Smart solutions for risk prevention through analysis of people movements

Smart Solutions for Risk Prevention through Analysis of People Movements

MariaGrazia Fugini[1], Stefano Pinardi[2], and Claudia Raibulet[2]

[1] Politecnico di Milano, Dipartimento di Elettronica e Informatica, Via Ponzio, 34/5, 20133, Milan, Italy

[2] Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione, Viale Sarca 336, Ed. U14, 20126, Milan, Italy

fugini@elet.polimi.it, {pinardi,raibulet}@disco.unimib.it

Abstract. The high number of accidents in living areas, work environments, and ambient, in general, can benefit of prevention mechanisms able to identify the causes and the indications which precede accidents, and to put in place strategies to avoid risks whenever this is possible. To this aim, this paper presents a risk management architecture for monitoring movements within a smart ambient and managing possible risks. In particular, we present a methodology for movement analysis aimed at detecting and preventing risks. Results from experimentations are discussed.

Keywords: risk, emergency, risk management, smart environments.

1 Introduction

One of the goals of a smart environment is to support and enhance the abilities of its occupants in executing tasks. In particular, in living areas, work plants, or construction environments, these tasks can present a high level of risk, ranging from moving around in a (potentially dangerous) space to interacting with objects/machines and handling (potentially dangerous) tools.

To support the occupants in facing possible risks, the concept of *smart environment* is introduced along with concepts of self-healing features and ambient intelligence [2, 15]. Sensors and devices can be distributed in the areas to detect the current state (*context)* of the environment and determine what actions should be taken to face or prevent risks (self-healing system) [5, 13, 19, 21]. The context consists of any information that can be used to characterize the situation (or state) of an *entity*, namely, a person, a place, a physical or computational object, or a machine, a tool, a protection kit or a sensor network. This information can include personal movements, physical gestures, relationships between persons and objects in the environment, features of the physical environment such as spatial layout and temperature, and identity the location of people and objects in the environment.

This paper proposes an approach to *location-based risk analysis* in combination with a *movement recognition method*. To create a smart environment, where such an

M. Rautiainen et al. (Eds.): GPC 2011 Workshops, LNCS 7096, pp. 3–13, 2012.

approach is possible, smart tags can be employed, such as sensors, RFID, or smart phones that can be carried or worn by persons to allow the detection of their location, their proximity to potentially dangerous sources and to determine the risk they are exposed to at a given instant. In particular, we propose a method enabling to detect what risks can possibly arise due *to the movements* of a person in the area. We focus on the use of *inertial sensors* for such monitoring. Inertial sensors are low cost devices that measure linear forces and torque forces, and which can be worn on body. These are becoming smaller and smaller, so increasing the user acceptance of body-worn sensors. Movement classification with inertial sensor showed to be useful in many areas like ambient intelligent and healthcare applications, in neuroscience and for tracking activities [1, 12, 15, 18]. Inertial sensors can also be used for human computer interaction, gait and posture analysis, and motion capture, or to understand emotional status from body posture [3, 10]. A large literature exists about the use of inertial sensors to classify movements, with different results. Many of the proposed methodologies are specific to problems or applications or to the given technology. Hence, sometimes they are hardly to translate into practice, or they use minimal dictionaries (3-6 actions), hence not allowing for the use of a method that could be reused in different contexts [1, 23]. Recently, a new technique called *Distributed Sparsity Classifier* has been proposed [23], where a public database of movements is introduced, called WARD 1.0, which has been made available and which we used to compare and test the results of our approach. Based on information we can obtain from inertial sensors, we define applications that use the context to provide task-relevant information and/or services to a user, hence to be context-aware [7]. In this paper, we show how information collected from the environment can be conveyed to a *Risk Management System* able to compute and re-distribute information signalling risks and emergencies, where emergencies are defined as situations that have a high danger and must be faced immediately. Risk is instead defined as a situation which evolves smoothly (such as a gas loss or a moving machine) which can be notified and prevented.

This paper is organized as follows. Section 2 introduces our architectural perspective to risk management. Section 3 presents how movements in our application domain are monitored and the methodology for movement analysis, with tests and results and ICS and ECS (intra cluster inter cluster) measures. Section 4 presents a sample scenario. Conclusions and further work are given in Section 5.

