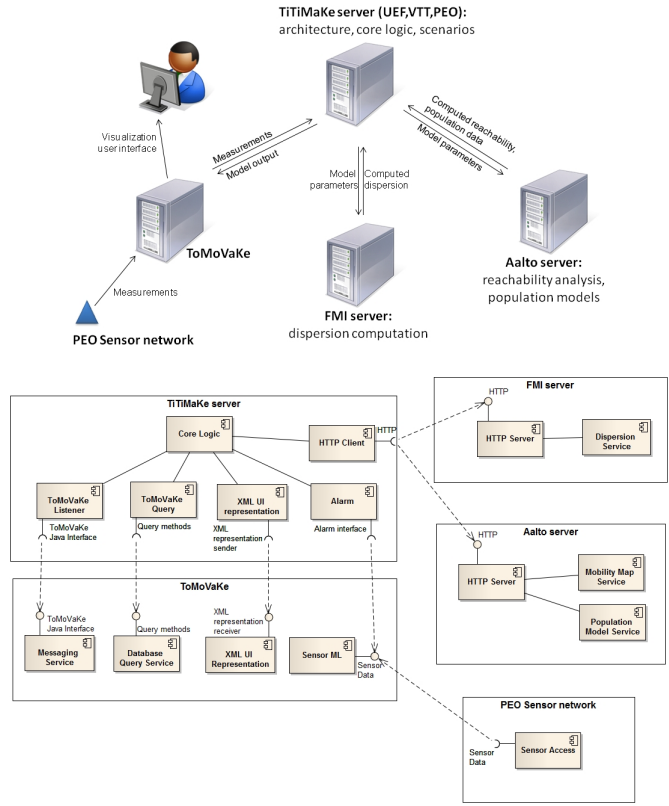


Fig. 1. TiTiMaKe architecture and logical decomposition.

supports the integration of new data sources and computational services, as well as their concurrent use in multiple scenarios. Concurrency issues are resolved with an internal event queue and attachable event listeners running in multiple threads. Because of this, the architecture supports also the reuse of components throughout different scenario implementations.

Figures 2 and 3 represent the domain view and the dynamic view of the TiTiMaKe core logic, respectively. The core logic has monitors, each of which executes a single scenario. A monitor has an associated listener that implements the scenario by orchestrating calls to distributed modeling services, integrating their results, and communicating with ToMoVaKe to present the modeling results. Listeners use detectors to detect certain events that are produced by sensors. For instance, a sensor could be collecting classified vehicle observations to a queue accessed by a detector. If the observations in the queue correspond to messages of interest, the detector forwards the observations to the listener. Note that the implementation of listeners, detectors, and sensors are specific to each scenario.

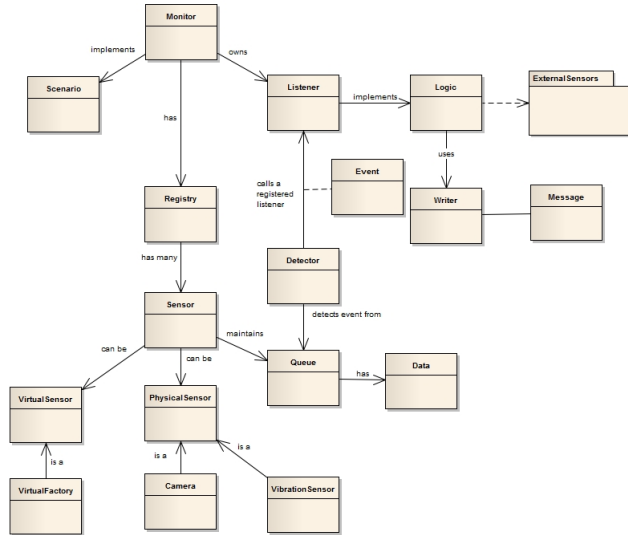


Fig. 2. Domain view of the TiTiMaKe core logic.

