A Dynamic Contextual Change Management Application for Real Time Decision-Making Support

Widad Es-Soufi(✉), Esma Yahia, and Lionel Roucoules

Arts et Métiers ParisTech, CNRS, LSIS, 2 cours des Arts et Métiers,
13617 Aix en Provence, France
{Widad.ES-SOUFI,Esma.YAHIA,Lionel.ROUCOULES}@ensam.eu

Abstract. Decision making is a fundamental process within organizations for many reasons. It is indeed involved at all levels (new product decisions, management and marketing decisions, etc.) and has a direct impact on companies' efficiency and effectiveness. Many researches are conducted to enhance the decision-making process by proposing decision support systems where the most frequent challenge is the change management. Indeed, all businesses operate within an environment that is subject to constant changes (like new customers' needs and requirements, organisational and technological changes, changes in key information used to derive decisions, etc.). These changes have a major impact on the quality and accuracy of the proposed decision if they are not detected and propagated, at the right time, during the decision-making process. The present work attempts to resolve this challenge by proposing a dynamic change management technique that allows three tasks to be automatically performed. First, continuously detect changes and note them. Second, retrieve from the detected changes those that are related to the decision rules. Finally, propagate them by computing the new value of the decision rule. The proposal has been fully implemented and tested in the supervision process of gas network exploitation.

Keywords: Change management · Dynamic change propagation
Decision making · Process patterns · Business process

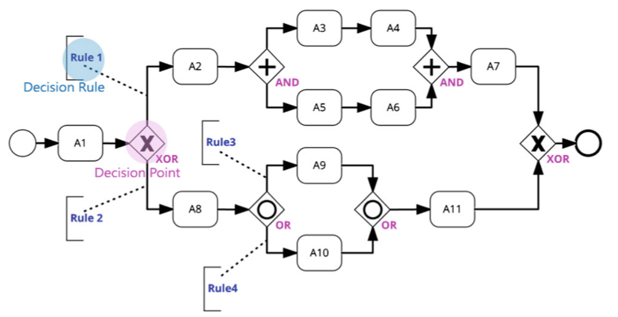
1 Introduction

The 21st century is characterized by rapid changes that are due to the technological progress, as well as the economic, political and social changes. In this context, it becomes critical that organizations perform two tasks. The first one is to develop capabilities to implement changes in order to improve their competitiveness and sustain their business growth. The second task consists in informing the decision-making process of these changes. Indeed, the latter is considered as one of the most complex and critical processes in an organization for three reasons among others. First, it is the key function to a company's success where any decision made, no matter how simple it may be, has a significant impact on the other processes and thus on the business performance [1]. Second, it is highly constrained by the decision context, that is often

Published by Springer Nature Switzerland AG 2018.
P. Chiabert et al. (Eds.): PLM 2018, IFIP AICT 540, pp. 759–767, 2018.
https://doi.org/10.1007/978-3-030-01614-2_69

subject to several changes which need to be considered to make the right decision. Third, it sometimes involves several teams (subject matter experts, business analysts, engineers, etc.) that collectively analyse problems, evaluate alternatives and decide.

Many researches focus on supporting engineers by developing decision supports systems (DSS), which are computational tools that enable decision makers to select, at each time, the best alternative among a set of alternatives that are sometimes contradictory. After the appearance of artificial intelligence and learning techniques, the concept of intelligent decision support system (IDSS) arose. It is a DSS that provides direct access to expertise by supporting learning in the decision-making process [2]. In [3], authors propose a process mining based IDSS that performs two tasks. First, it analyses the past process executions, using process mining techniques, to identify and generate the different process patterns as well as the decision points and the rules that control them (cf. Fig. 1). Second, it filters the detected patterns and propose a ranking of the ones that fit the decision context.