2 An Architectural Perspective towards Risks

Our *Risk Management System* (*RMS*) (see Figure 1) exploits the Monitoring, Analyzing, Planning, Executing (MAPE) loop [6, 11]. This loop observes the environment, detects the anomalies by analyzing the data collected through the monitoring step, decides if a risky situation occurred and if intervention/change is needed and of what type, and puts in place (executes) the planned modifications in the environment. The main tasks of the MAPE risk control loop are:

1) *Monitoring* is continuously performed through a set of sensors (e.g., RFID sensors, video cameras, inertial sensors) and devices (e.g., PDAs, PCs) called here *informative devices*. Such devices are distributed in the environment (e.g., on the machines/tools, on the persons and on equipment) and collect data regarding for example the level of gas, the movement of machineries and persons, the operations performed using a tool (e.g., a hammer, a water container), or the health conditions of persons.
2) *Analysis* on the monitored data verifies if data are out of ordinary values (e.g., the gas level is >= a threshold, or a machinery is moving towards an area where persons are working, the hearth pressure of a person is in the normal range).
   A threshold is a point value or a range which delimitates regions of risks from the normal/ordinary values. If the *Risk Threshold* is respected, the system operates normally. In between the *Risk* and the *Emergency Threshold*, the system is operating in a risky status, where preventive actions can still be put in place to prevent the risk. Beyond the *Emergency Threshold*, the system operates in *emergency* [8], and corrective actions are applied.
3) *Planning* is performed when in one (or more) monitored element is out of the admitted ranges. In the planning, our solution associates a risk evaluation function to each element in the environment. By combining all the risks evaluations, we determine if we are in face of a risk, an emergency, or a false alarm. In case of risk, in the planning, the *RMS* selects the most suitable strategy (a set of interventions) able to reduce or eliminate the risk.
4) *Executing* applies the strategies identified in the planning phase. Upon application of the strategies, the loop continues with the monitoring phase.

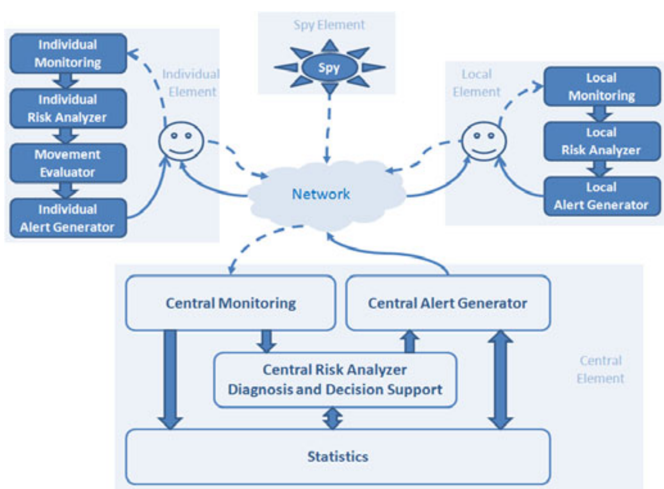

Fig. 1. Risk Management Architecture

The risk model supported by the architecture shown in Figure 1 is based on modeling explicitly both risk and environment entities meaningful for risk computation [7, 8]. In particular, the risk model is person centred, in that it relates the risk to the user behaviour and movements (work activities in our case study). We have presented our approach in [7, 8, 9]. In this paper, we deal with *monitoring the persons' activities* using body-worn informative devices (see next sections).

3 Monitoring Persons' Movements

There are many specific advantages in using body-worn sensors instead of video cameras for movement classification. Inertial sensors can be placed on specific segments of the body or in clothes accessories. Consequently, we know with absolute certainty which segment of the body the data refers to, and we do not have to solve "hidden parts" problems, nor solve color and luminance issues typical of video cameras. Also, we do not have to segment the body from the other environment information, or to disambiguate and interpret data when many people are present in the same area [4, 14, 17]. On the other hand, there is no general methodology to classify movements that can be used in different situations or with different technologies. The usual approaches are very specific to the given situation. After an accurate study of the Dictionary of actions [16, 20], very specific features are chosen and used. When we change the Dictionary, the recognition accuracy is not guaranteed.

