Tuning the Behavior of Context-Aware Applications

Using Semiotic Norms and Bayesian Modeling to Establish the User Situation

Boris Shishkov[1,2,3](✉)

[1] Faculty of Information Sciences, University of Library Studies and Information Technologies, Sofia, Bulgaria
b.b.shishkov@iicrest.org
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
[3] Institute IICREST, Sofia, Bulgaria

Abstract. Context-aware applications are to adapt their "behavior" to the surrounding context. In this paper, we analyze different ways to achieve adequate application behavior adjustment (based on context data) and we stress upon: (i) Bayesian modeling that is not only considered useful in this regard but is also not enough explored as it concerns context-aware applications; (ii) semiotic norms that have specific relevant strengths. Even though there is much experience as it concerns the challenge of capturing context data, more knowledge is still needed about how to use context data in order to effectively make the right judgement about the "current" user situation (context state). We consider this paper's contribution as relevant to the above-mentioned challenge.

Keywords: Context-awareness · Context data · Semiotic norm · Bayesian model

1 Introduction

A person would often need to adapt his or her behavior to the "current" situation, for example: if the nearby supermarket is open, then Samuel may purchase products and then cook dinner at home but if it is too late and all nearby shops are closed, then he would opt for getting back home and calling a 24/7 delivery company to order a pizza. In a similar way, an organization would "behave" differently in different situations, for example: if there are indications for an impending recession, then the organization managers would go for firing employees and cutting spending also in other ways while if the prospects are good, then the managers would be more relaxed as it concerns costs. Alferez and Pelechano claim that it is desirable to translate the ideas of adaptation in the natural world to software, assuming that such adaptations are carried out in response to changing conditions in the supporting computing infrastructure and/or in the surrounding physical environment; this is referred to as "context-awareness", especially as far as (enterprise) information systems are concerned - there should be

B. Shishkov (Ed.): BMSD 2019, LNBIP 356, pp. 134–152, 2019.
https://doi.org/10.1007/978-3-030-24854-3_9

"automatic" adaptation mechanisms to reconfigure them according to contextual changes because assigning manual reconfiguration tasks would be impractical [1, 2]. This is in line with the views of Dey et al., expressed still in 2001, suggesting that context-aware applications use context that is relevant to the interaction with users – by "application" is meant "ICT (Information and Communication Technology) application"; by "context" they mean information that concerns the state of people, places, and objects [3]. In further studies, Dey and Newberger argue that context is typically gathered in an automated fashion [4]. This claim (as it concerns context-aware applications) is in concert with the views of Alferez and Pelechano (see above) and is inspired by the observation that currently many users have to deal with diverse devices (including small (wearable) devices with sensing and computing capabilities) accessed through diverse interfaces and used in diverse environments. Hence, it is not surprising that in 2009, Papadimitrious stated that: Context-awareness, ubiquity (device independence, mobility, wireless support), quality of service provisioning, seamless discovery of services and content, and enhanced user control and effective delivery are important requirements of the future Internet [5].

This is the focus of the current paper, in general, and in particular - we address the specification of context-aware applications. The paper builds on the research presented in [6–8]. As it concerns such applications, we call "behavior" what the application delivers as functionality and we assume that different behavior variants are to be triggered corresponding to different (surrounding) situations. Further, we have identified three behavior perspectives (in this regard), namely: context-driven optimization of system-internal processes, context-driven maximization of the user-perceived effectiveness, and context-driven value-sensitivity. Nevertheless, we stay challenged by the adaptation issue itself: HOW the application "knows" which is the right behavior change to implement upon changing conditions? It is for sure that much data is available - we are showered by sensor data, reports, inferred data, and so on. Still, it would rarely be trivial reflecting such data into MEANINGFUL INFORMATION, as a basis for the application to adequately "establish" the "current" situation and hence "know" which behavior variant to trigger. We address this problem in the current paper.

The '16 claim of Alegre et al. that "the challenges of context-aware systems development are diverse and complex, provoking development techniques (and methods) to be commonly disconnected from each other, and focused on solving specific issues" [9] indirectly justifies the validity of the identified problem as well as the claim of Bosems and Van Sinderen that "designers cannot always anticipate the dynamics of context and associated user requirements" [10]. Actually, we elaborate the adaptation challenge as follows: (i) if a situation occurs that has been "foreseen" during the design, then a corresponding behavior variant (specified at design time) is triggered; (ii) otherwise, there is no other option but relying on an intelligent run-time adaptation. In the current paper, we focus on (i) and abstract from (ii).

Further, we make the following assumptions: • There are several possible situations in which the context-aware application ("application", for short) of consideration can be, featured as: Hypothesis 1, Hypothesis 2, Hypothesis 3, and so on. • It is possible to know in advance each of the hypotheses. • For each of the hypotheses, there is a corresponding desired application behavior variant and this corresponds to the adequate functioning of the application.

Hence, we pose the following research question: HOW CAN CONTEXT DATA BE USED TO EFFECTIVELY ADJUST THE APPLICATION BEHAVIOR FOR ACHIEVING ADEQUATE PERFORMANCE?

We consider this research question as relevant to the identified problem (see above) and we therefore claim the following contribution (of the current paper) that is two-fold: • We explicitly consider the way context data is used to adjust application behavior and in our view, even though this has been covered by related work (some related work was already discussed in this section), this has not been done explicitly. • We analyze different ways to achieve adequate application behavior adjustment (based on context data) and we stress upon Bayesian modeling [16] that is not only considered useful in this regard but is also not enough explored as it concerns context-aware applications. We also consider semiotic norms [13] in this regard.

The remaining of the current paper is structured as follows: A problem elaboration follows in Sect. 2 and a related work analysis – in Sect. 3. Section 4 is featuring the paper's conceptual background. Further, Sect. 5 is providing an analysis-driven proposal (complemented by a partial exemplification) featuring the use of semiotic norms and Bayesian modeling for the sake of establishing the user situation. Finally, Sect. 6 concludes the paper.

