Combining Context-Awareness and Data Analytics in Support of Drone Technology

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

[1] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
kivanova@math.bas.bg

[2] Faculty of Information Sciences, University of Library Studies and Information Technologies, Sofia, Bulgaria

[3] Institute IICREST, Sofia, Bulgaria
b.b.shishkov@iicrest.org

[4] Faculty of Technology, Policy, and Management, Delft University of Technology, Delft, The Netherlands
a.verbraeck@tudelft.nl

[5] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands
m.j.vansinderen@utwente.nl

Abstract. Drones performing an autonomous mission need to adapt to frequent changes in their environment. In other words, they have to be context-aware. Most current context-aware systems are designed to distinguish between situations that have been pre-defined in terms of anticipated situation types and corresponding desired behavior types. This only partially benefits drone technology because many types of drone missions can be characterized by situations that are hard to predict at design time. We suggest combining context-awareness and data analytics for a better situation coverage. This could be achieved by using performance data (generated at real-time) as training data for supervised machine learning – it would allow relating situations to appropriate behaviors that a drone could follow. The conceptual ideas are presented in this position paper while validation is left for future work.

Keywords: Drone technology · Context-awareness · Data analytics

1 Introduction

We address Unmanned Aerial Vehicles (UAV) [1] with the label "drones" in the remainder of this paper. As studied in [2–12]: (i) Drones are capable of replacing people in dangerous environments and can make use of advanced sensing capabilities allowing for situational awareness. (ii) Drones are available in different sizes – small ones can reach difficult to access places; larger drones can monitor buildings, cities, or regions for many hours in a row. (iii) Drones need to be able to adapt to changes in their environment, while performing their missions. This makes context-awareness [13, 14] relevant

B. Shishkov and A. Lazarov (Eds.): ICTRS 2022, CCIS 1730, pp. 51–60, 2022.
https://doi.org/10.1007/978-3-031-23226-8_4

to drone technology. Context awareness essentially concerns adaptive service delivery [15], for which three adaptation perspectives are possible: serving user needs, system needs, and public values [16]. Most current context-aware systems are specified to distinguish between several anticipated situation types that have been defined at design time, this leading to triggering corresponding desired behavior types [17, 18].

Nevertheless, this only partially benefits drone technology because drone missions are often carried out in difficult situations [12] and therefore they can suffer from situations that are hard to predict at design time. We suggest combining context-awareness and data analytics [19] for a better situation coverage. This could be achieved by using performance data (generated at real-time) as training data for supervised machine learning, which would allow the drone to apply appropriate behaviors in similar situations.

We refer to literature and previous work (see above) for the topics of drone technology and context-awareness, presenting our ideas on top of that. Validation is left for future work.

The remaining of the paper is structured as follows: Sect. 2 covers drone technology from a functional perspective. Section 3 presents a context-awareness conceptual model. In Sect. 4 we present our proposed conceptual framework. Finally, in Sect. 5, we discuss the framework and its limitations as well as our plans for future work.

2 Drone Technology – A Functional Perspective

Extensive literature exists about architectures for autonomous systems, with a nice overview in [2–12]. In this position paper, we mainly focus on the design choices for a drone system that relate to societal demands [20–22] and governance [23], as well as the technical capabilities of the drone [5]. In this, we view a drone as AGENT, in the