Process patterns:

P_1: $A_1A_2A_3A_4A_5A_6A_7$

P_2: $A_1A_2A_3A_5A_4A_6A_7$

P_3: $A_1A_2A_3A_5A_6A_4A_7$

P_4: $A_1A_2A_5A_6A_3A_4A_7$

P_5: $A_1A_2A_5A_3A_6A_4A_7$

P_6: $A_1A_2A_5A_3A_4A_6A_7$

P_7: $A_1A_8A_9A_{11}$

P_8: $A_1A_8A_{10}A_{11}$

P_9: $A_1A_8A_9A_{10}A_{11}$

Fig. 1. The identified process patterns, decision points and decision rules

In the present paper, authors focus on developing an executable application whose objective is to inform process mining based IDSSs of the changes that dynamically occur during the decision-making process. It updates the decision context whenever a change is detected and automatically filters the patterns that are generated from the IDSS based on the decision rules, also generated from the IDSS. Authors assume that, no matter how many changes occur, the developed application helps to make the right decision given a context at time t_i, where i goes from 0 (start of the decision-making process) to n (end of the process). Furthermore, authors assume that the application enables innovation by discovering new patterns from the already existing ones.

This paper is organized as follows. In Sect. 2, a literature review on change management during the decision-making process is presented and discussed. In Sect. 3, the proposed application is presented, then tested and verified in an industrial case study in Sect. 4. Finally, the future work is discussed before concluding the paper.

2 Contextual Change Management During Decision Making

Change management is the process in which a change is identified, analysed, evaluated, planned and propagated [4]. Three major types of change are identified in literature. First, the change that comes before making a decision such as an organisational change, technological change, etc. [5]. This type of change helps companies to remain relevant by keeping up with technological advances and changes in the marketplace and is generally planned and evaluated before deciding to successfully implement it. The second type is the product change management also called engineering design change [6]. This type concerns the modifications that are applied to products to meet the clients' needs among others and is, thus, linked to the first change type. The product change management consists in propagating the changed (i.e. new) product data throughout the product lifecycle. The third type is the contextual change management [7]. It consists in predicting, identifying or managing the context changes for successfully addressing dynamic (also called real time) decision making support. Indeed, in dynamic environments, only real time contextual data can provide decision makers with the accurate knowledge that is necessary to make decisions [8]. In the present paper, authors are interested in the contextual change management since it responds to their objective, which is to inform the IDSS of the changes that affect the decision-making context. Therefore, only the last type of change is explored and discussed in the following sub-sections.

2.1 Literature Review

A decision is defined by [9] as an irreversible commitment to a proposition or a plan of action based on evidence, the expected metrics associated with the outcome and the context in which the decision is being made. The context is an essential and powerful concept for effective real time decision-making. The context definition that is adopted by most researchers is that it is "any information that can be used to characterise the situation of an entity" [10]. This context information can be imported from sensors, derived from other information, reasoned, computed or explicitly entered by users [7]. There are two categories of context [11]. The dynamic context that consists of dynamic information such as heart rate, pressure data streams and the static context where information are static such as decision makers names, etc. [7].

There are many approaches in literature that focus in understanding the behaviour of the decision-making context. Some approaches intend to model the context and are based on Bayesian reasoning techniques [12], fuzzy logic [11], graphical modelling [13] and some other ontology-based modelling approaches that allow a formal representation of concepts and their relationships that is used for knowledge reasoning about context [14]. In [15], authors propose an overview of the context using the user behavior modelling. In addition to the modelling based approaches, there are other approaches that intend to predict the future context, in order to improve the contextual change management, by using data mining techniques such as classification, neural networks, and statistics-based techniques [7]. Other approaches use expert systems and rule engines for context prediction [7].

2.2 Discussion

Most of the approaches explored in the precedent sub section are related to the context modelling or prediction. these techniques face some challenges that concern, among others, the prediction of human actions that may be part of the real time decision context [16]. The latter, as defined in [7], is a representation of the environment as a set of contextual data which provide the decision maker with the necessary knowledge to decide as soon as these data are obtained. In authors' context, these contextual data are both static and dynamic and are not always pre-defined or inferred since some of them change constantly, will be known only in real time or interfere with other contexts (e.g. A station's pressure may depend on the other surrounding stations behaviour, as well as on its lifetime and/or the climate change). In addition, authors need to express the context in a deterministic way, since it is used to filter the end-to-end patterns based on some decision rules in the form "if (pressure >25) then send an emergency technician", here, a precise value of the pressure (i.e. 25) is needed rather than the distribution or prediction of pressure values. To achieve their objective, authors need continuous insights on the environmental changes and explicit knowledge of the current context. Therefore, they assume that the explored approaches cannot be used to achieve their objective and propose in the next section an application that, based on the contextual data that are obtained on the move, aims at supporting real time decision making in an informed way.

