Annales UMCS Informatica AI XIV, 2 (2014) 60–75
DOI: 10.2478/umcsinfo-2014-0011

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Vector Approach to Context Data Reliability

Marcin Alan Tunia[1]*

[1] *Institute of Telecommunications of WUT,*
Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract – Context-aware processing is a part of intensively developed ubiquitous computing and mobile systems. Surrounding awareness is used to introduce new functions and solutions. Some categories of the context data are taken for security purposes in the context-aware security implementations. This kind of data has to meet some conditions since it is used for decision making about security mechanisms adaptation and configuration. One of these conditions is reliability. The paper presents vector approach to context data reliability assessment introducing mechanism which allows to assess reliability parameters for further usage in context aware security processing. The following aspects of the context data are taken into account: interface reliability, data quality, data source reliability and security level. Introducing reliability metric for context data may be beneficial to other mechanisms which utilize context data. The vector form of reliability may be even more flexible than the scalar value.

1 Introduction

Modern IT systems process great deal of data, that not only come from inside the system but also from outside. Complex integrated environments consist of highly specialized systems dedicated to perform certain tasks. These systems implement many interfaces for receiving data from other systems or data sources. Context data is one of data types. Once obtained it is being used for decision making and performing assigned tasks such as reasoning about environment and adapting security parameters according to current situation. It is possible to manipulate with system performance and output by altering input data. Considering many data sources and data providers that system may use to produce output, not every input data can be considered fully reliable. In the context security aware system every piece of data that comes from outside

*M.Tunia@tele.pw.edu.pl

through one of the interfaces should be evaluated with reliability mechanism before further processing. Assigning reliability rating for the piece of data should affect the way system processes that data. This approach may improve protection against data manipulation or results of other connected systems malfunction. This paper includes the vector method of assuring data reliability awareness based on various aspects of delivery: interface, data quality, source reputation and security services involved. The presented vector approach is compared to the single value approach.

This paper deals with context aware systems, which produce and execute decisions based on entities context. The entity may be user, system, resource or people in the neighbourhood. To determine context of each entity, required for decision making, the context aware system usually needs to contact several external agents. Not every agent may be considered as much trustworthy as other agents. Any context data manipulation (data removal, modification or insertion) may lead to improper context situation evaluation and finally to lowering or inflating service's security level. There is also possibility of manipulating with system performance and functions accessible for users. By lowering security level attacker may weaken system protection and gain point of entry. By inflating security level, when it is not necessary, attacker may cause service more difficult to use (for example, because of additional authentication methods or a lack of access to some functions).

Moreover, using external context data for decision making causes an additional risk of performing wrong actions on the basis of wrong data. To minimize both types of risk: security risk and the one connected with wrong decisions, context data evaluation mechanism may be implemented. Such mechanism should answer the following questions concerning the received context data:

- Is it what system expected to receive?
- Is it the exact response to the request?
- What source do data come from?
- What kind of protection was involved during data delivery?
- What channel was used for data delivery?

The paper is structured as follows. Section 2 describes the related work connected with context data, reputation systems and context security. Section 3 presents a general concept of context data reliability assessment and defines component blocks for the reliability vector mechanism. After defining of all parts of the reliability vector mechanism, the simple use case is introduced in Section 4. The paper is concluded with the summary and future work description in Section 5.

2 Related Work and Motivation

2.1 Context-Awareness

Context-aware computing first research is considered to be Active Badge Location System [**1**], published in 1992. It introduces individual badges worn by employees and

providing each person's location. In further studies there were many various definitions of context-aware system. In 1994 Schilit and Theimer [**2**] defined context-aware software as able to *ądapt according to its location of use, the collection of nearby people and objects, as well as the changes to those objects over time".* According to [**3**] context-aware computing was considered synonymous with the terms like: situated [**4**], reactive [**5**], adaptive [**6**], context-sensitive [**7**], responsive [**8**], environment-directed [**9**]. Apart from the Schilit's and Theimer's definition, research papers include various definitions of context:

- Brown et al. use the definition by enumeration: *łocation, identities of the people around the user, the time of day, season, temperature, and so forth."* [**10**],
- Ryan et al. similarly define context by examples: *"user's location, environment, identity, and time"* [**11**],
- Chen et al. understand context as *"the set of environmental states and settings that either determines an application's behaviour or in which an application event occurs and is interesting to the user."* [**12**],
- Schmidt et al. define context as *"knowledge about the user's and IT device's state, including surroundings, situation, and, to a lesser extent, location."* [**13**],
- Dey et al. suggest the following definition: *Ćontext is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."* [**3**][**14**].