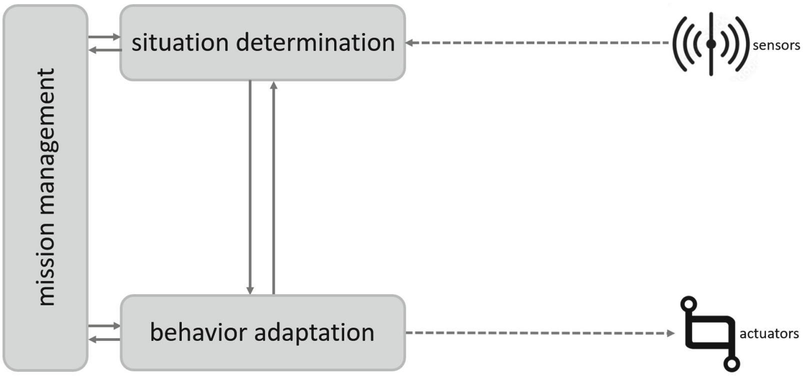

Fig. 1. Drone technology – a functional perspective

category of Multi-Agent Systems, referring to Wooldridge [24]. As such, the drone is *autonomous to some degree and adaptive, and has* ***three key features***, namely: (i) The ability to gather relevant contextual information by means of *sensing*; (ii) The ability to analyze this data (and possibly generate conclusions and/or decisions), by means of *algorithms*; (iii) The ability to adapt its behavior in *response to changes in the environment*. This is visualized (inspired by previous work [2]) in Fig. 1.

As Fig. 1 suggests, drones are essentially driven by a corresponding mission and mission management is hence crucial. It is sensitive to the "current" situation that is to be somehow determined by the drone – this is often done by means of reasoning on data from sensors. The mission management also concerns the drone's behavior adaptation. In summary, it is necessary for a drone to get relevant information (for the sake of determining the "current" situation) and be able to adapt its behavior accordingly (for the sake of delivering situation-specific services); as it concerns the former/latter, a drone would count on sensors/actuators.

3 Context-Awareness

As a problem theory for *context-aware* systems we postulate that *end-users* (*users*, for short) of *information systems* often have different needs for services provided by such systems, where different needs correspond to different context situations. As studied in [13], *context-aware* (information) systems are a "treatment" for this problem if they can provide **context-specific services to users in accordance to their context-dependent needs** [25–39]. "Context" here is the *context* of the *context-aware* system, where the former is a given (i.e., not designed) and the latter is the object of design. A *context-aware* system that is transferred to practice would interact with its context. Two kinds of interactions can be distinguished: one for *collecting data on the context* and another one - for *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. This makes perfect sense, as the *part-of* relation is an essential prerequisite for the system we want to design, viz. to make a connection between what the *context* is and what a *user* needs.

We frame the design problem with the diagram in Fig. 2. The diagram shows that a **user**, being *part of* a **context**, has *one or more* **user need**s (or sets of *user needs*), where each distinguished *user need* results from a corresponding unique **context situation**. A *context* can be conceived as a *temporal composition of one or more context situations*, where each *context situation* has a unique set of *properties* that collectively are relevant to a specific *user need*. A useful **context-aware system** is able to detect the *context situation* at hand and then offers one or more **situation-specific services** that satisfy the *needs* of the *user* being in, or experiencing, that situation.

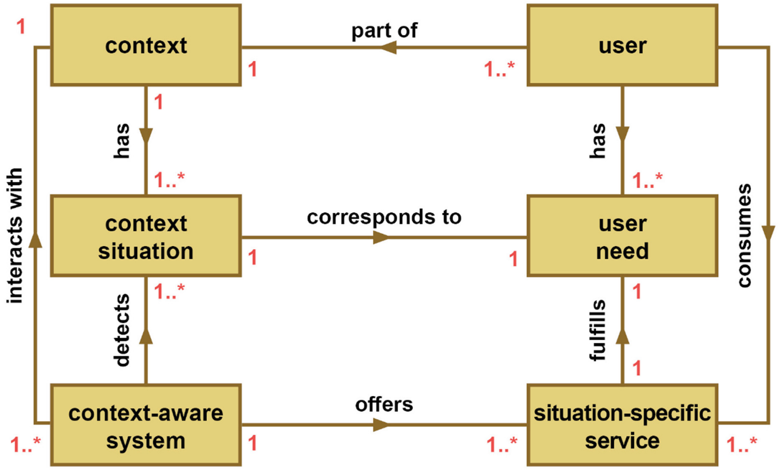

Fig. 2. Framing the problem of context-awareness (Source: [13], p. 122 ©2021, Springer, reprinted with permission)

4 Solution Directions

In the current paper, we consider ***drones*** viewed as a ***context-aware system*** (in general) and in particular – their role for the benefit of mitigations after disruptive events, such as natural disasters, pandemics, military conflicts, and so on [2], sticking to ***Systemics*** [14, 15, 21]. As visualized in Fig. 3, where the grey area stays for our system of consideration, we emphasize on *system-user interaction* (indicated at the bottom of the figure) and on the *environmental input signals*, (indicated at the top of the figure); for the sake of brevity, we omit the unavoidable reflection of system behavior to the environment. Further, taking a functional holistic perspective on a drone system, we abstract from the system duality vision assuming two overlapping systems, namely the one responsible for *motion planning* and the other one – for the achievement of concrete *goals*.

