Incorporating Trust into Context-Aware Services

Shishkov, Boris; Fill, Hans Georg; Ivanova, Krassimira; van Sinderen, Marten; Verbraeck, Alexander

DOI
10.1007/978-3-031-36757-1_6

Publication date
2023

Document Version
Final published version

Published in
Business Modeling and Software Design - 13th International Symposium, BMSD 2023, Proceedings

Incorporating Trust into Context-Aware Services

Boris Shishkov[1,2,3(✉)], Hans-Georg Fill[4], Krassimira Ivanova[1], Marten van Sinderen[5], and Alexander Verbraeck[6]

[1] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
b.b.shishkov@iicrest.org, kivanova@math.bas.bg
[2] Faculty of Information Sciences, University of Library Studies and Information Technologies, Sofia, Bulgaria
[3] Institute IICREST, Sofia, Bulgaria
[4] Digitalization and Information Systems Group, University of Fribourg, Fribourg, Switzerland
hans-georg.fill@unifr.ch
[5] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands
m.j.vansinderen@utwente.nl
[6] Faculty of Technology, Policy, and Management, Delft University of Technology, Delft, The Netherlands
a.verbraeck@tudelft.nl

Abstract. Enabling technologies concerning hardware, networking, and sensing have inspired the development of context-aware IT services. These adapt to the situation of the user, such that service provisioning is specific to his/her corresponding needs. We have seen successful applications of context-aware services in healthcare, well-being, and smart homes. It is, however, always a question what level of trust the users can place in the fulfillment of their needs by a certain IT-service. Trust has two major variants: policy-based, where a reputed institution provides guarantees about the service, and reputation-based, where other users of the service provide insight into the level of fulfillment of user needs. Services that are accessible to a small and known set of users typically use policy-based trust only. Services that have a wide community of users can use reputation-based trust, policy-based trust, or a combination. For both types of trust, however, context awareness poses a problem. Policy-based trust works within certain boundaries, outside of which no guarantees can be given about satisfying the user needs, and context awareness can push a service out of these boundaries. For reputation-based trust, the fact that users in a certain context were adequately served, does not mean that the same would happen when the service adapts to another user's needs. In this paper we consider the incorporation of trust into context-aware services, by proposing an ontological conceptualization for user-system trust. Analyzing service usage data for context parameters combined with the ability to fulfill user needs can help in eliciting components for the ontology.

Keywords: Context awareness · Trust · Data analytics

B. Shishkov (Ed.): BMSD 2023, LNBIP 483, pp. 92–109, 2023.
https://doi.org/10.1007/978-3-031-36757-1_6

1 Introduction

Many service-based IT systems interact with the user in a pre-defined manner to execute their tasks [1–3]. This is a reflection of a fixed set of corresponding user needs that are hence considered "static", at least for the particular servicing time frame [4]. Nevertheless, user needs may be highly dynamic and are often evolving over time. ***Context awareness*** improves IT services, by the development of *systems that adapt their servicing to the situation and/or needs of the user* [5–9] – this conceptual vision has been conceived in the 1990s [16, 17]. Still, it took one decade since then for enabling technologies to become available, namely developments in *hardware*, such as device miniaturization combined with low power consumption, in *networking*, such as high-bandwidth wireless communication and positioning-related capabilities, and in *sensing*, such as miniaturized sensors for many different phenomena and the availability of sensor networks [2, 10].

When using a service, the user wants to have a certain *guarantee* in advance that the service "will offer what it promises". Said otherwise, the service is expected to be able to *satisfy the user needs in the relevant contexts.* We can call this the ***trust*** *of the user in the service and/or in the system.* Two concepts of trust exist: policy-based, where a reputable organization provides guarantees about the fitness-for-purpose of the service, and reputation-based, where other users share their experiences with the use of the service with new users [11, 12]. *Policy-based trust* and *reputation-based trust* are used in different types of environments. Services that are accessible only to a small and defined set of users, such as back-office services and services for critical infrastructures, typically use *policy-based trust* only. *Reputation-based trust* would not make sense here, since the group of users is small, and users usually do not have a choice whether to use the service or not. Services that are open to a large and more heterogeneous group of users, such as commercial services offered through Web platforms, can make use of *reputation-based trust*, usually combined with some form of *policy-based trust*. Examples why *policy-based trust* is still needed when *reputation-based trust* is present, are issues with faking the reputation scores, e.g., by buying clicks, showing fake reviews, or manipulating reputation scores shown to the users. Hence, a good reputation score combined with trust in the organization offering the score, established through *policy-based trust*, helps in addressing these issues.

Nevertheless, a problem emerges when ***combining either of the two types of trust with context awareness*** since neither the *policy-based*, nor the *reputation-based trust* concept can give the guarantee of fulfilling the user needs anymore. With servicing adapting to the user *context*, applying *policy-based trust* may be challenging because the "envelope" of that *context* would often appear to be unknown or ill-defined. Hence, the *context* could go out-of-bounds to address a contextual situation that was not foreseen at design time. For *reputation-based trust*, the fact that the *reputation* was excellent in *context A* of using a service does not mean that when the service adapts to *context B*, it would also be excellent. Thus, we argue that trust, in a sense, assumes a constant and stable service offering, whereas context-aware services can adapt to the context, thereby "breaching" the assumptions on which that trust was based.

Policy-based trust is about restricting access and confining usability, because of its assumptions for rigorous designs that constrain the service within the boundaries of

what was specified beforehand. Hence, this may substantially hamper the use of *context-aware* servicing principles and a question to answer is: ***How can we allow for context awareness in services governed by a policy-based trust principle***?

Further, for systems governed by *reputation-based trust*, *context awareness* causes two types of "surprises": (i) The wider use of a service as a result of *context awareness* may lead to situations that have not been anticipated at design time; (ii) The broader access to services would reduce *trust* because it is often unclear in what *context* the existing *reputation* score has been obtained. Users would become dissatisfied if the service does not satisfy their need in their specific *context* in spite of the fact that the *reputation* scores (based on other *contexts*) suggest otherwise. This leads to a second question to answer: ***How can reputation-based trust be implemented in a context-aware service***?

The first research question takes the *policy-based trust* as a given and looks at solutions in terms of how to implement *context awareness* in a more rigorously governed service system. In contrast, the second research question takes *context awareness* as a given and looks at what strategies for *reputation-based trust* would be effective for *context-aware* systems. The solution direction proposed in this paper concerns an ***ontological conceptualization*** that carefully defines elements of *context* and elements of *trust*, using the same ontological base, allowing for reasoning across the involved technical areas. We claim that this conceptualization could work to address the issues in both research questions, since they both address the integration of *context awareness* and the two dominant *trust* models.