While definitions by enumeration cannot be applied in many cases, it is better to use the general definition like the last three examples. For the purpose of this document the last listed definition is used. It is widely accepted in other research works (for example [**15**]), because it can help to identify context data from other types of data.

Dey et al. define the context-aware system as follows [**3**]:

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

To specify the definition above, context aware system may be treated as a functional block which performs a defined set of actions in order to produce expected results on the basis of input context data. It was pointed out in the literature that the system may also perform security actions using contextual information. Context aware security is the type of security where the additional context data is taken into account while performing security tasks (for example securing communication or access control). Hu et al. introduce context-awareness in securing medical records [**16**]. Al-Muhtadi et al. propose the context-aware mechanism for assuring authentication and access control [**17**]. Covington et al. propose the system architecture that provides more flexible, context-aware security services such as authentication, access control and adaptable security subsystem. [**18**]. Kotulski et al. propose security aware framework, which allows systems to adapt security measures in context-changing environments [**19**].

In other research Buchholz et al. introduced the Quality of Context (QoC) definition as follows:

Quality of Context (QoC) is any information that describes the quality of information that is used as context information. Thus, QoC refers to information and not to the process nor the hardware component that possibly provide the information. [**20**].

There are defined five QoC components: precision, probability of correctness, trustworthiness, resolution, up-to-dateness. There is research conducted on defining each QoC component, including trustworthiness. For example Manzoor et al. describe in [**21**] the formula for calculating trustworthiness, but it takes into account only distance from the data source and accuracy, which cannot be used for assuring full security for the system. Moreover, only one value is produced, which gives not much information about full trustworthiness of the received piece of data. There is need to introduce more detailed trustworthiness description, for example in the form of the vector of values. This idea is developed in the further Sections.

2.2 Trust and Reputation

The trust and reputation terms are being used in environments, where many entities interact with each other in order to achieve common benefits. Peer-to-peer networks are a good example of such environment. Like for context-aware computing, there are also various definitions of trust and reputation in the literature. Reputation is defined as:

- *ópinion or view of one about something"* Sabater and Sierra [**22**],
- *"perception that an Agent creates through past actions about its intentions and norms"* Mui et al. [**23**],
- *"[it] helps us to manage the complexity of social life by singling out trustworthy people in whose interest it is to meet promises"* Miztal [**24**],
- *ąn expectation about an agent's behaviour based on information about or its past behaviour"* Abdul-Rahman et al. [**25**].

For purpose of this document the reputation best term corresponds to the last definition (Abdul-Rahman et al.). Trust is considered to be:

- *ąn Agent's belief in another Agent's capabilities, honesty and reliability based on its own direct experiences"* Wang et al. [**26**],
- *"the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context"* Grandison et al. [**27**],
- *ą particular level of the subjective probability with which an agent will perform a particular action, both before we can monitor such action and in a context in which it affects our own action."* Gambetta [**28**].

For purpose of this document trust best description corresponds to the last definition (Gambetta). Reputation may be connected with the entity that agent wants to interact with and is determined by that entity's past actions. Trust is the way that agent sees

the entity with given reputation. Trust and reputation are strongly connected with each other and in literature their definitions are sometimes very similar.

In the next Sections the reputation term will be used to determine how reliable the data source is and the trust term will be used in the context of received data reliability, considering various aspects.

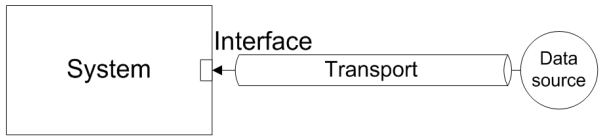
2.3 Trust Vector

During the trust-related research there was pointed out that vector of trust values may be more elastic and give better results in trust-aware decision making than single value. Wang et al. [**29**] point out that apart from single trust value there should be calculated service trust trend and performance consistency level to determine if entity has currently rising or falling trust tendency. Similarly, Li et al. [**30**] suggest the usage of three-value vector instead of single value to provide precise trust management system. The values are as follows: final trust level, service trust level and service performance consistency level. The presented approach has an application in e-commerce and e-service environments. Zhang et al. [**31**] present the trust vector approach in transaction services such as auctions. Vector is used instead of a single value to deal with transaction amount imbalance and item imbalance problems, with the usage of transaction context. Ray et al. [**32**] introduce the trust vector that consists of three components: experience, knowledge and recommendation.