We suggest envisioning **two systems** (**SA** and **SB**) that complement each other, as inter-related parts of the drone system of consideration, as visualized in Fig. 3 (arrows indicate corresponding data flows; "***CA***" stands for "*context-awareness*" and "***DA***" stands for "*data analytics*"):

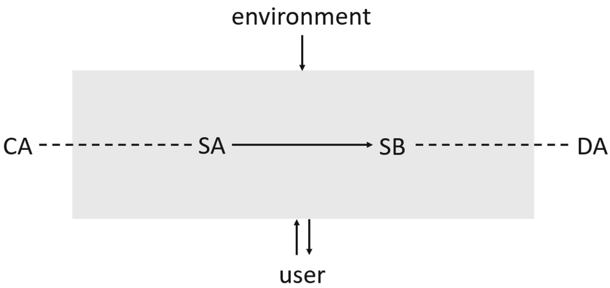

Fig. 3. A vision for combining context-awareness and data analytics

SA has been designed to distinguish between different situations that concern the user/environment, in the sense that SA is *capable of capturing a number of data values* (for example: sensor readings) whose combination points to the "current" situation type. Then SA would ***trigger accordingly a corresponding behavior type***. We may take another perspective on this: (i) There are a number of possible SITUATION INSTANCES that are recognizable by the systems and those instances are characterized by corresponding ATTRIBUTE VALUES; (ii) The different possible system behaviors are in (several) behavior CLASSES and *AS PRE-DEFINED AT DESIGN TIME*: *for each recognizable situation instance there is a corresponding desirable behavior class*. For example, there may be three behavior classes relevant to the drone-mitigation case, namely: MONITOR, BRING THINGS, and FLY BACK. As it concerns situations, there may be relevant attributes, such as (suffering) person(s) identified: Yes or No; the drone has enough power (fuel and/or battery): Yes or No; there is overall emergency: Yes or No; drone supplies are: normal, scares, or none, and so on; the identified person(s) are: in close proximity to the drone, in mid proximity, or away, and so on. Hence, depending on the values, we derive an INSTANCE TUPLE and for each *instance tuple*, SA "knows" which *behavior class* to trigger. This all is rooted in the *CA paradigm* (see the dashed line at the left side of Fig. 3 and refer to Sect. 3) and validated in many cases, such as AWARENESS [40].

We argue that using this alone would mean that we should: (i) either *spend too much time and resources during the design*, for identifying, specifying, and designing things that concern very many potential *situation instances*; (ii) or *assume high levels of risk that a situation instance would pop up that cannot be "recognized" by the system.* Systems servicing critical enterprises would count on developments that are backed by "huge" resources and (i) would then be realistic, which we nevertheless consider not the case as it concerns most current civil drone systems.

Hence, we count on ***SB***, rooted in ***DA*** (see the dashed line at the right side of the Figure), to complement ***SA*** in a useful way, considering ***SML - Supervised Machine Learning*** [19] and ***Statistics*** [41]. In particular, the "running" of ***SA*** would produce "for free" as "side effect" ***training data that would feed SB***. Since this would be *labelled data*, it would adequately let ***SB learn to dynamically compose behaviors*** for situations not recognized by ***SA***. Still, this all should "happen" at a ***lower granularity level***, assuming that *situation instances* point to particular ***attribute values*** that are considered by ***SB***.

When ***SA*** has not "recognized" a situation instance, it could at least pass to ***SB*** *attribute-values-level data* concerning what was captured. This can be used by ***SB*** to "decide" ***which behavior class to trigger***.