This way of modeling can be assisted by *data analytics* concerning the service usage – based on historic data, user entities can be clustered and *context* situations can be predicted, as well as *trust*-related attitudes and the user perception of service quality. Service performance indicators can help in making such predictions [33]. *Machine learning* [13] (for example: *Bayesian modeling*) and *covering/clustering algorithms* can then partition the *context-aware* usage space into sub-spaces for which different *trust* levels would apply, and provide suggestions for boundaries for the *context* parameters, outside of which the service should not be used when a minimum *trust* level should be attained. Of course, one should be careful with fully automating these predictions, as a future situation might differ significantly from those described by historic data.

Note that in addition to *user-system* trust (the user's trust in the system), three other forms of trust exist: *system-user* trust (the system's trust in the user), *user-user* trust (trust that users of a system have in each other), and *system-system* trust (trust of systems in other systems on which they are dependent). In this work, we just focus on user-system (or user-service) trust.

The remainder of the current paper is structured as follows: Sect. 2 presents a conceptual model of context awareness, applying a functional perspective and taking into consideration related work. Section 3 provides rigorous definitions of trust concepts and the dimensions of trust strategies. In Sect. 4 we present our proposed ontology-driven conceptualization, partially justified by an example as well as by a discussion addressing some benefits and limitations of our propfosal (Sect. 5). We conclude the paper in Sect. 6.

2 Context Awareness

Among what has inspired us in considering *context awareness* are works and discussions of Albrecht Schmidt, such as [15] and our previous work, such as [3]. What determines the notion of "*context awareness*"? In our view, this is innate with regard to our smart human behavior, for example: a person would navigate his/her way around without being familiar with the place; or: a teacher would switch his/her phone to silent mode when in class. In contrast, any machine or computer device is "blind" for the *context*, for example: a mobile phone would ring whether or not the owner is busy; or: a laptop may be forced to restart no matter if this is convenient for the user or not. Hence, human beings are *context-aware* by nature and one would not even notice this, while to date many computer systems do not have such capabilities. For this reason, it is necessary that we DESIGN computer systems in such a way that they are capable of *perceiving the real world and acting upon what they interpret from it*. We have inspiring examples in this direction from the years since the new millennium: (i) The navigation system of a smart phone is an example of *context awareness* since the GPS-receiver of the phone allows for its "knowing" where it is and guiding the user; (ii) Related to the previous example concerning a smart phone: a driver could be diverted to avoid a "sensed" traffic jam, counting on the phone's location data and broader *context* that is captured (and used); (iii) There are house lighting systems, counting on sensors for establishing whether it is dark *and* somebody is moving (i.e., present) in the house. Of course, one can go from a "*context-aware*" mode to an "*explicit use*" mode - for example: one would fix the lighting to "on" if there are maintenance works in the building. Hence, *context awareness* is about making the usage of technology easier, by *freeing users from doing things that the system can do as well*. The above examples show that some useful realizations of this are currently present. Nevertheless, they stem from ideas that point back to the early 1990s, when the inspiring scientist and visioner Mark Weiser stated (in his '91 essay entitled "The Computer for the 21st Century", further reflected in [16]) that "specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence". This has paved the way to a discipline, labelled "ubiquitous computing" that in turn pushed towards what is currently labelled as "*context-aware computing*" or "*context awareness*", as explicitly used by Bill Schilit already in 1995 [17]. All those concepts have been carefully addressed by relevant scientists in 2009 [18] where Anind Dey summarized what was widely agreed upon by then [19]:

- Since situational information, such as facial expressions, emotions, past and future events, the existence of others around, and relationships to them are crucial for humans to understand what is occurring, it is necessary to improve the "language" that humans can use to interact with computers.
- It is also necessary to increase the amount of situational information, or *context*, that is made available to computers.
- *Context* is defined as: any information that can be used to characterize the situation of an entity.

- 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.

Considering this, we have introduced *three categories* of *context-aware* systems, with regard to adaptive service delivery [14], where the following adaptation perspectives are possible: serving *user needs*; *system needs*; and *public values.*

We argue that the abovementioned scientists (namely: Weiser, Schilit, Dey, and Schmidt) are the pioneers in the area of *context-aware computing*. In addition, other relevant works (authored by them and other scientists) have helped to further improve our understanding of the notion of *context* and to make serious progress in the development of *context-aware applications* [1, 5, 20, 21]. Finally, we have considered relevant R&D *context awareness* projects, such as CyberDesk [22], AWARENESS [2, 23], and SECAS [24] to get further insight. Our observation is that most projects follow bottom-up (*technology-driven*) developments (as opposed to *user-centric* developments); we consider this a serious obstacle with regard to adequately conceptualizing *context awareness.*

Other relevant literature contains for instance the useful survey of Alegre et al. [7] that is mainly focused on the development of *context-aware applications* as well as on the consideration of *public values*. The same holds for the works of Alférez and Pelechano [8] – they consider the dynamic evolution of *context-aware systems*, the development itself, and the relation to *web services*. The latter holds also for the *service-orientation perspective* as proposed by Abeywickrama [9]. In line with the abovementioned observation, all these works take a primarily *technology-driven perspective* and are less concerned with the *user perspective*. The same holds for other works touching upon the *adaptive delivery of services*, always considered in a bottom-up perspective, featuring *decision-making* [25], *safety* of stakeholders [26], and *routing* [27]. The technology-driven perspective is also visible in the *systematic literature review* in the doctoral thesis of Van Engelenburg [28].

We therefore conclude the following: As it concerns the *conceptual perspective*, not much has been added after Mark Weiser - 1991. As it concerns the *1991–2023 developments*, they mainly concern *enabling technologies* and their successful relevant implementations. We see ***room for improvement concerning the user perspective*** and the ***alignment between context awareness and data analytics***, for the sake of providing ***new ways of context gathering*** that also concerns a possible *prediction of context situations and/or user preferences/attitudes.*

As already mentioned, *users* often have *needs evolving over time* that relate to corresponding *context situations. Context-aware systems* are expected to be providing ***context-specific services to users in accordance with their context-dependent needs***. When delivering services, the *system* would interact with the *context*. Hence, not only *collecting data on the context* is important but also *delivering a service* that *matches* the *context*. The fact that the *service* is delivered to a *user* means that the user is part of the *context*; *context-aware* service delivery concerns the connection between what the *context* is and what a *user* needs. Hence, considering the above from a functional perspective gives two key processes that often go one after another, namely ***situation determination*** and ***behavior adaptation***, as suggested by Fig. 1.