2 Problem Elaboration

As mentioned in the introduction, we do problem elaboration in the current section, considering as a starting point the aim of effectively using context data for the sake of adequately adjusting application behavior. As also mentioned in the introduction, we are particularly challenged in general by the application adaptation itself and in particular - by the issue of "letting" the application "know" which is the right behavior change to implement upon changing conditions. Finally, we see no other decision "trigger" to this than CONTEXT DATA – we argue that it can only be context data that would "say" to the application that the surrounding context is changing and hence application behavior updates need to be realized accordingly. Said otherwise, we need data concerning the application context in order to "capture" context changes that in turn require application behavior adaptations. This problem is nevertheless not new and just one example featuring this claim points to the period 2005-08 when the AWARENESS framework was dominated by a similar focus; in particular, the AWARENESS framework was covering tele-monitoring services as follows: a health-monitored person is away from hospital but "wearing" a "body area network" (consisting of body vital sign sensors + device(s) performing processing and connectivity) that would allow the AWARENESS platform "know" if something in the situation of the person is changing that requires an update in the AWARENESS support, for example: in case of an upcoming epileptic seizure, it would no longer be enough to just monitor the person and it would be needed to activate emergency help [11]. There are also other similar examples featured in some of the related work sources considered in the introduction.

Still, in our view, a limitation of all those works is that even though they consider the problem of adapting application behavior based on capturing context data, they only address more or less “simple” cases when it is somehow “straightforward” to align the captured data to the need to do a particular thing. For example, the AWARENESS sensor readings are considered in a simple way – some particular value combinations point to the “conclusion” that “an epileptic seizure is coming”; otherwise it is assumed that the person is in normal condition. In this regard, the AWARENESS platform would count on ECA rules [12].

We would not challenge those achievements. However, we claim that in this way it would be difficult to resolve some more complex situations, especially when the context data readings are not a “straightforward basis” for identifying a context change. This we claim for AWARENESS and also for the related work we have studied – see the introduction and the following section.

We hence argue that the system engineering community still misses EXPLICIT and EXHAUSTIVE ways of considering context data, driven by the purpose of updating application behavior (if needed).

We contribute to filling this gap, by analyzing the relevance of OS - Organizational Semiotics [13] and Data Analytics [14]. In particular, we address the OS Norm Analysis Method as well as Statistical Data Analytics [15] and especially the Naïve Bayesian Classification Approach [16], expecting that they have potential to add value in this regard. This will be especially considered further on in the paper, after the related work analysis and the introduction of several essential relevant concepts.

3 Related Work

The current related work analysis section is organized as follows: Firstly, we consider our previous work that we find relevant with regard to the identified problem (see Sect. 2); Secondly, we expand our analysis to cover also other relevant work.

As it concerns our previous work: • In [8], we have considered the specification of context-aware applications, making it explicit that following context changes, the application behavior is to be updated accordingly. Even though we have proposed some solution directions in this regard, we have only implicitly considered context data and the challenge of approaching it. • In [17], we have taken a systemics [18] perspective over context-awareness, addressing in this regard the environment and its changes, to which the system should adapt. Nevertheless, context data has been considered just abstractly. • In [6], we have considered three system adaptation perspectives with regard to context-aware systems, namely: (a) driven by the goal of optimizing the system-internal processes; (b) driven by the goal of maximizing the user-perceived effectiveness; (c) driven by the goal of achieving sensitivity to public values. Further, we have explicitly established that in each of those cases we have a different perspective over the context – as it concerns (a), the context is about what is happening inside the system; as it concerns (b), the context concerns the user, as it concerns (c), the context concerns public values. Nevertheless, we have only implicitly considered in this regard the way context data is used. • In [7], we have considered business process modeling from the perspective of context-awareness, addressing in particular business process variants – different business process variants could be relevant to corresponding context situations. Again, we have only implicitly considered in this regard the way context data is used.

As it concerns other related work: • Anind Dey is among the most recognizable researchers addressing context-awareness [3, 4]. He has serious achievements in considering the notion of context and also the development of context-aware applications. We argue nevertheless that he has not explicitly considered the challenge of properly using context data for sensing a context change, counting instead on a more "intuitive" approach to this challenge. • The same (lack of explicit consideration of context data) holds for most recognizable R&D context-awareness projects, such as AWARENESS [11], as discussed already. • Bosems and Van Sinderen have considered the notion of "context-aware computing" as the combination of sensor, reasoning, and other technology that provides systems with real-time awareness … [10] but the "reasoning" has not been explicitly considered and is mainly related to ECA rules [12] that in our view have only limited "power" as it concerns complex situations and corresponding context data considerations. Further, those authors are more focused on deriving higher-level context information based on "raw" context data than on the consideration of the context information itself for adequately sensing context changes. • The useful survey of Alegre et al. [9] is mainly focused on the development (featuring implementation concerns) of context-aware applications as well as on the consideration of some public values but not so much on the consideration of context data. • The same holds for the works of Alférez and Pelechano [1, 2] – they consider the dynamic evolution of context-aware systems, the development itself, and the relation to web services. • A service-orientation perspective with no explicit context data consideration is also characterizing the works of Abeywickrama [19, 20].

Even though we do not claim exhaustiveness with regard to the current related work analysis, we are convinced that it covers some of the most representative researchers and works relevant to the problem considered in this paper.

Hence, we argue that it is still a question how to effectively consider context data for the sake of adequately updating the behavior of a context-aware application (if needed).

As mentioned in the previous section, our proposal is featured in the following two sections, with us firstly bringing forward the conceptual background and secondly – our proposed solution directions.

4 Conceptual Background

The current section is organized as follows: Firstly, we present the meta-model "governing" the essential concepts that we consider relevant with regard to context-awareness; Secondly, we address some of them, namely the concepts "system", "environment", and "user" – we argue that those concepts are important as it concerns the context-data-driven adaptation of application behavior; Finally, we summarize our context-awareness views, also touching upon context-aware applications.