In referring to *SML* and *Statistics*, let us consider (for example) ***CovA*** – the ***Covering Algorithm*** and ***BCA*** – the ***Naïve Bayesian Classification Approach*** [43]. In applying *CovA*, ***SB*** uses the *training data* (in terms of a number of *tuples*, featuring *situation instances* and corresponding *behavior classes*) "inherited" by ***SA***, to generate <u>**RULES**</u> corresponding to each of the *behavior classes*. Hence, in the event of ***SA*** not recognizing a *situation instance*, it "goes" to ***SB*** which in turn applies the abovementioned *rules* to it in order to establish a match with regard to one of the *behavior classes*, triggering it accordingly. In applying *BCA*, ***SB*** similarly uses the *training data* (see above) but in a different way and restricted by the *BCA* limitation of considering maximum two *behavior classes*. Then it would be a question which of the two *behavior classes* is more adequate with regard to the *situation instance* that has not been recognized by ***SA***. To answer this question for the benefit of ***SB***, we need to apply the *Bayes Theorem* that allows for classifying a *tuple* (featuring a *situation instance*) with regard to two *behavior classes*, using the abovementioned *training data*. That is how ***SB*** would identify the relevant *behavior class* and would trigger it accordingly. And in the end, even though *Neural Networks* [42] can bring invaluable pattern-recognition-related support to drone's motion planning, we argue that methods, such as the ones considered above (and possibly also *decision-tree classifiers* [19]) are most appropriate for combining *CA* and *DA*. That is not only because the *attributes* are <u>precisely defined</u> by ***SA*** but also because <u>traceability</u> is important - the system "decisions" have to be explainable.

Further, it would be possible "<u>exporting</u>" ***SA***'s *training data* to other systems and/or "<u>importing</u>" (for the benefit of ***SB***) external *training data*. Nevertheless, for this the *training data* STRUCTURE (featured by particular *attributes*) should be the same – for example, if ***SA*** is recognizing situations, considering particular *attributes* while later ***SB*** would be covering unrecognized situations, considering other *attributes*, then the overall *quality-of-service* would be low and with limited *traceability* potential. Also, we must not forget the existence of the problem of confusion between *causality* and *coincidence*. Finally, ***SB***'s *inheriting* classification models from other *CA* (*drone*) *systems* poses the need for addressing, issues, such as *data reliability*, *data pre-processing*, *data harmonization*, and so on.

In summary, going back to the example featuring *drones used for monitoring that aims at mitigations after disruptive events* [2], we have just two *behavior classes*, namely: <u>*monitoring of people in normal health in an affected area*</u> (when just monitoring is needed), and <u>*providing support to persons needing urgent help in this area*</u>. Here, at a lower granularity level we may <u>consider values of attributes</u> featuring the *health situation*, the *area*, and so on. Hence, a *situation instance* unrecognized by ***SA*** and hence "passed" to ***SB*** would be: *a person needs urgent help outside of the affected area* (for instance – when the drone can identify some accident outside of the area which is on the focus of its mission). Then, with this not having been anticipated at design time, ***SB*** may try to identify the right *behavior class* to trigger, for example, by applying *CovA* or *BCA* (see above), possibly resulting in: *call ambulance*.

Hence, there are several things that are essential: • the (drone) system (and in particular ***SA***) should be capable of identifying situation instances; • in this, its possibilities are not unlimited, in the sense that only a limited number of situation instances (these anticipated at design time) are covered; • for each recognized situation instance, the system (and in particular ***SA***) switches to a corresponding behavior class (and this was also "prepared" at design time); • any unrecognized situation instance is to be "passed" to ***SB***; • it uses training data (in terms of tuples featuring situation instances and reflecting corresponding attribute values and a relevant behavior class) "inherited" by ***SA***; • in this, ***SB*** applies *CovA*, *BCA*, or another appropriate method for classifying the situation instance with regard to the behavior classes.