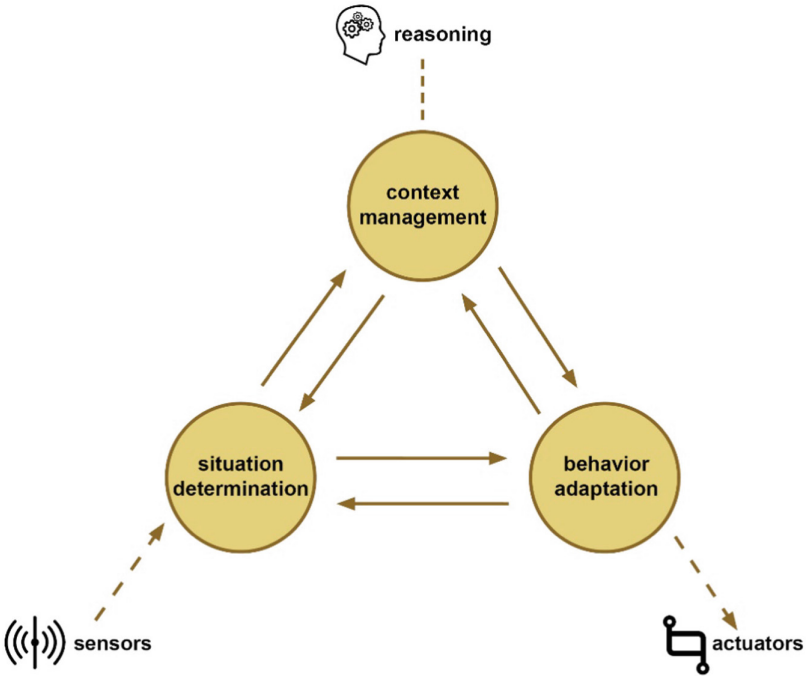

Fig. 1. Context awareness – a functional perspective

As shown in the figure: (a) *Situation determination* is often supported by *sensors*; we assume this in the current paper, acknowledging nevertheless that there may be also other ways of determining the (user) situation, for example: supported by *data analytics* and *predictive modeling* [13, 33, 34]. Hence, the (user) situation is determined using incoming sensor data, by *inferring higher-level context information*. (b) *Behavior adaptation* is needed such that *service delivery* is *aligned* with the *context situation* and corresponding (*user*) *needs*. This has effect on what the system is "doing", materialized by *actuators*. (c) *Context management* is needed to align incoming (sensor) data and the corresponding system behavior adaptation. This assumes *reasoning*, as illustrated in the figure, that is two-fold: (c_1) When a situation is determined, it should be established to which corresponding (user) needs it points and when this is not straightforward (because of precision-related and/or other issues) then the "*context manager*" may "ask" for more interpretation "attempts" (that is why the arrows between *situation determination* and *context management* are in both directions). (c_2) When the actual (user) needs are established, the "*context manager*" would "ask" the system to adapt its servicing accordingly and when the *behavior adaptation* requires more and/or more precise information, then the system would ask the "*context manager*" to provide more information (that is why the arrows between *context management* and *behavior adaptation* are in both directions).

Finally, there are arrows in both directions between *behavior adaptation* and *situation determination*, to indicate that implementing a behavior adaptation may require *real-time sensor data* (for example: concerning an actual location) and sometimes refining data

featuring the (user) situation may require *information concerning actuators' operation being updated*.

As mentioned earlier in the paper, *trust-related issues* referring to *context awareness* are:

- The system could adapt beyond the boundary where it can be trusted and user-system trust before and after behavior adaptation of the system could be different.
- *Reputation* scores can have been provided for other *contexts* than the one the user is currently facing.

The next section provides an elaboration concerning *trust*.

3 Trust

When approaching the topic of *trust* in the area of *information systems*, a multitude of aspects can be considered, ranging from organizational, technical to legal aspects. In the following we will consider two main directions of *trust* that are essential pillars for many current systems without claiming exhaustiveness of all *trust* aspects [11]. The first direction is ***policy-based trust*** where access to information or services is regulated via some technical means and thus leads to *trust* by restricting the access to information to particular (groups of) users. This includes for example the use of *authentication mechanisms* such as passwords or digital signatures. The result of *policy-based trust* is the issuance of a permission to access a resource or the denial of that access. More fine-grained variants may be defined, e.g. for further detailing the types of permissions issued and also non-functional aspects – e.g. whether data is secure – could be added.

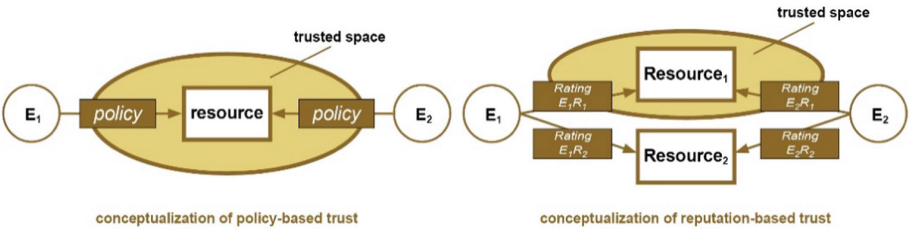

Fig. 2. Policy-based trust vs reputation-based trust

For conceptualizing this relationship, we can describe two *entities* (E_1 and E_2) that have access to a *resource* via a common *policy*. The *trusted space* is thus defined through this policy that is the same for all entities, see Fig. 2 – left.

The second direction is denoted as ***reputation-based trust***. Here, the level of *trust* into a resource is calculated based on some kind of *reputation* assigned by other entities - either of the same or a different kind. Examples include *rating systems* for resources such as websites, documents, or products - either *explicitly* (via ratings by users) or *implicitly* (via references). The result of *reputation-based trust* is thus not a binary decision but rather a gradual description of how much *trust* can be placed in some resource, see Fig. 2 - right.

This can be conceptualized as follows: Two *entities* E_1 and E_2 which access *resources* R_1 and R_2 each conduct a *rating* of each *resource*, i.e. Rating E_1R_1 indicates that entity E_1 has rated resource R_1 with some numerical value. The combination of all ratings for a resource R_i from all entities E_i then defines the *trusted space*. The combination of the ratings may either be defined *centrally*, e.g. by the provider of the resource, or in a *decentralized fashion*, i.e. by each entity.