4.1 Meta-Model

We refer to our previous work [6] featuring a proposed meta-model that we consider relevant to the problem addressed in the current paper. The meta-model is presented in Fig. 1, using the notations of the UML Class Diagram [21].

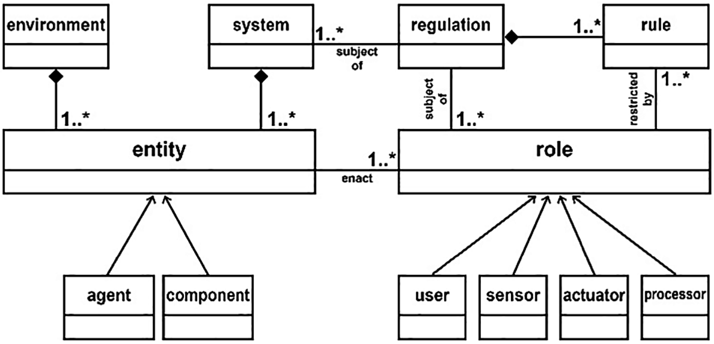

Fig. 1. Considered meta-model (Source: [6], p. 197)

As it is seen in the figure, we consider a *system* and its *environment*. Both are composed of numerous *entities* which in turn can be *components* (non pro-active) or *agents* (pro-active and intelligent). One *entity* (an *agent*, for example) can enact many different *roles* (and in this research, we limit ourselves to four role categories, namely: *user*, *sensor*, *actuator*, and *processor*) that are restricted by corresponding *rules* and are subject of *regulations*. A regulation in turn is composed of many *rules* and is affecting not only the *roles* but the *system* as a whole.

Since we are taking particularly an *agent perspective*, we consider it important *matching roles to their corresponding executing agents* because it is not for sure that *anybody* would have the right capabilities of fulfilling a *role*.

4.2 Essential Concepts - Elaboration

As mentioned already, in the current sub-section we address the concepts: “system”, “environment”, “user”. In this regard, we refer to the *system definition* of Bunge [18]:

> **Definition**: Let T be a nonempty set. Then the ordered triple $\sigma = <C, E, S>$ is **system** over T if and only if C (standing for *Composition*) and E (standing for *Environment*) are mutually disjoint subsets of T (i.e. $C \cap E = \emptyset$), and S (standing for *Structure*) is a nonempty set of active relations on the union of C and E. The system is *conceptual* if T is a set of conceptual items, and *concrete* (or material) if $T \subseteq \Theta$ is a set of concrete entities, i.e. things.

Hence: • There are “items” belonging to the system under consideration; • There are also items not belonging to the system under consideration (Some of those items would appear to belong to the system environment; Others would therefore appear to belong neither to the system nor to the system environment).

What about the USER? This notion is not explicitly considered above and we need to discuss it – for this, we use Fig. 2.

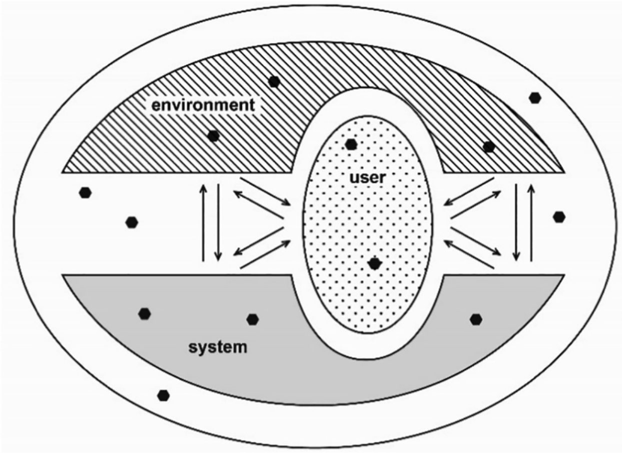

Fig. 2. Considering the notions of *system*, *environment*, and *user*

As the figure suggests, it is a "delicate" issue whether the *user* belongs to the *system* or to the *environment*. From one point of view, the *system* is driven by the goal of delivering something to the *user* and hence, the user is to be considered part of the *system*; nevertheless, from another point of view, the *user* is not among the entities who are delivering the product/service because the *user* is consuming it and hence the *user* is not to be considered part of the *system* (and is thus part of the *environment*) [17]. It is therefore not surprising that a lack of consensus is observed about how the *user* is to be considered. Hence, we clearly distinguish between: (i) what belongs to the *system*; (ii) what belongs to the *environment*; (iii) what belongs to the *user*.

Further, in line with what was stated above: there are items that neither belong to the *system*, nor to the *environment*, nor to the *user*.

Finally, those "items" (visualized in Fig. 2 as small black hexagons) actually reflect ENTITIES (as according to the meta-model – see Fig. 1) and they in turn fulfill actor-roles (ROLES, for short), for example: if a professor sends a fax, then (s)he is fulfilling the *role* "secretary".

In summary, there is interaction among entities (fulfilling corresponding roles) in several perspectives: • between system and environment; • between system and user; • between environment and user. Other entities are not involved in interactions, at least as it concerns the view over the system under consideration.

For example: John brings his Mitsubishi Colt to a Mitsubishi garage [22] for a motor vehicle service and they establish that one of the exhaust pipes would need to be replaced. Their doing the replacement for John concerns a *system-user* "relation".

Nevertheless, they do not have in stock the particular Mitsubishi part (an exhaust pipe) needed for the car of John and they have to order it from an "external" company – Bosal [23]. Their arranging this with Bosal concerns a *system-environment* relation. Still, since Bosal is outside the Mitsubishi "family", the order can only be paid (and thus guaranteed) by the *user* directly. Hence, in order to allow the garage to fulfill the order, John would have to do a payment to Bosal (this is just for the exhaust pipe itself; apart from this, John would have to pay to the garage for their servicing the car, replacing the exhaust pipe, and so on) and this concerns an *environment-user* relation. In summary, as it can be seen from the above example, often, in delivering a service to the *user*, the "system" needs some interaction with entities belonging to the system *environment*. It is also possible that the *user* himself/herself would need to interact with entities belonging to the system *environment*, in the process of utilizing a service delivered by the *system*. The entities belonging to the system *environment* are only those entities with whom the *system* and/or the *user* would need to interact in the process of the *system-to-user* service delivery. All other system-external entities are "outside" the system *environment*.