3 Proposal: The Dynamic Change Management Application

The workflow presented in Fig. 2 is the overview of the proposal that enables the current objective to be achieved.

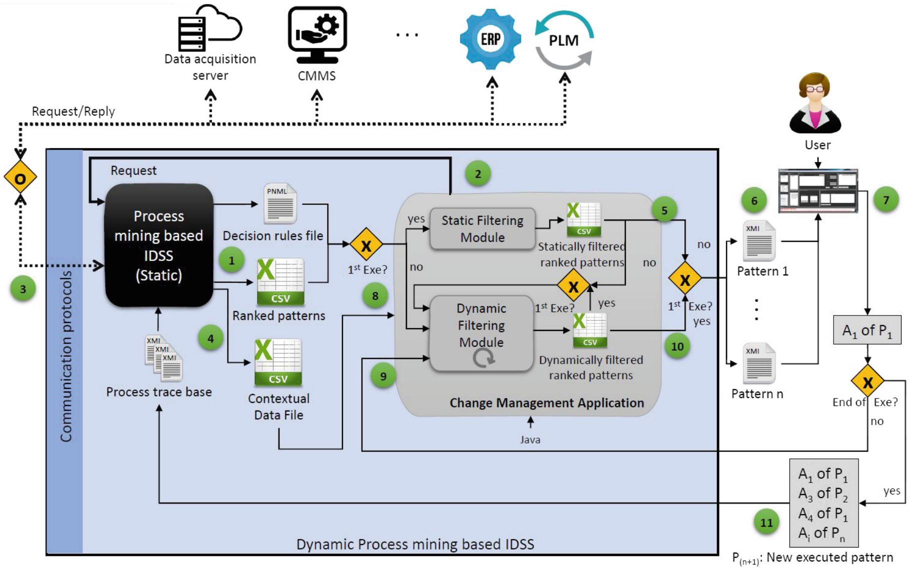


Fig. 2. Workflow of the proposal (Expressed in BPMN(http://www.bpmn.org/))

The first step consists in executing the static IDSS to generate the “decision rules” and the “ranked patterns” files from the process trace base as explained in [3]. This base contains the past process executions, also called traces, that were generated with respect to the trace metamodel identified in [17]. The “decision rules” file contains the rules that control how the process flows and is expressed in PNML (Petri Net Markup Language) format. The latter is an XML based standardized interchange format for Petri nets that allows the decision rules to be expressed as guards, this means that the transition from a place (i.e. activity) to another, can fire only if the guard, and thus the rule, evaluates to true. For instance, rule 1 in Fig. 1 is the decision rule that enables the transition from A_1 to A_2. The second document, generated by the static IDSS, is the “ranked patterns file”. It contains the distinct end-to-end process patterns (e.g. patterns P_1 to P_9 in Fig. 1), that were identified in the process trace base and sorted by frequency.

Once the two files are generated by the IDSS, and if it is its first execution for a given decision problem, the static filtering module is invoked. This latter requests the current context from the IDSS (step 2) which transmits the request to the connected systems via the communication layer (step3). This layer is beyond the scope of the present paper, authors aspire to present it in detail in another article. Among the systems that are connected to the IDSS, authors cite the data acquisition server that collects data from the different field devices (e.g. pressure sensors), the CMMS (Computerized Maintenance Management System) that delivers maintenance data such as the date of failure/repair, the ERP (Enterprise Resource Planning) that delivers the operational processes information (e.g. delivery date) and PLM software (Product Lifecycle Management) that contains all the product/process information throughout the product lifecycle (e.g. design data).