We can further distinguish between different ***trust strategies*** [12]. In an *optimistic strategy*, it is assumed that trustful resources are the default. Only if a violation or deviation occurs, further actions are needed. In a *pessimistic strategy*, trust is restricted unless a reason is given for not doing so. The *centralized strategy* proposes to use central organizations in which trust is placed. The *investigative strategy* requires entities to conduct their own investigations for deciding about their trust in resources. In a *transitive strategy*, delegation to other entities for determining trust in a resource is assumed. Finally, for the sake of exhaustiveness, we would like to mention the extreme example where the user would distrust anything unless rules indicate trust in a resource can be granted – we refer to this as "*zero-trust strategy*" [35].

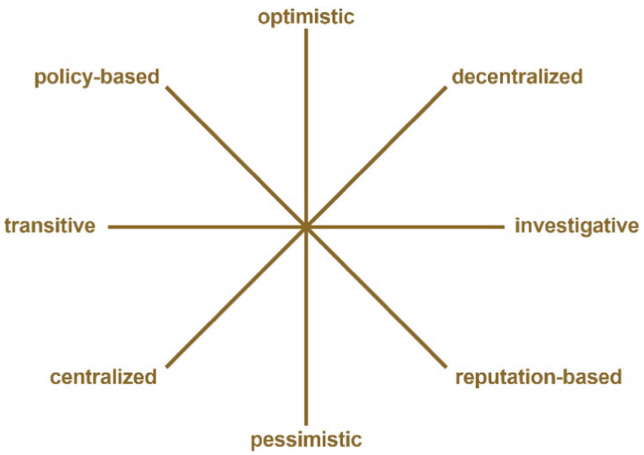

Fig. 3. An example of combining trust strategies

To briefly exemplify the above, we consider earth observations in different countries: (i) When they are governed by a state organization and concern everyday-life-related prognoses/warnings, earth observation centers would count on monolithic architectures, leaning towards {centralized, policy-based, transitive, pessimistic} trust strategies, for instance: NESDIS [40] and JMA [41]. (ii) When observations stem from agreements of independent organizations, such as universities, companies, and so on, they would usually count on federated structures, leaning towards {decentralized, policy-based, transitive} trust strategies, for instance: EPOS [42]. (iii) Finally, when open environmental data is created, counting on community-based infrastructures, such as Sensor. Community [43], one would lean towards {decentralized, reputation-based, investigative, optimistic} trust strategies.

These five strategies may be combined as different dimensions, as exemplified in Fig. 3. If we would be combining a *policy-based strategy* with the *investigative* and the *pessimistic* strategy, then: access to resources is restricted by policies and we in general assume that *trust* is only established if the result of the policy leads to a positive outcome; in addition, we employ the *investigative strategy* whereby we can inspect ourselves whether *trust* can be placed in a resource or not, thereby probably assuming a *pessimistic* outcome first. A prerequisite for the *investigative strategy* is that all necessary information is transparently available. A typical technological solution for this latter case would be *blockchains* [44].

4 Proposed Ontology-Driven Solution Directions

Starting from a general consideration of *service provisioning*, we provide in this section *conceptual views* concerning *context awareness* and *trust*. We also consider their alignment as well as possible added value of *data analytics.*

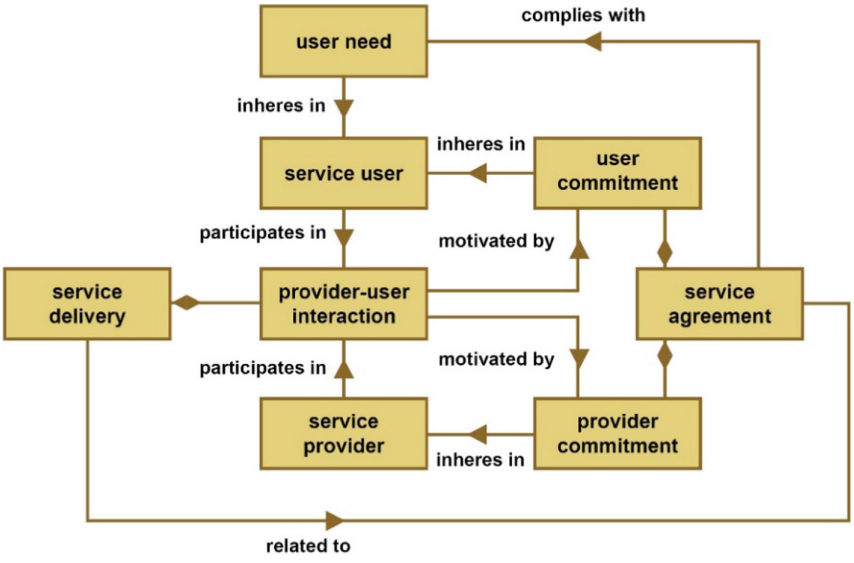

Fig. 4. Simplified service model

We are essentially focused on analyzing and/or designing enterprise information systems and in this we stick to the way of modeling suggested by Shishkov [10], which builds upon the ontological views of Dietz [31] that are in turn stemming from the systems-related views of Bunge [36]. From this perspective, we should have a **SYSTEM** under consideration, that is composed of entities interacting with other entities. Further, those entities that do not belong to the *system* but are interacting with entities of the *system* comprise the *system* **ENVIRONMENT**. Finally, instead of considering the (human) entities themselves as composition elements of a *system*/*environment*, we consider the **ROLES** in

which they appear. Otherwise, it would be confusing considering some entities who may appear in different *roles*, including nontypical ones. With regard to *service provisioning*, we argue that two essential *roles* are **SERVICE PROVIDER** and **SERVICE USER**; each of them can be fulfilled not only by a human entity but also by an IT *system*. We are particularly interested in *service provisioning* because we claim that a **SERVICE MODEL** is needed as a basis for reasoning about **CONTEXT AWARENESS** and **TRUST**. We propose such a model (a simplified one) based on previous work [29, 30], see Fig. 4.

As seen from the figure, we focus on *service delivery*; we acknowledge that for a full account of the *service* concept, *service offering* + *service negotiation* also need to be considered. *Service delivery* starts after a *service user* and a *service provider* have reached a *service agreement*, which is composed of **COMMITMENTS** (from the side of the *service provider* and from the side of the *service user*). The *service agreement* complies with the **NEEDS** of the *service user*, assuming the *service user* has agreed upon *commitments* regarding *service delivery*, where *service delivery* consists of the execution of *provider-user interactions* aimed at fulfilling the *commitments* established in the *service agreement*.