2.4 Motivation

Context-aware security is a part of the context-aware computing. It requires high quality context data to adapt security mechanisms according to current context situation. Thus processed data should have certain quality, determined by the Quality of Context metrics. Various categories of context data may be delivered by different, independent providers, so there is need to determine if the provided data is safe enough to be used for security decisions. Reliability attribute of QoC idea is able to provide such information. If a piece of data is reliable enough, it can be used for further processing. Reliability can be presented either with a single value or a vector of values. As it was pointed out the vector is considered more flexible than a single value. If the decision making module receives information about single value of reliability, it is not aware about various aspects of data delivery process that may influence the reliability and be important for certain decision process. For example, data source and security services used may be important for decision process X, but process Y would require only information about data source. Vector of delivery characteristic values is universal representation of context data reliability for various security decision processes. While dealing with distributed data sources connected with different providers, there should be a mechanism able to determine each data source reliability. For that purpose reputation system may be used.

The reliability vector presented in this paper is a new method to deal with the Quality of Context reliability parameter, which takes into account the aspects of data delivery



Rysunek 1. Data delivery to the system

process, such as interface, overall data quality, data source and security services used during delivery.

3 Reliability Vector Approach

For the purpose of this document the context data reliability is defined as follows:

Context Data Reliability: *context data attribute which determines how much we can trust obtained data to be transferred correctly, not altered by unauthorized entity and include desired information.*

3.1 External Context Data - Reliability Vector

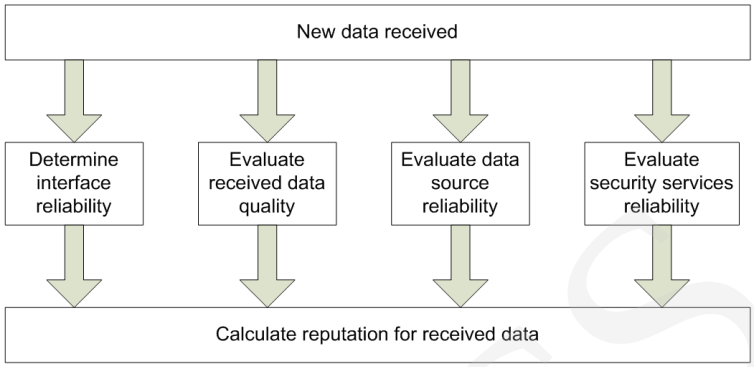
Some data received by the context aware system may come from internal components of the system, but other data may be obtained from the external environment using defined interfaces. In this paper there is considered the second method of data acquisition - external sources. Fig. 1 shows the way the system receives data. Firstly, data is produced in the data source, then transported through a specified channel optionally using security services and received by the system with one of the interfaces. Thus external data reliability depends on the source reputation, data quality and the way it came to the system (the interface and transport method).

Fig. 2 presents the actions that data reliability mechanism has to perform in order to produce a final vector of reliability values. Instead of producing one reliability value the vector of values is determined. It allows context-driven mechanisms to choose certain components of the vector in further context data reliability-aware processing.

When new data is received with one of the interfaces, there are determined four values:

- Interface reliability value,
- Data quality value,
- Data source reputation value,
- Security services dependent value.

Next, these four values make input for the final data reliability calculation block. In the next sections each block is described.



Rysunek 2. Actions performed to produce data reliability vector

3.2 Interfaces Reliability

Every interface has its own characteristics. Table 1 presents the sample interface categorization which helps to assign reliability values to pieces of data received through each interface of the context security aware system.

To calculate reliability of received data there is need to estimate interface reliability first. This estimation is needed while taking into account the way data was delivered to the system. Because of a wide variety of interfaces this is an important criterion.

To calculate interface reliability there should be used the reliability function I_r. It assigns each interface a risk value, which is connected with interface categorization, individual characteristics of the interface and past experience with that interface. We can define the reliability function as follows:

$$I_r = I_{cat} \cdot I_{char} \cdot I_{exp} \tag{1}$$

where I_{cat} is the interface categorization risk value, I_{char} is the interface individual characteristic risk value, I_{exp} is the interface experience risk value.