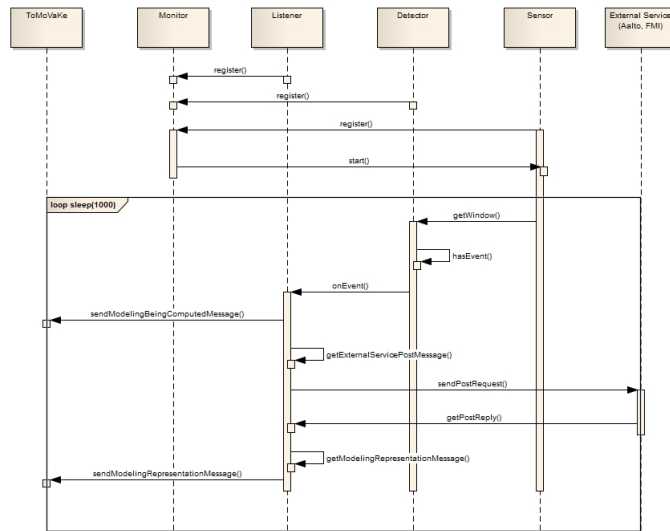


Fig. 3. Dynamic view of the TiTiMaKe core logic.

4 Case Studies

We present now two scenarios that were implemented using the TiTiMaKe architecture: "Unauthorized driving on a private road", and "Fire at an industrial site with chemicals".

4.1 Unauthorized driving on a private road

The scenario starts with vibration sensors detecting a vehicle on a private road. The system alerts about this by flashing a red icon on top of map on screen. At the same time, the system initiates a reachability computation, to find out how far the vehicle could drive within the next few minutes. Once the reachability computation is completed, the results are shown on the map. The person on duty can also bring up pictures of the vehicle as taken by the surveillance camera. The pictures are fetched when the person on duty clicks the flashing red icon. Based on the images of the vehicle, and the reachability results, the person on duty can either identify the vehicle or raise an alarm, if vehicle cannot be identified. Figure 4 depicts this scenario. The implementation of the scenario is depicted in Figure 5 as a sequence diagram. The communication logic of the sequence diagram was implemented as sensor, detector, and listener components in the TiTiMaKe architecture.

4.2 Fire at an industrial site with chemicals

This is a simulated scenario. The scenario starts when a fire in a factory causes an automatic fire alarm. The system fetches data about the factory and informs the person on duty and firemen about a danger of a chemical hazard. Upon an arrival, the firemen inform the control center about toxic smoke, whereby the system informs about potential toxins and visualizes a computed estimate of dispersion. Based on this, the control center starts to plan the need for additional resources and procedures. Based on a continuous on-site measurement activity, the system alerts the control center about how toxic the smoke is and computes using weather data an estimated dispersion and concentration on the map. Next, the system fetches population data within the concentration area and computes an estimate for the number of affected people and for potential risk groups. Then, the system produces an evacuation and alert message for the interest groups. The scenario is depicted in Figure 6.

As this scenario included externalities, such as an industrial site, chemical containers, and firemen, the implementation of the scenario was limited to its core functionality. The implementation is depicted in Figure 7 as a sequence diagram. In particular, the implementation focused on the interaction between the TiTiMaKe server and the computational services, namely dispersion modeling computed by the FMI server and dynamic population modeling computed by the Aalto server. The logic of the sequence diagram in Figure 7 was implemented as sensor, detector, and listener components in the TiTiMaKe architecture.

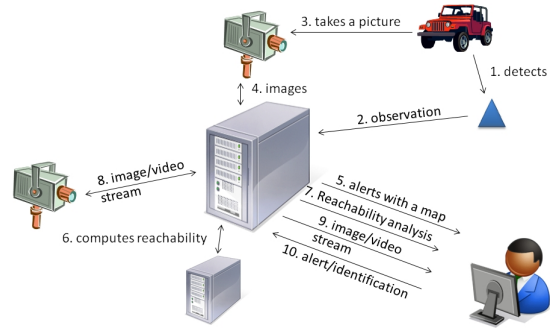


Fig. 4. Unauthorized driving on a private road.