This view is consistent with the *Language-Action Perspective* reflected in the *transaction* concept considered in the works of Dietz [31] and Shishkov [10], where interactions between parties are presented in terms of *commitments* and *negotiations* that are expressed and communicated by means of *elementary communicative acts*, such as *request*, *promise*, *state*, *accept*, and so on.

Taking all this into account and referring to our previous work - [14] (see Fig. 5 on p. 197, featuring our proposed meta-model) and [3] (see Fig. 1 on p. 122, featuring our *context awareness* conceptualization), we propose a **CONTEXT AWARENESS – TRUST CONCEPTUALIZATION** – see Fig. 5. As the figure suggests: One (human) *entity* may fulfill one or more **ROLES**, and types of *roles* (depending on the viewpoint) are **SERVICE PROVIDER / SERVICE USER**, **SENSOR/ACTUATOR**, **PROCESSOR**, **TRUSTOR / TRUSTEE**, and so on. One *role* is restricted by one or more **RULES** and one **REGULATION** comprises one or more *rules*; one or more *roles* are subject of one *regulation*. Going back to *entities*, they are the *composition elements* not only of our **SYSTEM** under consideration but also of its corresponding **ENVIRONMENT** (any *entity* that does not belong to a *system* but interacts with *entities* belonging to the *system* is considered part of the *system environment*; we certainly have the broader notion of **UNIVERSE-OF-DISCOURSE** to cover also *entities* that belong neither to the considered *system* nor to its *environment*). Finally, one or more *systems* are subject of a *regulation*.

Narrowing the discussion to **CONTEXT AWARENESS**, we consider the *role type* **SERVICE USER** and the *system type* **CONTEXT-AWARE SYSTEM**. A *service user* may consume one or more **SITUATION-SPECIFIC SERVICES** and one *context-aware system* is offering one or more such *services*. Then, what is the essence of a *context-aware* service delivery? It concerns the *situation-specific service* being delivered, that should fulfill a particular **NEED** of the *service user* who in turn has one or more *user needs*. Concerning this servicing, the *service user* is part of a broader **CONTEXT** that has one or more **CONTEXT SITUATIONS**. Finally, in its delivering a *situation-specific service* for the benefit of the *service user*, the *context-aware system* should be capable of detecting

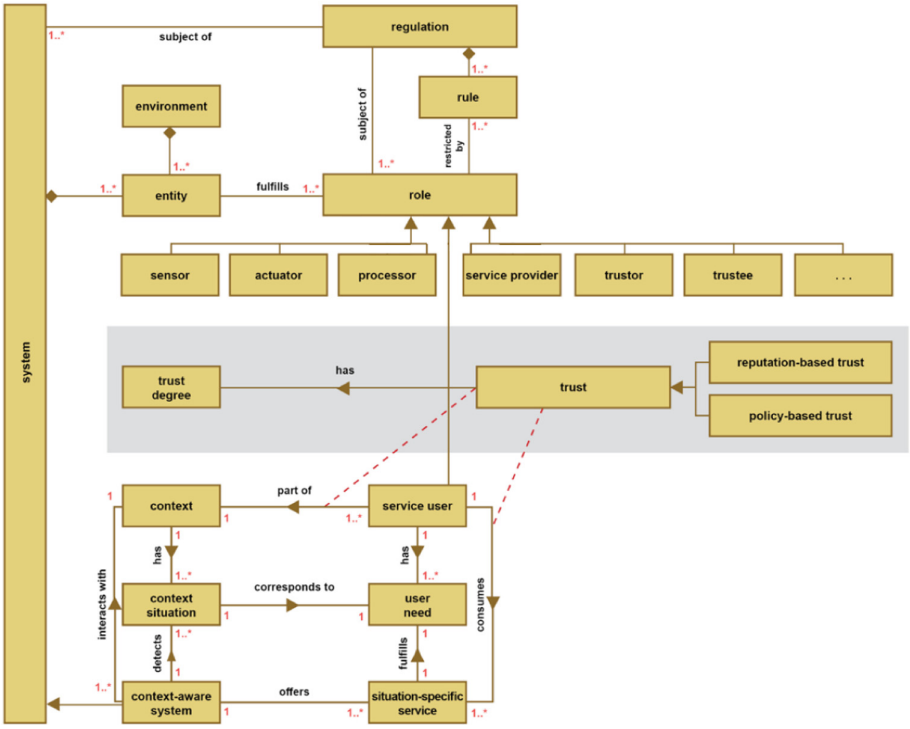

Fig. 5. Context awareness – trust conceptualization

the relevant *context-situation*(*s*) and this is what makes the servicing *situation-specific*, as an essential feature of *context awareness.*

How would we project *trust* in this? We have done this using the *association class* **TRUST** in two directions (see the dashed red lines in Fig. 5):

- [*service user* – *context* association] When a *service user* is consuming a *service*, it is to be taken into account what is his/her **TRUST DEGREE** with regard to the *context* in which (s)he is consuming the *service.*
- [*service user* – *situation-specific service* association] Concerning the above, it is also to be taken into account what is the **TRUST DEGREE** of the *service user* with regard to the *service* itself.

Related to this is the **TRUST DEGREE** class, as represented in the figure (actually, the quantitative perspective of *trust* may be represented using this class). We have also represented the two types of *trust* considered in the current paper, namely **POLICY-BASED TRUST** and **REPUTATION-BASED TRUST**.

And in the end, the *trust relationship* exists at the ROLE level but is driven by a corresponding ENTITY attitude. Said otherwise, it makes difference WHO is fulfilling the *service user* role. We will illustrate in Sub-Sect. 5.1 that different persons fulfilling a *role* would act differently because of different *trust attitudes* both as it concerns the *context* and the particular *service* being consumed.

In this discussion, we are taking a *viewpoint* featuring mainly the **SERVICE USER** *role* type, and it is possible to also take other relevant *trust-related viewpoints*, including a *viewpoint* featuring the **TRUSTOR** and **TRUSTEE** *role* types assuming that the *service user* is (overlaps with) the *trustor role type* and the service providing *system* is a *trustee*. Further, we consider *trust* as a complex mental state (concerning a *trustor* and a *trustee*) that is related to *beliefs* about the *capabilities* and *vulnerabilities* of the *trustee*, and of *intentions* of the *trustor* regarding a *goal* for which (s)he needs *actions* (or absence of *actions*) from the *trustee* – see Fig. 6. In considering this, we refer to related work [32].