Once the IDSS receives the context from the connected systems, it copies it in a “contextual data” file which is sent to the static filtering module (step 8). The execution of this module generates the “statically filtered ranked patterns” file (step5) that associates to each pattern, of the “ranked patterns” file, a decision rule flag whose value is either 1 or 0. 1 means that the pattern satisfies the decision rules in the current context’s conditions and is, thus, ready to be proposed to the user (step 6) via the IDSS graphical user interface presented in [18].

The user chooses one of the proposed patterns and starts to execute its first activity (step 7). If the executed activity does not refer to the end of the process (i.e. “activityEvent $\neq$ end” as explained in [17]), then both the activity and the pattern to which it belongs are sent to the dynamic filtering module (step 9). The latter requests the context at time t (execution of the steps 2, 3, 4, 8), and takes as inputs the “contextual data” and the “statically filtered ranked patterns” files as well as the last executed activity and its pattern in order to update the statically filtered ranked patterns to a new file called “Dynamically filtered ranked patterns”, where the decision rule flag is updated according to whether the decision rules are satisfied by the new current context. If the latter satisfies the rule 1 (cf. Fig. 1) for example, the activity A_2, and all the patterns that contain it (P_1 to P_6) are updated to 1 while the patterns (P_7 to P_9) are updated to 0.

The patterns whose decision rule flag equals to 1 are proposed to the user based on their frequency order (step 6). Then the steps 7, 9, 2, 3, 4, 8 are repeated in a loop until the last executed activity refers to the end of the process (i.e. “activityEvent = end” as explained in [17]). In this case, the new pattern that contains all the executed activities is sent to the process trace base to enrich it (step 11). Note that the new executed pattern is sent to the base even if it is one of the existing patterns, since the component that generates the “ranked patterns” selects only the distinct patterns and associates to each one the number of its occurrence (frequency).

The proposed change management application enables the IDSS, developed in [3], to be dynamically aware of context changes. It is fully implemented using the Java programming language.

4 Case Study: Supervision of Gas Network Exploitation

As discussed in [3], systems supervision is a business process that consists of complex decision based activities. During this process, supervisors have to take actions (i.e. decisions) depending on both the supervision result and the set-point (i.e. target value for the supervised system) in order to maintain the well-functioning of an industrial process, in case an alarm is received or when the performance target is seeming far from being met. The authors prefer to reuse the same supervision case study, presented in [3], where the supervisor of an industrial process should take, in the shortest time, the right decision in case an alarm is received. The IDSS is intended to support this supervisor during his problem-solving process by proposing the right decision even if the context (that can be described by the field sensors values, season, supervisor’s name, etc.), changes considerably and quickly (e.g. the pressure of a gas station at time t_i may be very different from the pressure at time t_{i+1}).

The problem-solving process starts by receiving the malfunction alarm. The IDSS identifies the decision points and rules (c.f. Fig. 3) as well as the different end-to-end patterns and export them to two files: The decision points/rules file in pnml format and the ranked patterns file in csv format [3].

After encountering the first gateway, the IDSS asks the connected systems for the context at time t_1 (step 2,3 in Fig. 2) based on which it statically filters the ranked patterns by filling the decision rule column to 1 if the pattern satisfies the rule, 0 otherwise (step 5 in Fig. 2). All the patterns are then displayed in the IDSS interface, based on their frequency, where those whose decision rules flags equal to 1 are printed in dark black and can be selectable while the other patterns are displayed in grey but can also be selectable as the IDSS aims at supporting the user, not restricting his choices. Therefore, as depicted in Fig. 4(a), the pattern “P4T1John” is the only one that satisfies the context at time t_1 (cf. Table 1 and rules in Fig. 3) and is selected by the user. The latter starts to execute the next proposed patterns activities, namely, “Send an emergency technician” and “Measure the flow rate”. Before proposing these activities, the IDSS does not ask for any context information since it does not encounter any gateway, where one choice among others must be made. After the activity “Measure the flow rate” is executed, the IDSS asks for the context at time t_2, according to which it filters the patterns. In this case, the pattern that satisfies the new context is still the same (P4T1John) and the IDSS continues proposing its next activity “Simulate the pressure setpoint decrease”. Once this latter is executed, the IDSS asks for the context at time t_3, based on which it filters the patterns. The new pattern that satisfies the rules is henceforth “P3T1John” as depicted in Fig. 4(b).