As it concerns the perspectives considered in the 2nd paragraph of the introduction, in the remaining of this paper, <u>we only focus on application behavior adaptations driven by the goal of maximizing the user-perceived effectiveness</u>. Further, as mentioned in the introduction, we only consider in the paper situations that can be "foreseen" during the design, such that corresponding behavior variants (specified at design time) are triggered accordingly. This also corresponds to the assumptions made in the introduction.

Thus, considering Fig. 2 could be a good starting point in approaching the problem (see Sect. 2), taking the above into account. In this, we are to <u>focus on the system-to-user service delivery</u>. Further, we are to be "sensitive" to the <u>different situations the user may find himself/herself in</u>. Finally, <u>for each of those situations, the system should offer a corresponding adequate behavior variant</u>.

Those issues are already context-awareness-specific and will be considered in the following sub-section.

4.3 Adopting Context-Awareness

Referring to previous work [8], we consider as a key context-awareness feature the capability of a system to adapt its behavior based on the user situation, as illustrated in Fig. 3. As the figure suggests, each situation of the user "asks for" a corresponding system behavior variant.

This is in line with what was already stated about our particular focus in the current paper, just covering the goal of achieving a user-perceived effectiveness and also abstracting from situations not foreseen at design time.

We also abstract from numerous design-related issues, such as the system behavior specifications, the "switch" between one behavior variant to another, and so on.

We only focus in this paper on CONTEXT DATA and HOW it helps identifying the USER SITUATION. Referring to the example considered in the previous sub-section: If John is the only person driving the Mitsubishi Colt and we are able to "sense" if the car is moving or not, then based on sensor data, we would know if John

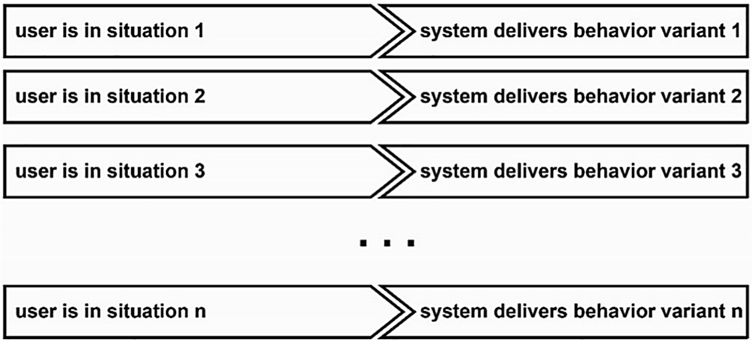

Fig. 3. Visualizing a key feature of context-aware systems

is in his car, driving, or John is outside his car. If John is driving, then particular (vehicle-specific) services would be offered to him; otherwise, standard services would be offered.

This is a simple example featuring a case when it is straightforward to use context data in order to immediately establish the user situation (this in turn gives the possibility of offering the right services accordingly).

Nevertheless, real-life cases may be more complex. For example: The managers of the HKairport Electronics Shop, located at the Hong Kong Airport [24], would look for ways to effectively approach customers, taking into account that all customers are passengers whose time is often severely restricted by pending flights. Still, many passengers have loyalty cards that would immediately provide the shop with much data. Then if Suzan, Dave, and Steve are three customers possessing loyalty cards, how could it be established for each of them whether (s)he is more likely to buy this or that kind of product or not? If in particular, the person under consideration is SUZAN and the product type – TABLET, then the question is: IS IT MORE LIKELY THAT SUZAN PURCHASES A TABLET? This points to exactly two user situations: (i) It is more likely that Suzan purchases a tablet – Situation 1; (ii) It is more likely that Suzan would not purchase a tablet – Situation 2. If it is Situation 1, then the tablets vendor would immediately approach Suzan with focused questions and suggestions. If it is Situation 2, then the tablets vendor would most probably "ignore" Suzan and approach another customer instead. Hence, it is not as easy as in the above example (featuring John and his Colt) to establish (based on the context data) which the "current" user situation is.

Further, we argue that this is not so much an issue of how we derive the context data (one possibility is to use sensors, another possibility is to use reports, and so on) but it is more an issue of WHAT WE DO (and HOW) with the context data, such that we adequately establish the "current" user situation.

How we propose dealing with this challenge is featured in the following section.

5 Proposed Solution Directions

The current section is organized as follows: Firstly, we carry out an analysis featuring the context data consideration challenge with regard to context-aware systems, in general, and context-aware applications – in particular. Secondly, we introduce and discuss semiotic norms and Bayesian modeling as relevant and potentially useful with regard to the mentioned challenge. Finally, we provide partial exemplification, as a first step in justifying and validating the appropriateness of using semiotic norms and Bayesian modeling to establish the user situation.

5.1 Analysis

We argue that a key issue to be discussed when addressing context data is how we get the data. In this regard, our observation is (considering related work – see Sect. 3) that most often context-aware applications count on sensor data. An example for this is the AWARENESS Body Area Network (BAN) that uses sensors attached to a person's body, for getting "vital signs - vs" (vs represent "higher-level" context data featuring blood pressure, pulse, and so on, that is "inferred" based on "lower-level" sensor readings), for the sake of determining the situation of the person [11]. Nevertheless, it is also possible to count on "predictions", referring to Statistics [15] and/or Machine Learning [14]. For instance, in the HKairport Electronics Shop (see the example considered in the previous section), it might be possible to predict the likelihood that a particular customer would purchase a particular kind of product – this in turn would allow for determining his or her situation (if the person would most probably purchase a particular kind of product, then (s)he would be treated in one way; otherwise, the person would be treated in another way). Anyway, the idea of using prediction data instead of sensor data has advantages and drawbacks. It is certainly useful to apply Statistics/Machine Learning for getting a precise prediction concerning the situation of the user, especially in complex cases when the applicability of sensors is limited. At the same time, predictions, as uniquely based on the *training data*, would not evolve in time (as the user needs may); thus, prediction-based applications might get outdated soon. We argue that a solution would be to use Machine Learning also for adapting predictions if the context would change.