Categorization risk has discrete values and the range depends on the values assigned for each category. These values should be defined arbitrarily during test phase of the reliability mechanism. Individual risk and experience risk values may be continuous or discrete. Their range should be defined during implementation phase of the reliability mechanism. All risk values should be positive. The range for the final interface reliability value is between I_{r-min} and I_{r-max}.

3.3 Data Quality Evaluation

The second step to define data reliability is data quality evaluation. It takes into account what kind of data systems expected to receive and what it really received. To evaluate this value the following aspects should be taken into account:

Tablica 1. Interfaces categorization

Interface category	**Description**
Human multimedia interface	Delivers data connected with user behaviour, obtained by multimedia devices such as microphone or camera. This kind of interface may deliver data with high error rate considering user's real intentions and behaviour.
Human data interface	Delivers data input from human source entered with low error rate device such as keyboard or touchscreen.
Device wireless interface	Delivers data sent with wireless technology using atmos-phere as the medium. Transmission may be detected or intercepted by every entity in the range of transmitter.
Local wired device interface	Delivers data with wired connection (for example Ethernet cable). All communication is being made through local network that belongs to one administrator. The possibility of external eavesdropping is minimal.
Internet device interface	Delivers data through public network (for example the Internet). Possibility of data interception, manipulation or loss is relatively big.
Plugged device interface	Delivers data from the device plugged temporarily to the system with defined standard (for example USB, serial connection, optical drive). There is significant uncertainty for data reliability because of a lack of data carriers history and the fact that carriers are portable.
Sensor interface	Delivers data that come from various sensors (for example fingerprint reader, retinal scanner, temperature and pressure sensors). This data may be delivered in a raw form, without much processing. It is assumed that sensors are directly connected to the system.

- Contradictions in the received data set (for example checksums validity, internal conformity),
- Compliance with the request and the received data set concerning each request condition and its equivalent in response (if there was data request),
- Quality comparison with the previously received data under the same conditions as the currently received data set,

- Other typical aspects important for the system, which is being developed.

Data quality evaluation function should receive all above listed pieces of information as its parameters as follows:

$$D_q = f_q(p_1, p_2, \cdots, p_n) \tag{2}$$

where f_q is data quality evaluation function and $p1, p2, \cdots, pn$ are function parameters (received data characteristics). Function f_q should give quantified or continuous values from the range, between f_{q-min} and f_{q-max}.

3.4 Data Source Reliability Evaluation

The third step to calculate data reliability is data source reliability evaluation. The system can receive data from various sources. Some of them may be more trustworthy than others, so there is need to assign each data source a reliability value. Simple reputation mechanism may be used for that purpose. On the basis of the previous experience and opinion from the external sources (for example other systems using the same data source) the system may calculate final data source reliability as follows:

$$S_r = f_r(S_{exp}, S_{rep}) \tag{3}$$

where f_r is data source reliability evaluation function, S_{exp} is data source aggregated internal experience value, S_{rep} is data source aggregated external reputation value. Like functions in the previous sections, f_r should give quantified or continuous values from the range, between f_{r-min} and f_{r-max}.

3.5 Security Level Evaluation

The forth step is security level evaluation of the received data set, based on digital security services used to protect data during transmission. Each used security service increases total value of security level component. The sample security services are as follows (based on [**33**]): accountability, integrity, confidentiality, availability, privacy, non-repudiation, auditability, authenticity and trustworthiness. The list is not closed and may contain other security services characteristic of certain delivery methods.

Each security service should be given importance value, that defines how important it is it for the system and how security is provided. The security level evaluation may be performed as follows:

$$S_l = f_s(s_1, s_2, \cdots, s_n) \tag{4}$$

where f_s is security level evaluation function and $s1, s2, \cdots, sn$ are function parameters (each security service characteristic of the received data set). The function f_s should give quantified or continuous values from the range, between f_{s-min} and f_{s-max}.

Tablica 2. Interface categorization

Interface name	**Category**	I_{cat} **value**	I_{char} **value**	I_{exp} **value**
USB port	Plugged device interface	0.8	0.7	1
Webservice	Internet device interface	0.5	0.9	1
GUI	Human data interface	1	1	1

3.6 Reliability Assessment

Once all four values are calculated (interface reliability, data quality, data source reliability, security level) the final reliability vector can be constructed as follows:

$$R_V = [I_r, D_q, S_r, S_l] \tag{5}$$

If there is need to produce only one reliability value for the received data set, it can be assessed as follows:

$$R = a_i \cdot I_r + a_d \cdot D_q + a_{sr} \cdot S_r + a_{sl} \cdot S_l \tag{6}$$

where a_i, a_d, a_{sr}, a_{sl} are importance coefficient for each component. It is important to balance all input values so that no value would cover the others. Importance coefficients allow to control significance of each of four values: I_r, D_q, S_r and S_l.