In this paper, we show why the method of *FFxIVFF* significantly improves accuracy of movement classification by using inertial sensors. The method exploits a procedure, called *FFxIVFF*, which automatically changes the movement features weighting in function of the movements [16, 20]. *FFxIVFF* transforms the space, lowering, in the Dictionary, the importance of those features that are less discriminative, and incrementing those features that are more frequent within a given class. The procedure automatically adapts the feature weighting with respect to the given Dictionary, and hence, can be used in different contexts. Also, it greatly improves accuracy with respect to other recent concurrent methodologies, providing a general instrument for *movement classification with inertial sensors*. A movement classifier of this type can be used for example to understand which type of actions are executed by a person in an environment, such as a construction plant or an industrial area where there are people at work, given a proper Dictionary of actions as showed in [16, 20]. The ICS and ECS (inter cluster intra cluster) measures, given in the Section 3, shows that the method improve clusters density and separation, explaining why the *FFxIVFF* works well with inertial sensors. Also, in this research, this kind of classifiers is used at run-time in an environment in order to understand if a risky situation arises, in dependence not only of the person's location but specifically of the action the person is executing. For example, a person holding a lighter where free inflammable gas is present constitutes a risky situation, but until the lighter is off the risk is relatively low. An actual risk arises when a person tries to ignite it in the environment. A *movement classifier* with inertial sensors can understand this kind of risky situation, which depends on the person's actions and on the situation, and can help avoiding it. The benefits of intervening in such risky circumstances and avoid catastrophic situations are evident.

3.1 Methodology for Movement Classification

For movement analysis, we used five MTx inertial sensors of XSens [22]. Each MTx sensor has three components: an accelerometer, a gyroscope, and a magnetometer with three degrees of freedom, providing information on acceleration (+/- 50 m/sec2), rate of turn (+/-2 rad/sec), and earth-magnetic field (+/-1 normalized) in a three-axial reference system. In order to recognize a movement, we need a *database of movements* to create our test set and to generate the *FFxIVFF* [16, 20]. Consequently, a specific set of actions is executed by a set of people and features are extracted from the generated data, action by action, sample by sample using an iterative procedure. In order to obtain these data, we employ a tool which simulates data transmission from sensors (details are presented in [9]). In particular, each component – accelerometer, gyroscope, and magnetometer – has three dimensional data (X,Y,Z). Every datum is also considered in its 2D and 3D norm representation (|XY|, |XZ|, |YZ|, |XYZ|). Then, data are filtered using eight transformation functions (null, smoothing, low pass, mean, variance, variance with low pass, first derivative, second derivative) generating 840 different signals from the original ones. These transformation functions are very generic, and are quite common in the movement classification area. Then, for each transformation 10 generic features are chosen generating 8400 features. Feature values are quantized into 22 intervals for a total of 184.800 intervals: when hit a specific interval is marked 1, otherwise is left to 0. Hence, every action generates a sparse vector of 184.800 binary values. The given values are stored in a vector. Values are quantized, and *FF* and *IVFF* are calculated. Hence, any action is transformed into a *n*-dimensional vector in a *n*-dimensional Feature-Action space. The set of quantized features-action vectors of an action executed by a population and reweighted by *FFxIVFF* constitutes the pattern of the given type of action. When a new action is executed, we transform it (using a run-time procedure) into a vector of the Feature-Action space using the same methodology. Then, we calculate how similar it is to the given type of actions, using a similarity measure. In particular we use three popular similarity measures: a Ranking algorithm (Eq.1), an Euclidean Distance (Eq.2), and a Cosine Similarity (Eq.3). The formulas are the following:

$$\mathrm{rank_j} = \sum_{i=1}^{n} \mathrm{W_{i,j}} \tag{1}$$

$$\mathrm{dist_i} = \sqrt{\sum_{j=1}^{n} \left(\mathrm{W_{i,j}} - \mathrm{q_{i,j}}\right)^2} \tag{2}$$

$$\cos\theta = \frac{\mathrm{W_{i,j}} \cdot \mathrm{q_{i,j}}}{|\mathrm{W_{i,j}}||\mathrm{q_{i,j}}|} \tag{3}$$

where $\mathrm{W_{i,j}}$ represents the weight of the σ_i interval of action a_j of the Training-Set, and $q_{i,j}$ is the IVFF value associated to the feature of query.