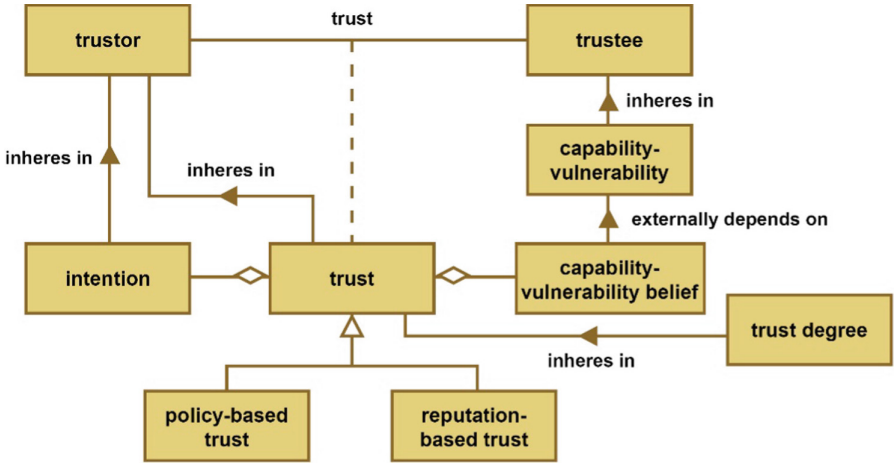

Fig. 6. Simplified trust model

Hence, in **linking trust to context-aware service delivery**, we would also consider identifying: (i) the **CAPABILITIES** and **VULNERABILITIES** of the *system* involved in the *context-aware service delivery*; (ii) how *vulnerabilities* of the *system* can be manifested by **THREAT EVENTS** that potentially cause *loss*, and how *actions* of the *service user* – motivated by the *service user's intentions* and based on his/her *trust* in the *service provider* – bring about *situations* that can trigger *threat events*.

In this regard, we have identified three challenges as follows:

- Aspects, such as *trust attitude* and *intention*, would be hard to capture by means of *sensors*, as in most *context-aware* systems (see Sect. 2).
- Reasoning about *vulnerabilities* is not always in technical terms and may concern aspects, such as *behavior patterns* and *preferences* – those are also hard to capture by means of *sensors*.
- Assuming *service provisioning* that covers very many *service users* would make it "hard-to-implement" arranging *sensor* facilitation for all, guaranteeing for technically solid and unbiased sensing feedback.

Inspired by those challenges, Shishkov and Van Sinderen [4, 33] have considered alternative ways to capture the *user situation* and other aspects concerning the *user*, emphasizing the relevant strengths of *data analytics*. Why is data analytics considered

adequate in this regard? Because: (i) It is not physically restricted to sensor facilitation and hence has the potential of scaling up; (ii) Recent big-data-related developments concern many possibilities to easily, reliably and at low cost provide relevant data; (iii) Counting on historic (training) data allows for applying powerful statistical approaches and algorithms, for the sake of making predictions; (iv) Beyond this, other machine-learning related techniques could be applied for achieving classifications, clustering, and so on; (v) One could often apply in combination data analytics and sensors, for example: capturing emotion via sensors but using the sensor data in combination with data derived by training-data-driven "conclusions"; and so on.

Hence, we argue that **DATA ANALYTICS** can play an important role in the processes of **building user *trust* when working with *context-aware* systems**. The realization of this can be seen in two directions: (1) Using data analytics to **collect and analyze large amounts of data** for the sake of **ensuring better servicing.** (2) **Providing greater transparency of the system's operation**, which leads to **building greater *trust***.

Regarding the first direction, big data collection tools are considered relevant with them allowing for analyses that aim at *personalizing user experience and preferences.* This could be usefully applied in *context-aware recommendation systems.* What is relevant here would span from *classical collaborative filtering algorithms* to state-of-the-art methods using *auto-encoders* to capture the complex interactions between the potential suggestions and the user [37]. This also concerns data analytics aiming at *monitoring the system security* by *tracking its activity* and potential *security threats.* In this way, *vulnerabilities* can be identified and addressed accordingly, thereby building greater user *trust*. Regarding the second direction, methods to make the system more *transparent* would often focus on *data collection processes*, *data processing*, and subsequent *interpretation.*

Further, the main *channels of data collection* in current *context-aware* systems can be divided into two main groups – the vast array of *IoT (Internet-of-Things) devices* [38], as well as the various *media channels*, especially social media. Such an increased data consumption requires special attention on *how to manage trust in the collection, storage and transmission of this data.* In [38], *trust* requirements have been identified, concerning IoT big data systems, such as interoperability-related requirements, security-related requirements, privacy-related requirements, and so on; also, state-of-the-art frameworks, models, and methods for an information-centric *trust* have been discussed, featuring IoT big data systems.

Finally, in terms of *information processing* and further *interaction with the service user*, the ability of the *system* to show the explainability of the decisions made is considered crucially important. In many cases *context-aware systems* use *machine learning models*, with *decision-making processes* that often appear as "black boxes" for the *service user*. This lack of *transparency* may lead to a *trust gap* between the *service user* and the *system.* A possible way to increase the *trust degree* is to explicitly show the *system accuracy* (as in [39]); another possibility would be to provide clear *explanations concerning the underlying logic and reasoning* that have gone into the *system's decision-making process.*

5 Evaluation

The current section presents an example that illustrates our *context awareness – trust* conceptualization and afterwards we discuss some benefits and limitations of our proposal.