Hence, we observe that most current context-aware systems are either SENSOR-based or PREDICTION-based, and, not claiming exhaustiveness, we assume two types of context-aware applications, namely: *sensor-based context-aware applications* and *prediction-based context-aware applications*.

Further, we have no doubt that in simple cases (for example: "user is at home" vs "user is driving" vs "other") developers would lean towards counting on sensors, while in more complex cases (when for example the behavior of the user would need to be "foreseen") developers would lean towards using predictions, as illustrated in Fig. 4(a). That is because in "simple" cases, we usually have some "physical" conditions determining the situation of the user and those conditions would often be easy to capture by means of sensors [25]. We argue that in contrast, most "complex" cases concern the "mental state" of the user (and/or (an)other person(s)) and this would be difficult to "capture" by means of sensors (at least counting on the current advances of sensor technology); nevertheless, prediction techniques could be effective in such cases because they would help "deriving" information about the "mental state" of a person, by considering *training data* that is featuring other persons [14].

Finally, we position such *training-data*-based approaches as mainly relevant to "complex" cases just because a simple case would not "justify" sophisticated analytics activities. In our view, in "simple" cases, it would be more appropriate applying *rules*. This is illustrated in Fig. 4(b).

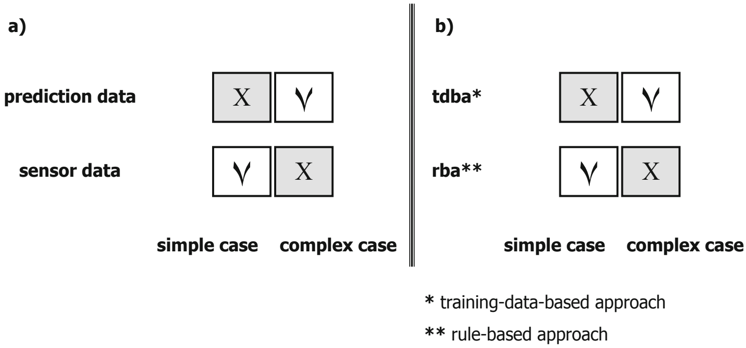

Fig. 4. Sensor-based vs prediction-based context-aware systems; Rule-based approaches vs training-data-based approaches.

We are thus challenged to address: • "simple" cases, counting on SENSOR DATA and RULE-BASED APPROACHES; • "complex" cases, counting on PREDICTION DATA and TRAINING-DATA-BASED APPROACHES.

In this, we are neither claiming that all "simple" cases should assume counting on sensor data and rule-based approaches (and that all "complex" cases should assume counting on prediction data and training-data-based approaches) nor we are claiming that it would always be possible to clearly distinguish between a "simple" and a "complex" case. Actually, we use those labels ("simple" vs "complex") mainly to distinguish between cases driven mostly by physical conditions and cases driven mostly by the "mental state" of persons. It is therefore established that physical conditions are usually easier to tackle by using sensors and rules while "mental states" would require more sophisticated "instrumentarium", possibly assuming prediction data and training-data-based techniques.

Hence, the contribution of the current paper (besides its analytical part) is partitioned in two streams: • RULE-BASED CAPTURING OF THE USER SITUATION; • TRAINING-DATA-BASED CAPTURING OF THE USER SITUATION.

As it concerns the rule-based capturing of the user situation, we consider ECA rules [12] to be currently most popular, applied in many developments, as for example the AWARENESS framework [11]. Even though we would not criticize ECA rules (we actually acknowledge their usefulness in many (mostly simple) cases), we would allow ourselves claiming that often a more solid approach is needed, especially in more complex cases when the rules would need to be aligned among each other and also with other techniques. That's why we would better appreciate a rule technique that not only has solid theoretical roots but also can be effectively combined with other modeling techniques. We argue that such a technique is the Norm Analysis Method that is "within" Organizational Semiotics [13]. In the following sub-section, we will briefly introduce the method and explain its relevant strengths.

As for the training-data-based capturing of the user situation, this concerns *supervised machine learning* [14] - when predictions are made with the help of labelled datasets, as in the case of *classification* (when the output variable is categorical) or in the case of *regression* [15] (that is about the relationship between two or more variables where a change in one variable is associated with a change in another variable). We argue that the *Naïve Bayesian Classification Approach* [14], that is helpful in establishing (based on attributes data and probabilities calculations) which hypothesis (out of two or more) is most likely to "occur", is a useful tool – we will briefly introduce it in the following section and we will also explain its relevant strengths.

5.2 Elaboration

As mentioned already, in this sub-section we will address the Norm Analysis Method and the Naïve Bayesian Classification Approach – they will be briefly introduced and their relevant strengths will be explained (as it concerns the context-data-driven establishment of the user situation, that is to trigger in turn application behavior adaptations).

5.2.1 Norm Analysis Method

With regard to *Organizational Semiotics* (*OS*), in general, and the *Norm Analysis Method*, in particular, we refer to [13, 17].

OS is based on the *Semiotics* theory and is focused on the nature, characteristics, and behavior of *signs*. *OS* adopts a subjectivist philosophical stance and an agent-in-action ontology; this philosophical position states that, for all practical purposes, nothing exists without a perceiving agent and the agent engaging in actions. Another essential *OS* concept is the notion of *affordance*: the affordance of the environment is considered to be "what it offers the animal, what it provides or furnishes, either for good or ill…".