3.2 The *FFxIVFF* Transformation

In this space, the centroids of the actions cluster are too close to each other, making the recognition a problem. In order to increase the accuracy, and use algorithms that are fast and feasible in run-time applications but are still reliable in term of accuracy, we transform the feature action space into a different space by appropriately reweighting the features values (see [15, 16]). Hence, two weights have been introduced: the *FF* ("Feature Frequency") and the *IVFF* ("Inverse vocabulary frequency"). The *FF* formula:

$$\mathrm{FF}_{\mathrm{i,j}} = \frac{\mathrm{n_{i,j}}}{|\mathrm{P}|} \tag{4}$$

where $\mathrm{n_{i,j}}$ is the number of occurrences of the σ_{i} feature in the action $\mathrm{a_j}$, and $|\mathrm{P}|$ represents the population cardinality. The *IVFF* formula is:

$$\mathrm{IVFF}_{i} = \log \frac{|\mathrm{A}|}{|\{\mathrm{a}\colon \sigma_{\mathrm{i}} \in \mathrm{a}\}|} \tag{5}$$

where $|\mathrm{A}|$ represents the cardinality of the vocabulary, and $|\{\mathrm{a} : \sigma_{\mathrm{i}} \in \mathrm{a}\}|$ the number of actions a where feature σ_{i} assumes values. The overall weight of a single feature is given by the multiplication: $\mathrm{W_{i,j}} = \mathrm{FF_{i,j}} * \mathrm{IVFF_i}$. The *FF* takes into account how frequent is a feature in the given population rising the importance of the features that appear in the same class of movements (Eq.4). The *IVFF* takes into account how frequent is a feature in the dictionary: a feature that is present in more actions is considered less discriminative, and its weight is lowered according to the formula (Eq. 5.). We note that the *FFxIVFF* space is different from the original values space: some dimensions can be canceled or enhanced depending on the role of features in the dictionary and population. The *FFxIVFF* transforms the original space into a space where clusters appear to be denser and more separated. This approach allows reaching high accuracy (the highest in literature as far as we know), using very simple and fast algorithms of similarity.

3.3 The Datasets

In order to test the proposed method, we used two databases: NIDA 1.0 (Nomadis internal Database of Actions) the database of the Nomadis Laboratory at the University of Milano-Bicocca (see [20]) and the WARD 1.0 (Wearable Action Recognition Database) created at UC Berkeley [23]. These databases are quite representative of a typical database of actions, since they are large (respectively 273 and 1200 samples) and have a large number of actions (respectively, 13 and 21). They are quite different for issues such as sensors technology, dimension of the vocabulary of actions, number and type of samples, and granularity of low level data. NIDA 1.0 contains 21 types of actions for a total of 273 samples. These actions are:

> 1. Get up from seat. 2. Get up from a chair. 3. Open a wardrobe. 4. Open a door. 5. Fall. 6. Walk forward 7. Run. 8. Turn left 180 degrees. 9. Turn right 180 degrees. 10. Turn left 90 degrees. 11. Turn right 90 degrees. 12. Karate frontal kick. 13. Karate side kick. 14. Karate punch. 15. Go upstairs. 16. Go downstairs. 17. Jump. 18. Write. 19. Lie down on a seat. 20. Sitting on a chair. 21. Heavily sitting on a chair.

WARD contains 13 types of actions performed by 20 people ranging from 20 to 79-years-old with 5 repetition per action, for a total of 1200 actions. The list of actions is the following:

1. Stand (ST). 2. Sit (SI). 3. Lie down (LI). 4. Walk forward (WF). 5. Walk left-circle (WL). 6. Walk right circle (WR). 7. Turn left (TL). 8. Turn right (TR). 9. Go upstairs (UP). 10. Go downstairs (DO). 11. Jog (JO). 12. Jump (JU). 13. Drive truck (PU).