5.1 Illustrative Example

Let's consider a system called "**TA**" (*Travel Assistant*), and also **Alice**, **John**, **Sara**, and **Richard** who are "*entities*" not belonging to **TA**. Imagine that each of them can fulfil the *role* "**BT**" (*Business Traveler*) that concerns the *role type* **user**. Then **BT** is part of context "**T**" (*Traveling*) that in turn has a number of *context situations*. Examples would be "**P**" (*Preparation*) - when **BT** needs travel arrangements, such as tickets and accommodation reservations; "**TIP**" (*Transport to Intermediate Point*) - when **BT** is in a process of reaching a bus station or an airport, or a highway (border) point, etc.; and "**O**" (*Orientation*) - when **BT** is in an unknown place and needs location-specific information/services. Imagine that the supportive system detects *context situation* **P** and hence provides to **BT** the *situation-specific service* "***accommodation arrangement***" to fulfil **BT**'s *need* for an accommodation reservation. Let us now consider the two *trust relations* as presented in Fig. 5 (see the dashed red lines):

- **BT is part of context T**: Here, a *trust relationship* exists at a *role level* but is driven by a corresponding *entity attitude*. In our example, the *role* **BT** can be fulfilled by **Alice**, **John**, **Sara**, and **Richard**. Imagine that: (i) **Alice** has no resistance for using any IT systems in any situation; (ii) **John** would always prepare everything beforehand such that he would not need any servicing during his business traveling; (iii) **Sara**, among other things, is involved in an intelligence project focusing on international crime, and for this reason she can only use services that are explicitly authorized by a particular person in the intelligence project; (iv) **Richard** is a brand-driven person who would only go for particular brands during travel. Therefore, the **TRUST ATTITUDE** of the particular *entity* (*person*) who is fulfilling the **BT** *role* (as it concerns *context* **T**) is important.
- When it comes to the **provision of the situation-specific service "*accommodation arrangement*"**, the *trust relationship* concerns the *service* itself. Again, the *entity attitude* is essential. In our example: (a) **Alice** would have a *high-level of trust* with regard to receiving *services* and would not mind using the *service* from **TA** in any way; (b) **John** would like to receive *extra guarantees* from the *system* that the accommodation is confirmed, pre-paid, and cannot be cancelled by the owner; (c) **Sara** would only consume **TA**'s *service* if the *recommendations* would be consistent with a received *authorization* for her project; (d) **Richard** would only consume a *service* from **TA** if the suggested accommodation is one of several selected *brands*. Therefore, again the **TRUST ATTITUDE** of the particular *entity* (*person*) who would be consuming the "***accommodation arrangement***" *service* is important.

5.2 Discussion

We argue that our proposed *ontological conceptualization* helps to sharply describe the relation between *context*, *service provision*, and *trust*. The *conceptualization* has both **strengths** and **limitations**.

One of the *strengths* of the proposed *conceptualization* is that it has been methodologically derived from context-awareness-specific and trust-specific concepts that have been *superimposed* for the sake of achieving an adequate *conceptual alignment*. A particular strength is that we have combined them in one ontological meta-model that establishes the *right restrictions* when either considering *trust* from a *context awareness* perspective or when considering *context awareness* from a *trust* perspective. The *derivation of concepts* stems from *well-focused state-of-the-art studies* featuring *context awareness* and *trust*, reflected in Sect. 2 and Sect. 3, respectively.

Another *strength* of the proposed *conceptualization* is that it is generic in the sense that it is *neither coupled to a specific use case nor is it narrowed to a particular application domain* and is *not restricted in methodological and/or notation terms.*

Limitations of our work are three-fold:

- The proposed conceptualization is still at high level and needs to be specified in more concrete terms;
- It is insufficiently discussed/researched if *policy-based trust* and *reputation-based trust* exhaustively cover the trust "space";
- The illustrative example and this discussion provide only partial justification of the proposed conceptualization and it is still in need of more solid validation (proof-of-principle or proof-of-concept).

6 Conclusions

This paper has considered the incorporation of *trust* in services delivered by *context-aware* (IT) systems, particularly addressing the *user's trust in the system*. We have conceptually aligned *context-aware computing* and concepts from *policy-* and *reputation-based trust*. Two research questions were formulated in the Introduction of the paper: *(i)* How can we allow for *context awareness* in services governed by a *policy-based trust* principle? *(ii)* How can *reputation-based trust* be implemented in a *context-aware service*? Our approach to these research questions was three-fold: First, we have presented a conceptual *context awareness* model, taking a functional perspective, rooting this model in key notions stemming from the evolution of *context-aware computing* in the 1991–2023 period and referring to the key state-of-the-art achievements. Second, we have presented conceptualizations of *policy-based trust* and *reputation-based trust*, and we have outlined possible *trust* strategies. Finally, we have methodologically derived an ontological conceptual meta-model that combines concepts of both *context awareness* and *trust*, also providing insight in the relevant strengths of *data analytics* for predictions of *context* situations and user attitudes, and for users clustering.

We have partially evaluated the conceptual meta-model, by considering an illustrative example and discussing some strengths and limitations of the model.

In future research, we plan to: (a) Consider a larger example and use it to fully validate our proposed conceptual model; (b) Reflect on our proposed conceptualization

in the light of Enterprise Architectures (EA), and study the effects of combining *context awareness* and *trust* in EA.

Acknowledgement. This work was partially supported by: (i) Contract "NGIC – National Geoinformation Center for monitoring, assessment and prediction of natural and anthropogenic risks and disasters" under the Program "National Roadmap for Scientific Infrastructure 2017–2023", financed by the Bulgarian Ministry of Education and Science; (ii) Digitalization and Information Systems Group - University of Fribourg; (iii) Faculty of Electrical Engineering, Mathematics and Computer Science – University of Twente; (iv) Faculty of Technology, Policy, and Management – Delft University of Technology.