5 Discussion and Conclusions

Nowadays, drones have become indispensable helpers in many situations, from various observations to detect damage in critical infrastructures such as roads, railways, or other facilities, through monitoring featuring flooded areas, agricultural crops, pollution spots, and deforestation, to active actions such as delivering first aid kits and spraying insecticides.

A specific feature of such missions (that are often carried out in difficult situations) is the *dynamic change in environment*, and it is often *impossible to foresee all scenarios at design time*. This is specific even though not exclusively valid for drones and concerns such context-awareness-related limitations. Still, the contribution of the current paper is limited to *drone technology*.

We have addressed this technology, referring to previous work and we have superimposed this with regard to a *context-awareness* conceptualization (again referring to previous work). On that basis, we have proposed solution directions featuring the ***combined application of context-aware computing and data analytics***, and assuming a ***system duality***, as follows:

- The first one is a "classical" context-aware system, which *incorporates recognition of different situations and sets appropriate behaviors* according to algorithms established at design time.
- Unlike the classical system, whose functions end there, in the proposed architecture, *this system feeds data, characterizing situations and behaviors to a second system*; those recognizable cases will be used by the second system to *build a classification model, such that it is capable of generating rules in a cases when the first system falls into a "non-recognized" situation.*

Validating the proposed idea is left for future work.
The *limitations* of our work are considered to be three-fold:

- An explicit discussion is missing concerning the ***criteria for "calling SB"*** since it may be that a situation is not "recognized" because of sensor failures and/or data-processing-related issues.

- A ***mapping architecture*** is missing as it concerns the two granularity levels, namely: the ***SA*** granularity level featuring *situations* and *behaviors*, and the ***SB*** granularity level featuring *patterns* of both.
- The work is limited just to ***drone technology***.

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

References

1. Griffin, G.F.: The use of unmanned aerial vehicles for disaster management. Geomatica **68**(4), 265–281 (2014)
2. Shishkov, B., Branzov, T., Ivanova, K., Verbraeck, A.: Using drones for resilience: a system of systems perspective. In: Proceedings of the 10th International Conference on Telecommunications and Remote Sensing (ICTRS 2021), New York, NY, USA. Association for Computing Machinery (2021)
3. Shishkov, B., Hristozov, S., Verbraeck, A.: Improving resilience using drones for effective monitoring after disruptive events. In: Proceedings of the 9th International Conference on Telecommunications and Remote Sensing (ICTRS 2020), New York, NY, USA. Association for Computing Machinery (2020)
4. Shishkov, B., Hristozov, S., Janssen, M., Van Den Hoven, J.: Drones in land border missions: benefits and accountability concerns. In: Proceedings of the 6th International Conference on Telecommunications and Remote Sensing (ICTRS 2017), New York, NY, USA. Association for Computing Machinery (2017)
5. Milas, A.S., Cracknell, A.P., Warner, T.A.: Drones – the third generation source of remote sensing data. Int. J. Remote Sens. **39**(21), 7125–7137 (2018)
6. Kayan, H., Eslampanah, R., Yeganli, F., Askar, M.: Heat leakage detection and surveillance using aerial thermography drone. In: Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU) (2018)
7. Pandey, S., Barik, R.K., Gupta, S., Arthi, R.: Pandemic drone with thermal imaging and crowd monitoring system (DRISHYA). In: Tripathy, H.K., Mishra, S., Mallick, P.K., Panda, A.R. (eds.) Technical Advancements of Machine Learning in Healthcare. SCI, vol. 936, pp. 307–325. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-4698-7_15
8. Hill, A.C., Laugier, E.J., Casana, J.: Archaeological remote sensing using multi-temporal, drone-acquired thermal and Near Infrared (NIR) imagery: a case study at the Enfield Shaker Village, New Hampshire. Remote Sens. **12**(4), 690 (2020)
9. Carrio, A., Sampedro, C., Rodriguez-Ramos, A., Campoy, P.: A review of deep learning methods and applications for unmanned aerial vehicles. J. Sens. **2017**, Article ID 3296874 (2017)
10. Erdelj, M., Natalizio, E., Chowdhury, K.R., Akyildiz, I.F.: Help from the sky: leveraging UAVs for disaster management. IEEE Pervasive Comput. **16**(1), 24–32 (2017)