When studying enterprises from the perspective of entities' behavior, it is necessary to specify the *norms* based on which this behavior is realized. Norms (featured by the OS Norm Analysis Method) are the rules and patterns of behavior, either formal or informal, explicit or implicit, existing within a society, an enterprise, or even a group of people working together to achieve a common goal. Norms are determined by Society or collective groups, and serve as a standard for the members to coordinate their actions. Hence, specifying an organization can be done by specifying the norms. Four norm types exist: *evaluative*, *perceptual*, *cognitive*, and *behavioral norms*. Each type of norms governs human behavior from different aspects. In business process modeling, most rules and regulations fall into the category of BEHAVIORAL NORMS - they prescribe what people *must*, *may*, *must not* do, reflecting the *three deontic operators*: • is obliged; • is permitted; • is prohibited. Hence, the following behavioral norm format is adopted:

```
whenever <condition>
if <state>
then <agent>
is <deontic operator>
to <action>
```

It is essential to recognize that norms are not as rigid as logical conditions. If a person does not drink water for certain duration of time, (s)he cannot survive. But an individual who breaks the working pattern of a group does not have to be punished in any way. Further, for those actions that are *permitted*, whether the agent will take an action or not is seldom deterministic.

A *norm analysis* is normally carried out on the basis of the results of a semantic analysis that would have "delineated" the area of concern. The behavior patterns specified in the semantic model are part of the fundamental norms that retain the ontologically determined relationships between agents and actions without imposing any further constraints.

The Norm Analysis Method can also be successfully related to other modeling tools.

As it concerns context-aware applications, we argue that in most of the SIMPLE cases (cases that are mainly driven by behavior patterns in the physical perspective), applying norms would be useful and effective.

We partially justify this claim by means of an illustrative example, presented in the following sub-section.

5.2.2 Naïve Bayesian Classification Approach

With regard to the *Naïve Bayesian Classification Approach* (*NBCA*), we refer to [14, 26]. *NBCA* concerns *Machine Learning* (*ML*), in general, and *Supervised ML* (*SML*), in particular. *ML* is a method of teaching computers to make predictions based on some data. Further, let's consider the equation: **`y=f(x)`** where x is the INPUT VARIABLE and y is the OUTPUT VARIABLE. *SML* works under SUPERVISION (it's "learning") – a machine is streamed with data which is LABELLED, such that the machine can make a prediction (with the help of a labelled data set – data for which we already know the target "answer"). *SML* is basically of two types (see the previous sub-section): • CLASSIFICATION – when the output variable is *categorical* (i.e. with two or more classes), for example: *red* or *blue*, *male* or *female*, and so on; • REGRESSION – when the output variable is a *real* or *continuous* value (*regression* is about a relationship between two or more variables where a change in one of them is associated with a change in another one).

The BAYESIAN CLASSIFICATION (BC), in particular, represents a *SML* method as well as a STATISTICAL method for classification, assuming an underlying probabilistic model. BC allows us to capture uncertainty about the model in a principled way, by determining the outcomes' probabilities. The BAYESIAN THEOREM is important in this regard - it is featuring the *a-posteriori* probability that a hypothesis holds given observed data; here, the following is to be explained: • *a-priori probability*: a probability that is derived purely by deductive reasoning; • *a-posteriori probability*: the revised probability of an event occurring after taking into consideration new information. The theorem is as follows:

$$P(H|\mathbf{X}) = (P(\mathbf{X}|H)P(H))/P(\mathbf{X})$$

and for grasping "P(**X** | H)", one should be aware that it is all about HYPOTHESES (for example: *Hypothesis 1* - Person will make a holiday booking; *Hypothesis 2* - Person will not make a holiday booking) and it is also about CLASSIFYING A DATA TUPLE (for example: featuring attributes, such as age, income, and so on). Thus, P(**X** | H) is about the probability, given a particular hypothesis (for example: *Hypothesis 1*), that the "item" has those "characteristics" (the particular values of the attributes, such as age, income, and so on, as provided for classification). P(H) is just the general (*a-priori*) probability that a hypothesis occurs (for example: the probability that *Hypothesis 1* occurs). Finally, P(**X**) is called "marginal likelihood" and it is the average likelihood over a range of attribute values (for example: if we have provided a particular tuple **X** for classification, with its particular attribute values, then the marginal likelihood would be featuring the probability that a data tuple has exactly those attribute values).

The *NBCA* builds on the *Bayesian Theorem* in the sense that the goal is to predict which hypothesis is most likely to occur with regard to a data tuple – this would mean the highest value for P(H | **X**). This thus also means the highest value for P(**X** | H)P(H).

Hence, if we MAXIMIZE P(**X** | H)P(H), then we know which hypothesis will occur most likely, given the particular data tuple to be considered.

As it concerns context-aware applications, we argue that in many COMPLEX cases (cases that would often concern the "mental state" of a person), applying *NBCA* would be useful and effective.

We partially justify this claim by means of an illustrative example, presented in the following sub-section.

5.3 Exemplification

Both examples, considered in the current sub-section, are "imaginary" toy examples aiming at illustrating our motivated claims concerning the potential strengths of the Norm Analysis Method and NBCA. The first example is inspired by the AWARENESS case [11] but is directed towards an application scenario that goes beyond the AWARENESS scope, namely Disaster Management [27]. The second example is inspired by the AllElectronics Case (featured in [14]) and adapted to fit the context-awareness focus (to some extent, imaginary details have been added).

5.3.1 Norm-Driven Context-Awareness Featuring Disaster Management

Briefing: Rescue workers are discovering Nick who seems to be injured. Since rescue operations are life-critical, they cannot be dominated by the intuitive judgement of rescue workers. Instead, in what they do, rescue workers are expected to follow rigorous rules (norms). The case situation is illustrated in Fig. 5 and it is to be noted that the considered information is simplified and partial, just supposed to serve as illustration.