4 Use Case

To illustrate how context data reliability calculation may be implemented let us suppose that we have a context aware system with three interfaces - USB port, Ethernet connection to the Internet and graphical user interface (GUI) for local access to the system. Ethernet connection is properly firewalled so that it is possible to transfer data only with defined webservice which is served by the web server. At first we have to define interface categories and their risk value. Table 2 shows the sample categories mapping. It is assumed that the values come from test phase of the mechanism and were defined arbitrarily by experts estimation to reach system needs best.

Secondly, we have to define data quality evaluation function. We define the following function parameters:

- p_1 - checksum validity, 1 if valid, 0 otherwise,
- p_2 - compliance with request,

$$p_2 = \frac{received\ parameters\ number\ (corresponding\ to\ the\ requested\ ones)}{number\ of\ requested\ parameters} \tag{7}$$

- p_3 - comparison with historical responses,

$$p_3 = \frac{response\ number\ with\ the\ same\ format\ and\ details\ as\ received\ data}{number\ of\ all\ historical\ responses\ to\ the\ same\ request} \tag{8}$$

Tablica 3. Importance values for security services

Service name	**Importance value**
Integrity	0.6
Confidentiality	0.4

If there are no historical responses to the same request, the p3 parameter is equal to 1. The data quality function has the following formula:

$$D_q = f_q = p_1 \cdot p_2 \cdot p_3 \tag{9}$$

Next we need to define the data source reliability evaluation function. To do that we need aggregated internal and external experience values defined as follows:

$$S_{exp} = 1 - \frac{number\ of\ previously\ detected\ data\ set\ inconsistency}{number\ of\ all\ previously\ received\ data\ sets} \tag{10}$$

$$S_{rep} = \frac{number\ of\ votes\ confirming\ trustworthiness\ of\ data\ source}{number\ of\ all\ votes} \tag{11}$$

S_{rep} reputation value is calculated by periodical voting conducted by a group of systems using given data source. If there are no previously received data sets, then S_{exp} is equal to 1. In the voting procedure there will be at least one vote because at least one system is always present in the secure context aware network. With no system present, there would be no network. If system has no experience with the given source, it assumes its trustworthiness and confirms it.

The data source reliability function has the following formula:

$$S_r = f_r = S_{exp} \cdot S_{rep} \tag{12}$$

Next, we have to define the security level evaluation function. To simplify the example we assume that the importance value is fixed for each security service and we take into account only integrity and confidentiality. The sample values are listed in Table 3. Like with the interface categorization, it is assumed that the values come from the test phase of the mechanism and were defined arbitrarily by experts estimation to reach system needs best.

We can define the security level evaluation function as follows:

$$S_l = f_s = s_1 + s_2 \tag{13}$$

where:

- s_1 - integrity coefficient, if the received data set is protected with the integrity function the parameter is equal to 0.6 (according to Table 3), otherwise it is equal to 0,

- s_2 - confidentiality coefficient, if the received data set is protected with the confidentiality function the parameter is equal to 0.4 (according to Table 3), otherwise is equal to 0.

Once we have all components for reliability assessment, we can define the final vector formula:

$$R_V = [I_r, D_q, S_r, S_l] \tag{14}$$

and aggregated reliability value:

$$R = I_r + D_q + S_r + S_l \tag{15}$$

We assume that all four component values are equally important so $a_i = 1$, $a_d = 1$, $a_{sr} = 1$, $a_{sl} = 1$.