11. Kopardekar, P., Rios, J., Prevot, Th., Johnson, M., Jung, J., Robinson III, J.E.: Unmanned aircraft system traffic management (UTM) concept of operations. In: Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference, Washington, D.C., USA (2016)
12. American Red Cross: Drones for Disaster Response and Relief Operations (2015). https://www.issuelab.org/resources/21683/21683.pdf
13. 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
14. 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. (eds.) BMSD 2022. LNBIP, vol. 453, pp. 240–248. Springer, Cham (2022) https://doi.org/10.1007/978-3-031-11510-3_15
15. 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
16. 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
17. 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
18. 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
19. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publ. Inc., San Francisco (2011)
20. Dietz, J.L.G.: Enterprise Ontology, Theory and Methodology. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-33149-2
21. Bunge, M.A.: Treatise on Basic Philosophy. A World of Systems, vol. 4. D. Reidel Publishing Company, Dordrecht (1979)
22. 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
23. van Veenstra, A.F., Janssen, M., Tan, Y.H.: Towards an understanding of E-government induced change – drawing on organization and structuration theories. In: Wimmer, M.A., Chappelet, J.L., Janssen, M., Scholl, H.J. (eds.) EGOV 2010. LNCS, vol. 6228, pp. 1–12. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14799-9_1
24. Bordini, R.H., Fisher, M., Wooldridge, M., Visser, W.: Model checking rational agents. IEEE Intell. Syst. **19**(5), 46–52 (2004)
25. 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), 97–166 (2001)
26. Silvander, J.: On context frames and their implementations. In: Shishkov, B. (ed.) BMSD 2021. LNBIP, vol. 422, pp. 133–153. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79976-2_8
27. 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)
28. 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

29. Alegre, U., Augusto, J.C., Clark, T.: Engineering context-aware systems and applications. J. Syst. Softw. **117**(C), 55–83 (2016)
30. Alférez, G.H., Pelechano, V.: Context-aware autonomous web services in software product lines. Proceedings of the 15th International SPLC Conference, CA, USA. IEEE (2011)
31. Abeywickrama, D.B., Ramakrishnan, S.: Context-aware services engineering: models, transformations, and verification. ACM Trans. Internet Technol. J. **11**(3), Article 10 (2012)
32. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: First Workshop on Mobile Computing Systems and Applications, pp. 85–90. IEEE (1994)
33. Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P.: The anatomy of a context-aware application. Wirel. Netw. **8**, 187–197 (2002)
34. 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)
35. Abecker, A., Bernardi, A., Hinkelmann, K., et al.: Context-aware, proactive delivery of task-specific information: the KnowMore project. Inf. Syst. Front. **2**, 253–276 (2000)
36. 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)
37. Chaari, T., Laforest, F., Celentano, A.: Adaptation in context-aware pervasive information systems: the SECAS project. Int. J. Pervasive Comput. Commun. **3**(4), 400–425 (2007)
38. Pawar, P., Van Beijnum, B., Hermens, H., Konstantas, D.: Analysis of context-aware network selection schemes for power savings. In: Proceedings of the Asia-Pacific Services Computing Conference, pp. 587–594. IEEE (2008)
39. Van Engelenburg, S.: Designing context-aware architectures for business-to-government information sharing. Ph.D. thesis. TU Delft Press (2019)
40. Wegdam, M.: AWARENESS: a project on context AWARE mobile NEtworks and ServiceS. In: Proceedings of the 14th Mobile & Wireless Communications Summit. EURASIP (2005)
41. Levin, R.I., Rubin, D.S.: Statistics for Management. Prentice Hall, Englewood Cliffs (1997)
42. Wasserman, T., Wasserman, L.: Motivation, effort, and neural network modeling: implications. In: Wasserman, T., Wasserman, L. (eds.) Motivation, Effort, and the Neural Network Model. NNMAI, pp. 145–160. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58724-6_12
43. 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