As it is seen from the figure, there is an injured person and there are two hypotheses (corresponding to two user situation types), namely: (i) There is an ambulance in close proximity (by this, it is meant: within 30 km); (ii) There is no ambulance available in close proximity. The rescue workers who have discovered the injured person are instructed to WAIT in the event of (i) and try to help – in the event of (ii). The idea is the following: • If there is an ambulance nearby, it would be less risky for the life and the health of the injured person to "wait" just for several minutes and then receive specialized help from the healthcare professionals who would arrive in the ambulance, rather than receiving urgent help immediately but not from healthcare professionals (because the rescue workers are not healthcare professionals even though they are trained to give first aid). • Nevertheless, with no ambulance available nearby, it might be too risky keeping the injured person wait for too long and the help provided by the rescue workers would be appreciated even though they are not healthcare professionals.

Next to that, the considered rescue procedure assumes the "30 km" as "measure" for what is to be considered as "nearby", as mentioned already.

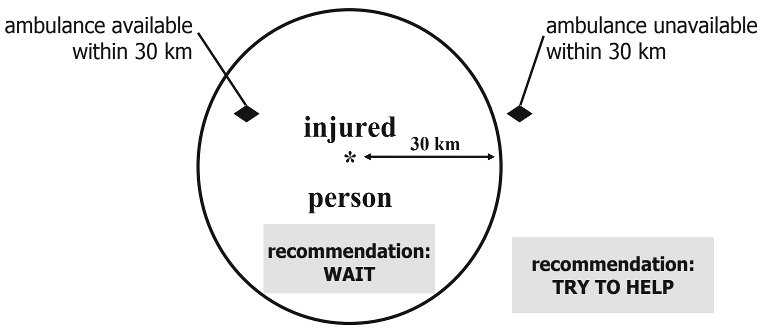

Fig. 5. Helping an injured person – featuring two user situation types

Hence, one would recognize the need for a context-aware system and exactly two user situation types are to be considered: • THE USER IS INJURED AND IT IS POSSIBLE THAT THE USER RECEIVES PROFESSIONAL HELP WITHIN "SEVERAL" MINUTES; • THE USER IS INJURED AND RECEIVING PROFESSIONAL HELP WITHIN "SEVERAL" MINUTES IS IMPOSSIBLE.

Firstly, it is obvious that sensors (+positioning technology [28, 29]) could be used for establishing the user situation. Secondly, we apply semiotic norms (featured in the

Norm Analysis Method) to specify the desired system behavior (we use the abbreviation "rw" for "rescue worker(s)" and the abbreviation "ip" for "injured person"):

```
whenever rw have discovered an ip
  if there is an ambulance nearby
  then rw
  is obliged
  to wait
```

```
whenever rw have discovered an ip
  if there is no ambulance nearby
  then rw
  is allowed
  to give first aid
```

In this way, norms help in ESTABLISHING THE USER SITUATION. Also, we may consider hierarchies of norms [17], with norms at one "level" governing norms "underneath". Then for the level beneath we bring forward the following norm:

```
whenever ip+rw are waiting for an ambulance
 if ip worsens dramatically
 then rw
 is allowed
to give first aid
```

And so on… (the rescue worker is ALLOWED, see the second and the third norms, because it is only him or her who could best decide whether (s)he has the capabilities to help and therefore (s)he cannot be "forced" to take action). That is how norms can be used for the benefit of the specification of context-aware applications. Norms are not only more exhaustive than ECA rules (as seen from the example) but they can be organized in hierarchies – something useful as a basis for analysis and design activities.

5.3.2 Prediction-Driven Context-Awareness Featuring Holiday Bookings

Briefing: The managers of an imaginary Travel Agency (TA) are interested to know about the "next" potential customer approaching TA whether it is more likely that (s)he would book a holiday package with them or not. For this they have TRAINING DATA featuring 14 persons who have approached TA in the past. Categorization was applied, concerning the following attributes: AGE (young (y), middle aged (m), senior (s)); INCOME (high (h), medium (m), low (l)); BOOKINGS – previous holiday bookings with TA (yes (y), no (n)); RATING – by this is meant "credit rating" (fair (f), excellent (e)). As mentioned already, we have used and adapted training data from an example considered in [14]. As for the training data itself, it is as follows: • 1-John-y-h-n-f; • 2-Nancy-y-h-n-e; • 3-Arnold-m-h-n-f; • 4-Eva-s-m-n-f; • 5-Richard-s-l-y-f; • 6-Kate-s-l-y-e; • 7-Sam-m-l-y-e; • 8-Dave-y-m-n-f; • 9-Sara-y-l-y-f; • 10-Tom-s-m-y-f; • 11-Boris-y-m-y-e; • 12-Ivan-m-m-n-e; • 13-Pattie-m-h-y-f; • 14-Carlos-s-m-n-e. Further, those who have done booking after approaching TA (Hypothesis 1) are: Arnold, Eva, Richard, Sam, Sara, Tom, Boris, Ivan, and Pattie; the rest have not done booking after approaching TA (Hypothesis 2). Finally, Ben, who is approaching TA is: SENIOR, of HIGH income, with NO previous bookings with TA, and his credit rating is FAIR. The **QUESTION** is whether it is more likely that Ben books a holiday package with TA or that he would not do so. Depending on the answer, TA would establish the USER SITUATION and adapt its "behavior" accordingly: if it is expected that Ben would book a package, TA would address him with a personal promotional offer.

=> We are to classify the data tuple **X** (**senior**, **high**, **no**, **fair**), needing to MAXIMIZE **P(X|$\mathbf{H_i}$)X P($\mathbf{H_i}$)**; i = 1, 2, where P(H_1) = P(books_holiday = yes), P (H_2)= P(books_holiday = no). Hence, P(H_1) = 9/14 = 0,643, P(H_2) = 5/14 = 0,357.