Now suppose that the system, configured as it was described above, receives the context data set with the webservice interface as a response to the geographical coordinates request (longitude, latitude, height). The response has valid integrity checksum but contains only two coordinates - longitude and latitude. There is no encryption, thus no confidentiality function. The data set comes from the trusted vendor with no bad history (no previous data inconsistency detected among 1000 received data sets). During the last voting session, there were 10 other systems using the data delivered by the same vendor and 8 of them confirmed the source trustworthiness. Coordinates in the received data set are formatted according to the specification, in the same format they are usually delivered (the same data quality as in previously received data). We can calculate the context data reliability parameters as follows:

$$I_r = I_{cat} \cdot I_{char} \cdot I_{exp} = 0.5 \cdot 0.9 \cdot 1 = 0.45 \tag{16}$$

$$D_q = f_q = p_1 \cdot p_2 \cdot p_3 = 1 \cdot \frac{2}{3} \cdot 1 \cong 0.67 \tag{17}$$

$$S_r = f_r = S_{exp} \cdot S_{rep} = (1 - \frac{0}{1000}) \cdot \frac{8}{10} = 0.8 \tag{18}$$

$$S_l = f_s = s_1 + s_2 = 0.6 + 0 = 0.6 \tag{19}$$

The reliability vector for the described case is:

$$R_V = [I_r, D_q, S_r, S_l] = [0.45; 0.67; 0.8; 0.6] \tag{20}$$

The aggregated reliability value is:

$$R = I_r + D_q + S_r + S_l = 0.45 + 0.67 + 0.8 + 0.6 = 2.52 \tag{21}$$

The calculated reliability vector determines each of four reliability aspects of received dataset and can be stored in the database for further usage. Various reliability-aware processes (including security services) may make use of this vector and take into account any number of available reliability aspects. If only aggregated value R was stored, some of the processes might give results with higher uncertainty because R aggregates all four reliability aspects. All processes would have to use the same reliability value,

irrespective of reliability aspects needed. This is an example of vector advantage over a single value. The uncertainty must be lowered to the minimum while dealing with security services, so the reliability vector is better in this case.

5 Summary and Future Work

This paper proposes the mechanism for context data reliability vector assessment in the context aware systems. The source reputation is taken into account as well as the way data is being transferred from the source to the system. The calculated reliability vector can be used for further context reasoning and bad-quality pieces of data detection in order to prevent them from affecting the system decision mechanisms. The vector form of the reliability allows other data processing mechanisms to choose what aspects of data have to be taken into account. Not all vector values have to be used during the context data based decision making. Introducing the aggregated reliability value may help mechanisms that require only one reliability value.

Many ubiquitous environments and systems serving mobile devices make context based decisions as the part of their functionality. The proposed context data reliability assessment can increase the protection level of such systems and environments by determining unreliable pieces of data. Such mechanism can be the part of context management framework for secure network services [**19**] and can provide reliability information for context-aware security services. Once the unreliable pieces of data are excluded from processing in the security adaptation phase, the risk of inappropriate security measures application is lower than without the reliability mechanism.

The described reliability vector is the beginning of studies on context data reliability and further work is planned as follows:

- The method of defining functions for calculating each vector component,
- Implementation of data reliability mechanism using publicly available context sources,
- Reputation system optimization and efficiency analysis,
- Research on the most precise and efficient way to define interface categories values, security services importance values, data quality evaluation procedures,
- The evaluation of proposed model will be performed during further simulations,
- Research on the decision making mechanism using the calculated reliability vector.

Literatura

[1] Want R., Hopper A., Falcao V., Gibbons J., The active badge location system, ACM Transactions on Information Systems (TOIS) (1992): 91–102.

[2] Schilit B. N., Theimer M. M., Disseminating active map information to mobile hosts, Network, IEEE, 8(5) (1994): 22–32.

[3] Abowd G. D., Dey A. K., Brown P. J., Davies N., Smith M., Steggles P., Towards a Better Understanding of Context and Context-Awareness, Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing (1999): 304–307.

[4] Hull R., Neaves P., Bedford-Roberts J., Towards situated computing,"Wearable Computers, 1997, Digest of Papers., First International Symposium (1997): 146–153.

[5] Cooperstock J. R., Tanikoshi K., Beirne G., Narine T., Buxton W. A., Evolution of a reactive environment, Proceedings of the SIGCHI conference on Human factors in computing systems (1995): 170–177.

[6] Brown M. G., Supporting user mobility, Mobile Communications, Springer US (1996): 69–77.

[7] Rekimoto J., Ayatsuka Y., Hayashi K., Augment-able reality: Situated communication through physical and digital spaces, Wearable Computers, Digest of Papers, Second International Symposium (1998): 68–75.

[8] Elrod S., Hall G., Costanza R., Dixon M., Des Riviéres J., Responsive Office Environments, Communications of the ACM, 36(7) (1993): 84–85.