3.4 Tests and Results

Even if the two databases are technologically different, the methodology has been applied – as is – without significant changes to both the databases, using a Leave One Out Cross Validation (LOOCV) methodology. Recognition accuracies are very high in both cases. The classification accuracies using the NIDA 1.0 database (273 samples of 21 type actions) are the followings: Ranking 89.74%, Euclidean Distance 95.23%, Cosine similarity 95.23%. The classification accuracies of algorithms using the WARD 1.0 database are the followings: Ranking 97.5%, Euclidean Distance 97.74%, Cosine 97.63%. We have to note that the accuracy of our methodology applied to the WARD 1.0 database outperforms their results [see 23]. The UC Berkeley researchers reached and accuracy of only 93.4 % using their databases. In particular, we reduced by three times the error rate (from 6.6% to 2.26%).

Other tests have been done to test the accuracy reducing the number of worn sensors, to check the sensitivity to sensor numbers [16]. Using WARD 1.0 with 3 sensors on the pelvis, the right wrist, and ankle, we reached an accuracy of 97.63% (with the cosine similarity), and using only one sensor we reached a 93.62% of accuracy (with cosine similarity), against the 93.4 % of U.C. Berkeley obtained on the same database with *all* the 5 sensors.

In order to understand why the proposed method has far better accuracy than former methodologies, we tried to understand which effects *FFxIVFF* has on dimension of clusters both in WARD 1.0 and NIDA 1.0 multidimensional spaces.

At this aim, the Intracluster (ICS) and Intercluster (ECS) have been calculated, and measures confirm that the *FFxIVFF* creates a space where clusters are more dense and well separated, justifying the higher accuracies obtained in the tests sections with the LOOCV methodology. Intracluster and Intercluster similarity have been measured in the values space, then in the *FF* space and finally in the *FFxIVFF* space. Cosine similarity has been used as a similarity measure. Results have been compared and showed in Table 1.

Note that *FF* and *IVFF* actually enhance cluster density and cluster separation, respectively. In particular, using the NIDA database, we can note that:

- *IVFF* enhances the Intercluster similarity 5.96 times wrt *FF*, and 12.23 times wrt the values space;
- *FF* enhances the Intracluster similarity, while thanks to *FFxIVFF* Intracluster similarity passes the critical threshold of 0.9.

While using WARD database we can note that:

- *IVFF* enhances the Intercluster similarity 4.08 times wrt the values space
- *FF* enhances the Intracluster similarity, while thanks to *FFxIVFF* Intracluster similarity passes the critical threshold of 0.9.

The Intracluster and Intercluster similarity confirm us the reasons why *FFxIVFF* works well. Clusters are more dense and further apart (the values of Intercluster similarity has been improved of one order), significantly reducing the probability of false positives. Thanks to that, it is possible to classify movements with greater accuracy even using a few sensors (three). Note that it is possible to recognize movements that involve parts of body that have not a sensor placed on it. The accuracy results, confirmed by ICS and ECS measures, sustains the idea that the *FFxIVFF* methodology is solid, and allow us to propose the use of this methodology in actual situations, even at runtime.

Table 1. Intracluster similarity before and after *FFxIVFF*

Database	**Type**	**Value Space (Cos)**	**FF space (Cos)**	**FFxIVFF space (Cos)**
Nida	Intercluster	**0,3506425**	0,1710151	**0,0286604**
	Intracluster	**0,7077682**	0,8759209	**0,9260479**
Ward	Intercluster	**0,2476419**	0,2367639	**0,0605941**
	Intracluster	**0,6038969**	0,8455223	**0,9311265**

The independence of technology, the possibility to use few sensors, and the feasibility of these results confirmed by the ICS and ECS measure, give us now the scientific ground for a real use of the method. In this case, we propose to use three body worn sensors, to understand which kind of movements has been executed, and to relate risk analysis to movements, as we can see in the next section.