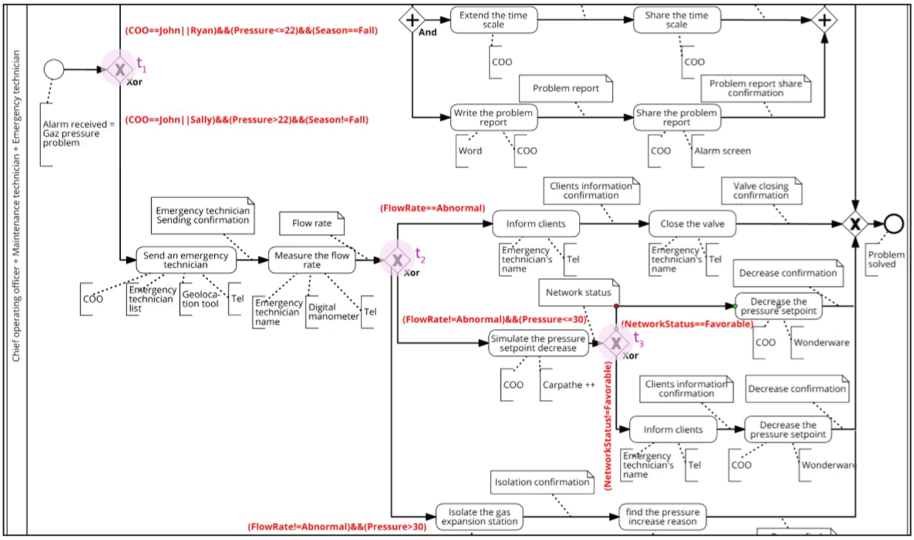
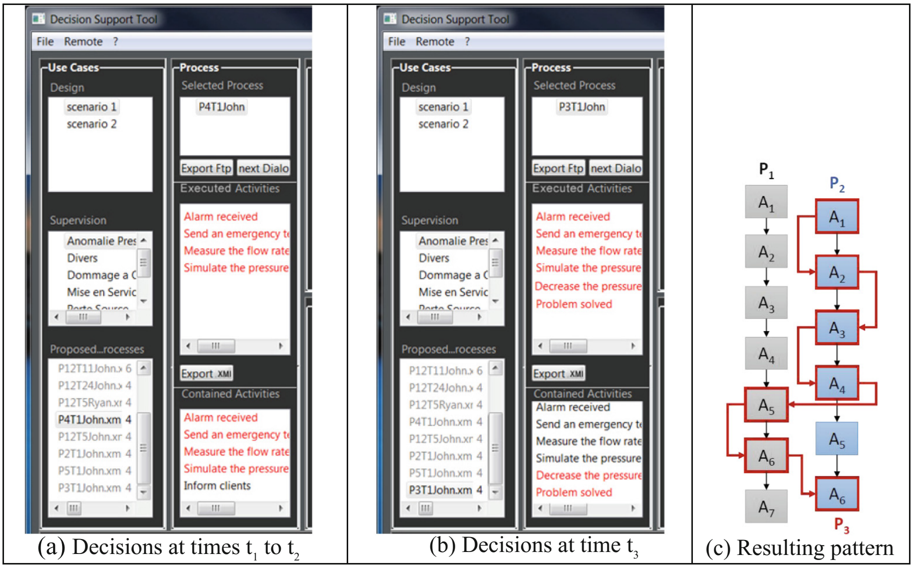