[9] Fickas S., Kortuem G., Segall Z., Software organization for dynamic and adaptable wearable systems, Wearable Computers, Digest of Papers, First International Symposium (1997): 56–63.

[10] Brown P. J., Bovey J. D., Chen X., Context-aware applications: from the laboratory to the marketplace, Personal Communications, IEEE, 4(5) (1997): 58–64.

[11] Ryan N. S., Pascoe J., Morse D. R., Enhanced reality fieldwork: the context-aware archaeological assistant, Computer applications in archaeology (1998).

[12] Chen G., Kotz D., A survey of context-aware mobile computing research, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (2000).

[13] Schmidt A., Aidoo K. A., Takaluoma A., Tuomela U., Van Laerhoven K., Van de Velde W, Advanced interaction in context, Handheld and ubiquitous computing, Springer Berlin Heidelberg (1999): 89–101.

[14] Dey A. K., Abowd G. D., Salber D., A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Human-computer interaction, 16(2) (2001): 97–166.

[15] Perera C., Zaslavsky A., Christen P., Georgakopoulos D., Context Aware Computing for The Internet of Things: A Survey (2013).

[16] Hu J., Weaver A. C., A dynamic, context-aware security infrastructure for distributed healthcare applications, Proceedings of the first workshop on pervasive privacy security, privacy, and trust (2004).

[17] Al-Muhtadi J., Ranganathan A., Campbell R., Mickunas M. D., Cerberus: a context-aware security scheme for smart spaces, Pervasive Computing and Communications, 2003.(PerCom 2003), Proceedings of the First IEEE International Conference (2003): 489–496.

[18] Covington M. J., Fogla P., Zhan Z., Ahamad M., A context-aware security architecture for emerging applications, Computer Security Applications Conference, Proceedings (2002): 249–258.

[19] Kotulski Z., Sepczuk M., Sitek A., Tunia M.A., Adaptable context management framework for secure network services, (to appear) (2014).

[20] Büchholz T., Küpper A., Schiffers M., Quality of Context: What It Is And Why We Need It, Proceedings of the 10th International Workshop of the HP OpenView University Association(HPOVUA) (2003).

[21] Manzoor A., Truong H. L., Dustdar S., On the Evaluation of Quality of Context, Proceedings of the 3rd European Conference on Smart Sensing and Context (2008): 140–153.

[22] Sabater J., Sierra C., Regret: A reputation model for gregarious societies, Fourth workshop on deception fraud and trust in agent societies, 70 (2001).

[23] Mui L., Mohtashemi M., Halberstadt A., A computational model of trust and reputation, System Sciences, 2002, HICSS, Proceedings of the 35th Annual Hawaii International Conference (2002): 2431–2439.

[24] Mitzal B., Trust in modern societies, Polity, Cambridge (1996).

[25] Abdul-Rahman A., Hailes S., Supporting trust in virtual communities, System Sciences, 2000, Proceedings of the 33rd Annual Hawaii International Conference (2000):4–7.

[26] Wang Y., Vassileva J., Trust and reputation model in peer-to-peer networks, Peer-to-Peer Computing, Proceedings, Third International Conference (2003): 150–157.

[27] Grandison T., Sloman M., A survey of trust in internet applications, Communications Surveys & Tutorials (2000): 2–16.

[28] Gambetta D., Can we trust trust?, Trust: Making and breaking cooperative relations (2000): 213–237.

[29] Wang Y., Li L., Two-dimensional trust rating aggregations in service-oriented applications, Services Computing, IEEE Transactions 4(4) (2011): 257–271.

[30] Li L., Wang Y., A trust vector approach to service-oriented applications, Web Services, 2008, ICWS'08, IEEE International Conference (2008): 270–277.

[31] Zhang H., Wang Y., Zhang X., A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments, Service-Oriented Computing and Applications (SOCA), 2012 5th IEEE International Conference (2012): 1–8.

[32] Ray I., Chakraborty S., A vector model of trust for developing trustworthy systems, Computer Security–ESORICS (2004): 260–275.

[33] Cherdantseva Y., Hilton J., A Reference Model of Information Assurance & Security, Availability, Reliability and Security (ARES), 2013 Eighth International Conference (2013): 546–555.

[34] Bellavista P., Corradi A., Fanelli M., Foschini L., A Survey of Context Data Distribution for Mobile Ubiquitous Systems, ACM Computing Surveys (CSUR), 44(4) (2012).