4 Movements' Analysis in Risk Management: A Sample Scenario

In work areas, users employ tools or drive machines. We only examine risk related to movements of persons in handling these tools and machines (namely, we ignore the case of risks *generated* by tools and machines). We can monitor the persons' movements and check the risk which they are exposed to. We assume that inertial sensors are worn by persons and allow risk to be identified by the *Movement Evaluator* module of the Risk Management Architecture. We assume that other sensors, e.g., RFID for determining the location of the various entities - objects, persons - in the environment, are in place to detect other risks. Based on these affirmations, we are currently extending the database of actions with work specific activities. We associate a Risk Level to each work related action in the range [0..5] where risk increases from 0 to 5. Examples of clusters of actions bound to their Risk Levels will include:

1. Enter work area. 2. Go upstairs at the second floor. 3. Use hammer. 4. Turn left 90 degrees. 5. Stop working. 6. Go downstairs. 7. Leave the work area.
Risk Level = 3

1. Enter work area. 2. Step into the truck. 3. Drive the truck. 4. Stop the truck. 5. Leave the area.
Risk Level = 5

The Movement Evaluator module of our architecture (see Figure 1) performs a continuous monitoring of the persons/tools and machinery through both identifications (e.g., RFID) obtained by applying passive tags on instruments and inertial sensors. Through controlled points in the environment, the system is aware of which tool a worker is interacting with at a given moment and hence, executes given check operations. A catalogue of all work tools and their assignment to workers is available. The Movement Evaluator determines if a person is at risk by using a *RiskEvaluatio*n function. Such function analyses the parameters characterizing each person/tool and machinery interaction by: a) evaluating the data retrieved from the sensors; b) correlating such data to the Risk Level assigned to actions. The trend of the *RiskEvaluation* function is defined by a utility function: e.g., linear, logarithmic, exponential, sigmoid. We assume that the output of an evaluation is normalized and always included in [0..1]. Namely, we have:

$$\textit{evaluation\_value} = f(\text{parameter})$$

where f denotes the evaluation function and parameter represents the elements monitored to determine the risk, in this case the actions the person is performing with which tool and machinery. The *evaluation_value* is numeric and expresses the risk for each parameter. If it is beyond a given value (fixed for that particular parameter), a potential risk is signaled. For example, for the “closeness to moving crane”, the lowest value corresponds to the best quality, while the highest value corresponds to a risk event. In our approach, a set of function f are stored in the *Movement Evaluator* files. There may be more parameters for one entity (e.g., tools/machineries, persons, protection garments, informative device). Depending on the risky situation, there may be various solutions for their management. For example, a worker is notified about his position to close to the location of a truck in movement or can receive a warning when walking forward in the direction of a dangerous machine. He is then required to use another type of tool or to adopt another strategy to achieve the same objective/task. These solutions are called *strategies* and are defined as sequences of actions to be executed with different priorities depending on the evaluated Risk Level. We have implemented a sample preliminary set of strategies in a prototype which is described in [9].

5 Concluding Remarks

In this paper we have presented an overview of our approach to risk management in smart environments based on the analysis of persons’ movements. We have provided details on the method we adopt concerning the computation of persons' movements in the environment. From the architectural point of view, our solution exploits self-healing mechanisms by implementing the MAPE loop. Currently, our prototype for risk management addresses risks exploiting this loop. Its design allows the evolution towards the consideration of the detailed persons' movements, which we plan to integrate in the prototype in the next future.

Acknowledgments. This work has been partially supported by the TeKNE Project and by the S-Cube European Network of Excellence in Software Services and Systems.