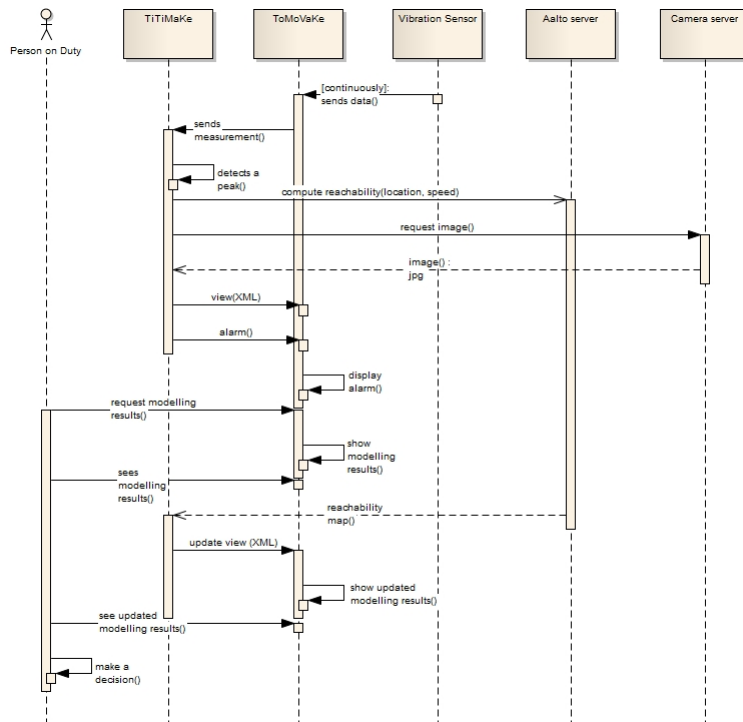


Fig. 5. Implementation of the scenario "Unauthorized driving on a private road" depicted as a sequence diagram.

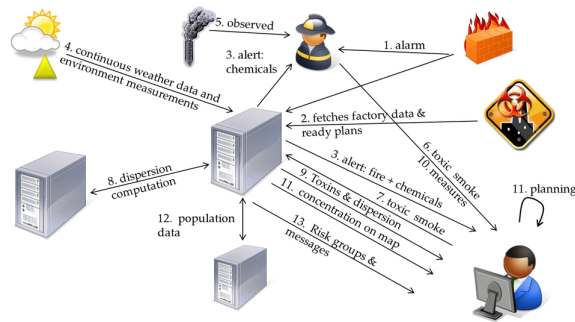


Fig. 6. Fire at an industrial site with chemicals.

5 Conclusion

In this article, we discussed the integration of distributed information services to support the implementation of applications for the fields of safety and security. To support the development of such applications, and as the main contribution, we presented here the TiTiMaKe architecture. We used the architecture to implement two scenario based applications: "Unauthorized driving on a private road", and "Fire at an industrial site with chemicals".

The implementation of both scenarios confirmed that the TiTiMaKe architecture does indeed support the development of scenario based applications by using a distributed service network. The tests, however, revealed how sensitive such a system is to defects, if robustness and resilience are not considered in the communication logic. To overcome such defects, central services of the system need to be duplicated or communication with minimal information has to be guaranteed some other way, for instance by providing alternative communication media. The TiTiMaKe architecture supports both of these measures.

One clear topic for future research is to integrate more services to the TiTiMaKe architecture and thereby to implement different kinds of scenario based applications. In this way, the core logic of the architecture becomes more versatile, thus supporting the development of more advanced safety-critical applications.

Another topic for future development has to do with the centralized aspect of the TiTiMaKe architecture. In current form, the core logic has a central supervisor. Although parts of the logic can be distributed, the supervisory logic must reside in a central server. This raises the issue of resilience, if the central server becomes unaccessible. An alternative is to investigate cooperative decision making by implementing loosely coupled autonomous servers that cooperate by communication. Such an arrangement would facilitate not only better robustness and resilience, but also better performance scaling when the number of concurrent applications increases.