Further:

P(**X**|H_1)=P(**X**|books_hol.=yes) =	P(**X**|H_2)=P(**X**|books_hol.=no) =
= P(age=s|books_hol.=yes) x	= P(age=s|books_hol.=no) x
x P(inc.=h|books_hol.=yes) x	x P(inc.=h|books_hol.=no) x
x P(book.=n|books_hol.=yes) x	x P(book.=n|books_hol.=no) x
x P (cr.r.=f|books_hol.=yes) =	x P (cr.r.=f|books_hol.=no) =
= 3/9 x 2/9 x 3/9 x 6/9 = 0,016	= 2/5 x 2/5 x 4/5 x 2/5 = 0,051

Since we need to maximize **P(X|$\mathbf{H_i}$)X P($\mathbf{H_i}$)**, we should just compare (i) 0,016 x 0,643 = **0,010** and (ii) 0,051 x 0,357 = **0,018**; (ii) is bigger than (i). Thus, we point to **HYPOTHESIS 2 ($\mathbf{H_2}$: books_holiday=no)**. Hence, the classifier predicts **books_holiday=no** for tuple **X**. Said otherwise, it is more likely that Ben would NOT book a holiday package with TA.

That is how TA establishes the USER SITUATION, such that it is capable of adapting its behavior accordingly – no need to address Ben with a personal promotional offer.

6 Conclusions

This paper builds on previous research of the authors, touching upon the specification of context-aware ICT applications and inspiring a key assumption, namely: a context-aware application is to adapt its behavior depending on the USER situation (we hence abstract from considering context-aware applications whose bottom-line goal is to optimize INTERNAL processes and/or to respond to PUBLIC values). In this, we find it useful to prepare at DESIGN TIME application behavior VARIANTS for each of the user situation types that are likely to occur. There are achievements in that direction. There are also advances concerning the capturing of context data – by counting on sensors, reports, and so on. Nevertheless, we find "room for improvement" as it concerns what we do with the available context data (and how we do it), such that we effectively and precisely establish the "current" user situation. Hence, a key contribution of the current paper is our exploring the potentials of semiotic norms and Bayesian modeling in this regard. We provide a conceptual overview, an analysis, and partial exemplifications. We need a more solid validation of our findings and this we plan as future work.

References

1. Alférez, G.H., Pelechano, V.: Dynamic evolution of context-aware systems with models at runtime. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds.) MODELS 2012. LNCS, vol. 7590, pp. 70–86. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33666-9_6
2. Alférez, G.H., Pelechano, V.: Context-aware autonomous web services in software product lines. In: Proceedings of the 15th International SPLC Conference on IEEE, CA, USA (2011)
3. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum. Comput. Interact. **16**, 2 (2001)
4. Dey, A.K., Newberger, A.: Support for context-aware intelligibility and control. In: Proceedings of the SIGCHI Conference on Human Factors in Computer Systems. ACM, USA (2009)
5. Papadimitriou, G.: Future Internet: The Cross-ETP (2009). http://www.future-internet.eu/fileadmin/documents/reports/Cross-ETPs_FI_Vision_Document_v1_0.pdf. Accessed December 2011
6. Shishkov, B., Larsen, J.B., Warnier, M., Janssen, M.: Three categories of context-aware systems. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 185–202. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_12
7. Shishkov, B., Mendling, J.: Business process variability and public values. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 401–411. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_31
8. Shishkov, B., van Sinderen, M.: From user context states to context-aware applications. In: Filipe, J., Cordeiro, J., Cardoso, J. (eds.) ICEIS 2007. LNBIP, vol. 12, pp. 225–239. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88710-2_18
9. Alegre, U., Augusto, J.C., Clark, T.: Engineering context-aware systems and applications. J. Syst. Softw. 117, C (July) (2016)
10. 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
11. AWARENESS: Freeband AWARENESS Project (2008). http://www.freeband.nl
12. Cano, J., Delaval, G., Rutten, E.: Coordination of ECA rules by verification and control. In: Kühn, E., Pugliese, R. (eds.) COORDINATION 2014. LNCS, vol. 8459, pp. 33–48. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43376-8_3
13. Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, New York (2000)
14. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publ. Inc., San Francisco (2011)
15. Levin, R.I., Rubin, D.S.: Statistics for Management. Prentice Hall, Englewood Cliffs (1997)
16. Hristea, F.T.: The Naïve Bayes Model for Unsupervised Word sense Disambiguation. SpringerBriefs in Statistics. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33693-5
17. Shishkov, B.: Enterprise Information Systems, A Modeling Approach. IICREST, Sofia (2017)
18. Bunge, M.A.: Treatise on Basic Philosophy. A World of Systems, vol. 4. D. Reidel Publishing Company, Dordrecht (1979)
19. Abeywickrama, D.B.: Context-aware services engineering for service-oriented architectures. In: Bouguettaya, A., Sheng, Q., Daniel, F. (eds.) Web Services Foundations. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-7518-7_12

20. Abeywickrama, D.B., Ramakrishnan, S.: Context-aware services engineering: models, transformations, and verification. ACM Trans. Internet Technol. J. **11**(3) (2012). Article 10. ACM
21. UML: The Unified Modeling Language (2019). http://www.uml.org
22. Mitsubishi Motors: Mitsubishi Motors Corporation (2019). http://www.mitsubishi-motors.com
23. BOSAL: Bosal Group (2019). http://www.bosal.com
24. HKairport Shop: Hong Kong Airport Electronics Shop (2019). https://www.hkairportshop.com/en_hk/electronics.html
25. Shishkov, B., Mitrakos, D.: Towards context-aware border security control. In: BMSD 2016, 6th International Symposium on Business Modeling and Software Design. SCITEPRESS (2016)
26. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. M. Kaufmann Publ. Inc., San Francisco (2011)
27. Wellington J.J., Ramesh P.: Role of Internet of Things in disaster management. In: Proceedings of the International ICIIECS Conference, Coimbatore (2017)
28. Garvanov, I., Kabakchiev, H., Behar, V., Garvanova, M., Iyinbor, R.: On the modeling of innovative navigation systems. In: Shishkov, B. (ed.) BMSD 2019. LNBIP, vol 356. Springer, Cham (2019)
29. Garvanov, I, Kabakchiev, Ch., Behar, V., Rohling, H.: Experimental study of moving man detection by acoustic forward scattering radar system. In: Proceedings of the 6th ICTRS Conference. ACM, New York (2017)