References

1. Dey, A., Abowd, G., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum.-Comput. Interact. **16**(2–4), 97–166 (2001)
2. Wegdam, M.: Awareness: a project on context aware mobile networks and services. In: Proceedings of the 14th Mobile & Wireless Communications Summit. EURASIP (2005)
3. Shishkov, B., van Sinderen, M.: Towards well-founded and richer context-awareness conceptual models. In: Shishkov, B. (ed.) BMSD 2021. LNBIP, vol. 422, pp. 118–132. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79976-2_7
4. Shishkov, B., van Sinderen, M.: On the context-aware servicing of user needs: extracting and managing context information supported by rules and predictions. In: Shishkov, B. (ed) Business Modeling and Software Design. BMSD 2022. Lecture Notes in Business Information Processing, vol. 453. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-11510-3_15
5. Dey, A.K., Newberger, A.: Support for context-aware intelligibility and control. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, USA (2009)
6. Bosems, S., van Sinderen, M.: Models in the design of context-aware well-being applications. In: Meersman, R., et al. (eds.) OTM 2014. LNCS, vol. 8842, pp. 37–42. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45550-0_6
7. Alegre, U., Augusto, J.C., Clark, T.: Engineering context-aware systems and applications: a survey. J. Syst. Softw. **117**, 55–83 (2016). https://doi.org/10.1016/j.jss.2016.02.010
8. Alférez, G.H., Pelechano, V.: Context-aware autonomous web services in software product lines. In: Proceedings of the 15th International SPLC Conference. IEEE, CA, USA (2011)
9. Abeywickrama, D.B., Ramakrishnan, S.: Context-aware services engineering: models, transformations, and verification. ACM Trans. Internet Technol. J. **11**(3), 1–28 (2012). ACM
10. Shishkov, B.: Designing Enterprise Information Systems, Merging Enterprise Modeling and Software Specification. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-22441-7
11. Bonatti, P., Duma, C., Olmedilla, D., Shahmehri, N.: An integration of reputation-based and policy-based trust management. Networks **2**(14), 10 (2007)
12. O'Hara, K., Alani, H., Kalfoglou, Y., Shadbolt, N.: Trust strategies for the semantic web. In: Proceedings of the 2004 International Conference on Trust, Security, and Reputation on the Semantic Web - Volume 127 (ISWC'04). CEUR-WS.org, Aachen, DEU, pp. 42–51 (2004)
13. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publ. Inc., San Francisco, CA, USA (2011)
14. Shishkov, B., Larsen, J.B., Warnier, M., Janssen, M.: Three categories of context-aware systems. In: Shishkov, B. (ed.) Business Modeling and Software Design. BMSD 2018. Lecture Notes in Business Information Processing, vol. 319. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_12

15. Knittel, J., Shirazi, A.S., Henze, N., Schmidt, A.: Utilizing contextual information for mobile communication. In: CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA (2013)
16. Weiser, M.: The Computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev. **3**(3), 3–11 (1999). ACM, New York, NY, USA
17. Schilit, B.N.: A system architecture for context-aware mobile computing. Ph.D. dissertation, Columbia University, New York, USA (1995)
18. Krumm, J. (ed.): Ubiquitous Computing Fundamentals. Taylor and Francis Group, LLC (2009)
19. Dey, A.: Chapter 8 - Context-Aware Computing. In: Krumm, J. (ed.) Ubiquitous Computing Fundamentals. Taylor and Francis Group, LLC (2009)
20. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: First Workshop on Mobile Computing Systems and Applications, pp. 85–90. IEEE (1994)
21. Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P.: The anatomy of a context-aware application. Wirel. Netw. **8**, 187–197 (2002)
22. Dey, A.K.: Context-aware computing: the cyberdesk project. In: AAAI Spring Symposium on Intelligent Environments, AAAI Technical Report SS-88-02, pp. 51–54 (1998)
23. van Sinderen, M., van Halteren, A., Wegdam, M., et al.: Supporting context-aware mobile applications: an infrastructure approach. IEEE Commun. Mag. **44**(9), 96–104 (2006)
24. Chaari, T., Laforest, F., Celentano, A.: Adaptation in context-aware pervasive information systems: the SECAS project. Int. J. Perv. Comput. Commun. **3**(4), 400–425 (2007)
25. Borissova, D., Cvetkova, P., Garvanov, I., Garvanova, M.: A framework of business intelligence system for decision making in efficiency management. In: Saeed, K., Dvorský, J. (eds.) CISIM 2020. LNCS, vol. 12133, pp. 111–121. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47679-3_10
26. Garvanova, M., Garvanov, I., Kashukeev, I.: Business processes and the safety of stakeholders: considering the electromagnetic pollution. In: Shishkov, B. (ed.) BMSD 2020. LNBIP, vol. 391, pp. 386–393. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52306-0_28
27. Dimitrova, Z., Dimitrov, V., Borissova, D., Garvanov, I., Garvanova, M.: Two-stage search based approach for determination and sorting of mountain hiking routes using directed weighted multigraph. Cybern. Inf. Technol. **20**(6), 28–39 (2020). Print ISSN 1311-9702 Online ISSN 1314-4081. https://doi.org/10.2478/cait-2020-0058
28. Van Engelenburg, S.: Designing context-aware architectures for business-to-government information sharing. Ph.D. thesis. TU Delft Press (2019)
29. Nardi, J.C., et al.: Towards a commitment-based reference ontology for services. EDOC, pp. 175–184 (2013)
30. Nardi, J.C., et al.: A commitment-based reference ontology for services. Inf. Syst. **54**, 263–288 (2015)
31. Dietz, J.L.G.: Enterprise Ontology, Theory and Methodology. Springer, Heidelberg (2006)
32. Amaral, G., Sales, T.P., Guizzardi, G., Porello, D.: Towards a reference ontology of trust. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 3–21. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_1
33. Shishkov, B.: Tuning the behavior of context-aware applications. In: Shishkov, B. (ed.) BMSD 2019. LNBIP, vol. 356, pp. 134–152. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24854-3_9
34. Muff, F., Fill, H.-G.: A framework for context-dependent augmented reality applications using machine learning and ontological reasoning. In: Proceedings of the '22 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence. AAAI-MAKE 2022. Stanford University Press, Palo Alto, CA (CEUR Workshop Proceedings) (2022)

35. Kindervag, J.: No More Chewy Centers: Introducing the Zero Trust Model of Information Security. Forester (2010)
36. Bunge, M.A.: Treatise on Basic Philosophy, vol. 4, A World of Systems. D. Reidel Publishing Company, Dordrecht (1979)
37. Abinaya, S., Alphonse, A.S., Abirami, S., et al.: Enhancing context-aware recommendation using trust-based contextual attentive autoencoder. Neural Process. Lett. (2023). https://doi.org/10.1007/s11063-023-11163-x
38. Ahmed, U., Raza, I., Hussain, S.A.: Information-centric trust management for big data-enabled IoT. Big Data-Enabled Internet of Things **2020**, 411–432 (2020)
39. Antifakos, S., Kern, N., Schiele, B., Schwaninger, A.: Towards improving trust in context-aware systems by displaying system confidence. In: MobileHCI '05: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services
40. NESDIS: National Environmental Satellite, Data, and Information Service (2023). https://www.nesdis.noaa.gov
41. JMA: Japan Meteorological Agency (2023). https://www.jma.go.jp/jma
42. EPOS: European Plate Observing System (2023). https://www.epos-eu.org
43. Sensor. Community: Sensor Community (2023). https://sensor.community/en
44. Mendling, J.: Towards Blockchain Support for Business Processes. In: Shishkov, B. (ed.) Business Modeling and Software Design. BMSD 2018. Lecture Notes in Business Information Processing, vol. 319. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_15