References

1. Bao, L.: Physical Activity Recognition from Acceleration, Department of Electrical Engineering and Computer at the Massachusetts Institute of Technology, Master Thesis (August 2003)
2. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC® Sensor Web Enablement: Overview and High Level Architecture. In: Nittel, S., Labrinidis, A., Stefanidis, A. (eds.) GSN 2006. LNCS, vol. 4540, pp. 175–190. Springer, Heidelberg (2008)
3. Castellano, G., Villalba, S.D., Camurri, A.: Recognising Human Emotions from Body Movement and Gesture Dynamics. In: Paiva, A.C.R., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 71–82. Springer, Heidelberg (2007)
4. Cameron, J., Lasenby, J.: Estimating Human Skeleton Parameters and Configuration in Real-Time from Markered Optical Motion Capture. In: Proceedings of the 5th International Conference on Articulated Motion and Deformable Objects (2008)
5. Chen, N., Di, L., Yu, G., Min, M.: A Flexible Geospatial Sensor Observation Service for Diverse Sensor Data based on Web Service. ISPRS Journal of Photogrammetry and Remote Sensing 64, 234–242 (2009)
6. Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J.: Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525. Springer, Heidelberg (2009)
7. Conti, G.M., Rizzo, F., Fugini, M.G., Raibulet, C., Ubezio, L.: Wearable Services in Risk Management. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, Web2Touch Workshop, pp. 563–566 (2009)
8. Fugini, M.G., Raibulet, C., Ubezio, L.: Risk Characterization and Prototyping. In: Proceedings of the 10th International Conference on New Technologies of Distributed Systems (NOTERE 2010), Tozeur, Tunisia, May 31-June 2, pp. 57–64 (2010)
9. Fugini, M.G., Raibulet, C., Ramoni, F.: Service-Oriented Architecture for Risk Management. In: Proc. 11th Int. Conf. on New Technologies of Distributed Systems, NOTERE 2011, Paris, May 10-12 (2011)
10. Kleinsmith, A., Bianchi-Berthouze, N.: Recognizing affective dimensions from body posture. In: Paiva, A.C.R., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 48–58. Springer, Heidelberg (2007)
11. Laguna, M.A., Finat, J., González, J.A.: Mobile health monitoring and smart sensors: a product line approach. In: Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities To increase Digital Citizenship (EATIS 2009), pp. 1–8. ACM, New York (2009)
12. Lee, S., Mase, K.: Activity and Location Recognition using Wearable Sensors. IEEE Pervasive Computing, 24–32 (2002)

13. Lundgren, R.E., McMakin, A.H.: Risk Communication. In: A Handbook for Communicating Environmental, Safety, and Health Risks, 4th edn., Wiley Publ. (2009)
14. Moeslund, T.B., Granum, E.: A Survey of Computer Vision-Based Human Motion Capture. Computer Vision and Image Understanding (2001)
15. Mileo, A., Merico, D., Pinardi, S., Bisiani, R.: A Logical Approach to Home Healthcare with Intelligent Sensor-Network Support. The Computer Journal (2009)
16. Mileo, A., Pinardi, S., Bisiani, R.: Movement Recognition using Context: a Lexical Approach based on Coherence. In: Proceedings of the Sixth International Workshop on Modeling and Reasoning in Context (2010)
17. Okada, R., Stenger, B.: A Single Camera Motion Capture System for Human-Computer Interaction. IEEE Transactions on Information and Systems E91-D(7), 1855–1862 (2008)
18. Ohgi, S., Morita, S., Loo, K.K., Mizuike, C.: Time Series Analysis of Spontaneous Upper-Extremity Movements of Premature Infants With Brain Injuries. Phys. Ther. 88(9), 1022–1033 (2008)
19. Pederson, T., Surie, D.: Towards an Activity-Aware Wearable Computing Platform Based on an Egocentric Interaction Model. In: Ichikawa, H., Cho, W.-D., Chen, Y., Youn, H.Y. (eds.) UCS 2007. LNCS, vol. 4836, pp. 211–227. Springer, Heidelberg (2007)
20. Pinardi, S., Bisiani, B.: Movement Recognition with Intelligent Multisensor Analysis, a Lexical Approach. In: Workshops Proceedings of the 6th International Conference on Intelligent Environments, pp. 170–177 (2010)
21. Rubio, J., Padillo, I., Espina, J., Verástegui, J., López-de-Ipiña, J.: RFID in workplace safety solutions in Proceedings RFID Systech 2010. In: European Workshop on Smart Objects: Systems, Technologies and Applications, Ciudad, Spain, June 15-16 (2010)
22. XSens, `http://www.xsens.com`
23. Yang, A.Y., Jafari, R., Sastry, S.S., Bajcsy, R.: Distributed Recognition of Human Actions Using Wearable Motion Sensor Networks. Journal of Ambient Intelligence and Smart Environments (2009)