Chief operating officer + Maintenance technician + Emergency technician
Alarm received = Gaz pressure problem
t1
Xor
(COO==John||Ryan)&&(Pressure<=22)&&(Season==Fall)
(COO==John||Sally)&&(Pressure>22)&&(Season!=Fall)
And
Extend the time scale
COO
Share the time scale
COO
Problem report
Problem report share confirmation
Write the problem report
Word
COO
Share the problem report
COO
Alarm screen
Emergency technician Sending confirmation
Flow rate
Send an emergency technician
COO
Emergency technician list
Geolocation tool
Tel
Measure the flow rate
Emergency technician name
Digital manometer
Tel
t2
Xor
(FlowRate==Abnormal)
Clients information confirmation
Valve closing confirmation
Inform clients
Emergency technician's name
Tel
Close the valve
Emergency technician's name
Tel
Problem solved
Decrease confirmation
Network status
Decrease the pressure setpoint
(FlowRate!=Abnormal)&&(Pressure<=30)
(NetworkStatus==Favorable)
Simulate the pressure setpoint decrease
t3
Xor
COO
Wonderware
COO
Carpathe ++
Clients information confirmation
Decrease confirmation
(NetworkStatus!=Favorable)
Inform clients
Decrease the pressure setpoint
Emergency technician's name
Tel
COO
Wonderware
Isolation confirmation
Isolate the gas expansion station
find the pressure increase reason
(FlowRate!=Abnormal)&&(Pressure>30)

Fig. 3. Part of the supervision process with the identified decision points and rules



Selected Process: P4T1John

Executed Activities:
Alarm received
Send an emergency te
Measure the flow rate
Simulate the pressure

Proposed...rocesses:
P12T11John.x 6
P12T24John.x 4
P12T5Ryan.xr 4
P4T1John.xm 4
P12T5John.xr 4
P2T1John.xm 4
P5T1John.xm 4
P3T1John.xm 4

Contained Activities:
Alarm received
Send an emergency te
Measure the flow rate
Simulate the pressure
Inform clients

(a) Decisions at times t_1 to t_2

Selected Process: P3T1John

Executed Activities:
Alarm received
Send an emergency te
Measure the flow rate
Simulate the pressure
Decrease the pressure
Problem solved

Proposed...rocesses:
P12T11John.x 6
P12T24John.x 4
P12T5Ryan.xr 4
P4T1John.xm 4
P12T5John.xr 4
P2T1John.xm 4
P5T1John.xm 4
P3T1John.xm 4

Contained Activities:
Alarm received
Send an emergency te
Measure the flow rate
Simulate the pressure
Decrease the pressure
Problem solved

(b) Decisions at time t_3

(c) Resulting pattern

Fig. 4. Real time switching from one pattern to another while supporting decision making

Table 1. Contextual data at times t_1, t_2 and t_3

	t_1	t_2	t_3
Supervisor	John	John	John
Season	Winter	Winter	Winter
Pressure	25	28	28
Flow rate	Abnormal	Normal	Normal
Network status	Unfavourable	Unfavourable	Favourable

The IDSS proposes the activity "inform clients" and repeats all the steps until the activity "Problem solved", that refers to the end of the process (activityEvent=end), is met. Note that the user may refuse to execute the proposed pattern and start to execute new activities. In this case the IDSS listens and traces the full executed pattern in order to send it to its process trace base to enrich it.

5 Conclusion and Future Work

The objective of this paper is to inform process mining based IDSSs (Intelligent Decision Support Systems) of the changes that dynamically occur when supporting engineers during their decision-making processes. Through the proposed technique, the IDSS starts by statically filtering the identified end-to-end patterns, given the context at the beginning of the decision-making process, to fit the identified decision rules. Then the filtered patterns are proposed to the user who chooses one and starts to execute its activities. The IDSS requests the new context whenever a decision point is encountered to make sure that the patterns filtering takes account of the correct context, and thus propose the effective decision. The proposed technique is generic and can be used by every IDSS that uses process mining techniques to learn from data to infer decisions. Authors assume that the use of the process mining based IDSS along with the proposed change management application, promote innovation within organizations, since it helps to create new patterns (way of thinking or solving a problem) from several existing ones (Fig. 4(c)). Future work consists in testing this assumption and evaluate the IDSS using a real-world design problem. It also consists in continuing the development of the IDSS' interface to make it more interactive and to enable the user to add his own activities to the proposed pattern and/or trace his new executions.