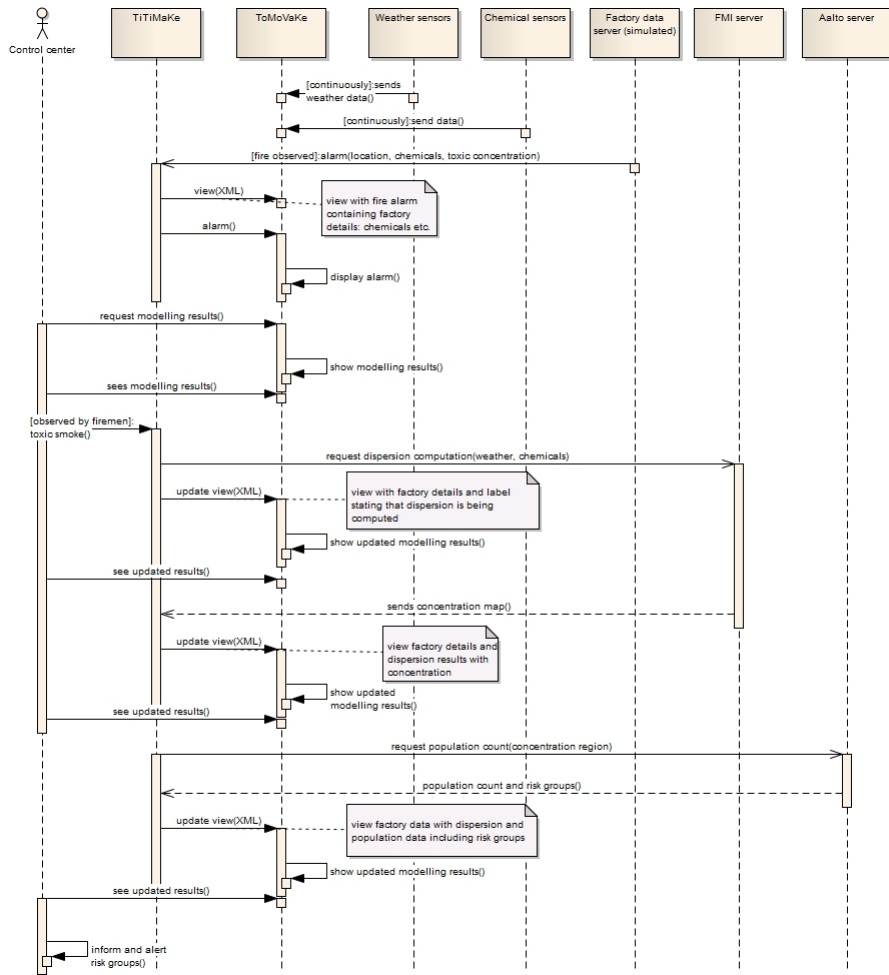


Fig. 7. Implementation of the scenario "Fire at an industrial site with chemicals" depicted as a sequence diagram.

Acknowledgements

We wish to thank the Finnish Funding Agency for Technology and Innovation (TEKES) and the industrial partners (APL Systems Oy, Control Express Finland Oy, Environics Oy, Finnish Defence Forces Research Centre, and Symo Oy) for funding this research. We also wish to thank all our academic partners (Aalto University, Emergency Services College, and Finnish Meteorological Institute) for excellent work and support during the project. Mauno Rönkkö and Mikko Kolehmainen would also like to thank European Regional Development Fund and TEKES for support to continue research in cooperation with Amphion Loudspeakers Oy, Genelec Oy, Mediatuotanto Mindcom Oy, Observis Oy, and Defence Forces Technical Research Centre. Last but not least, we wish to thank Heikki Isotalus and Jarmo Kastinen for technical support.

References

1. Carter, W., N.: Disaster management: A disaster manager's handbook. Asian Development Bank, Manila (1991)
2. Jaques, T.: Issue management and crisis management: an integrated, non-linear, relational construct. *Public Relations Review*, 33, 147–157 (2007)
3. Salmon, P., M., Stanton, N., A., Jenkins, D. P., Walker, G., H., Young, M., S., Aujla, A.: What really is going on? Review, Critique and Extension of Situation Awareness Theory. *Engineering Psychology and Cognitive Ergonomics*. LNCS 4562, Springer-Verlag (2007).
4. Knight, J., C.: Safety critical systems: challenges and directions. *Proceedings of the 24rd International Conference on Software Engineering, ICSE 2002*, 547–550 (2002)
5. Hohpe, G., Woolf, B.: *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley (2003).
6. Gamma E., Helm, R., Johnson, R., John Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1994)
7. Molarius, M., Huovila, H., Kotovirta, V., Wessberg, N., Yliaho, J.: UHHA - a framework for emergency management in chemical accidents, In *Proceedings of 7th Symposium on CBRNE Threats. Meeting the future challenges*, Defence Forces Technical Research Centre, Finland, Publications 18, 112–116 (2009)
8. Stocker, M., Silvonen, P., Rönkkö, M., Kolehmainen, M.: Detection and classification of vehicles by measurement of road-pavement vibration and by means of supervised learning. Submitted to *IEEE Transactions on Intelligent Transportation Systems* (2012)
9. Kukkonen, J.: Modelling source terms for the atmospheric dispersion of hazardous substances. *Commentationes Physico-Mathematicae* 115, *Dissertationes* No. 34. The Finnish Society of Sciences and Letters, Helsinki, (1990)
10. Riikonen, K., Nikmo, J., Kukkonen, J.: A validated assessment tool for consequence analysis of chemical emergencies - ESCAPE for Windows. In: Kujala, E., Laihia, K., Nieminen, K. (eds.) *Proceedings of the NBC 2000 Symposium on nuclear, biological and chemical threats in the 21st century*, 286–287 (2000)
11. Zhang Z., Sunila R., Virrantaus K: A spatio-temporal population model for alarming, situational picture and warning system. *Joint International conference on theory, data handling and modeling in Geospatial information science* (2010)