Acknowledgments. This research takes part of a national collaborative project (Gontrand) that aims at supervising a smart gas grid. Authors would like to thank the companies REGAZ, GDS and GRDF for their collaboration.

References

1. Blenko, M., Mankins, M., Rogers, P.: Decision Insights: The Five Steps to Better Decisions. Bain & Company, Boston (2013)
2. Arnott, D., Pervan, G.: Eight key issues for the decision support systems discipline. Decis. Support Syst. **44**(3), 657–672 (2008)
3. Es-Soufi, W., Yahia, E., Roucoules, L.: A process mining based approach to support decision making. In: Ríos, J., Bernard, A., Bouras, A., Foufou, S. (eds.) PLM 2017. IAICT, vol. 517, pp. 264–274. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72905-3_24
4. Crnkovic, I., Asklund, U., Dahlqvist, A.P.: Implementing and Integrating Product Data Management and Software Configuration Management. Artech House, Norwood (2003)
5. Helfat, C.E., et al.: Dynamic Capabilities: Understanding Strategic Change in Organizations. Wiley, New York (2009)
6. Ullah, I., Tang, D., Yin, L.: Engineering product and process design changes: a literature overview. Procedia CIRP **56**, 25–33 (2016)
7. Burstein, F., Brézillon, P., Zaslavsky, A.: Supporting real-time decision making: the role of context in decision support on the move, vol. 13. Springer, Boston (2011)
8. Delic, K.A., Douillet, L., Dayal, U.: Towards an architecture for real-time decision support systems: challenges and solutions. In: Proceedings 2001 International Database Engineering and Applications Symposium, pp. 303–311 (2001)
9. Resulaj, A., Kiani, R., Wolpert, D.M., Shadlen, M.N.: Changes of mind in decision-making. Nature **461**, 263 (2009)
10. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. **5**(1), 4–7 (2001)
11. Delir Haghighi, P., Krishnaswamy, S., Zaslavsky, A., Gaber, M.M.: Reasoning about context in uncertain pervasive computing environments. In: Roggen, D., Lombriser, C., Tröster, G., Kortuem, G., Havinga, P. (eds.) EuroSSC 2008. LNCS, vol. 5279, pp. 112–125. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88793-5_9
12. Fox, V., Hightower, J., Liao, L., Schulz, D., Borriello, G.: Bayesian filtering for location estimation. IEEE Pervasive Comput. **2**(3), 24–33 (2003)
13. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: Mattern, F., Naghshineh, M. (eds.) Pervasive 2002. LNCS, vol. 2414, pp. 167–180. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45866-2_14
14. Truong, B.A., Lee, Y.-K., Lee, S.-Y.: Modeling uncertainty in context-aware computing. In: Fourth Annual ACIS International Conference on Computer and Information Science (ICIS 2005), pp. 676–681 (2005)
15. Ziebart, B.D., Maas, A.L., Dey, A.K., Bagnell, J.A.: Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In: Proceedings of the 10th International Conference on Ubiquitous Computing, Seoul, Korea, pp. 322–331 (2008)
16. Nurmi, P., Martin, M., Flanagan, J.A., et al.: Enabling proactiveness through context prediction. In: Proceedings of the Workshop on Context Awareness for Proactive Systems, Helsinki, vol. 53 (2005)
17. Es-Soufi, W., Yahia, E., Roucoules, L.: Collaborative design and supervision processes meta-model for rationale capitalization. In: Eynard, B., Nigrelli, V., Oliveri, M.S., Peris-Fajarnes, G., Rizzuti, S. (eds.) Advances on Mechanics, Design Engineering and Manufacturing. Lecture Notes in Mechanical Engineering, pp. 1123–1130. Springer, Cham (2017) https://doi.org/10.1007/978-3-319-45781-9_112
18. Roucoules, L., Yahia, E., Es-Soufi, W., Tichkiewitch, S.: Engineering design memory for design rationale and change management toward innovation. CIRP Ann. Manuf. Technol. **65**(1), 193–196 (2016)