




A Method to Make an Existing System Adaptive

Francesca Gullà , Alessandra Papetti, Roberto Menghi, and Michele Germani

Department of Industrial Engineering and Mathematical Sciences,
Università Politecnica delle Marche, Via Brecce Bianche, 12, 60131 Ancona, Italy
{f.gulla, a.papetti, r.menghi, m.germani}@univpm.it

Abstract. The Adaptive systems are becoming essential into our daily life, thanks to the fast improvement of computing technology and the deployment of Internet of Things (IoT) devices. Despite many efforts have been made to improve adaptive systems design methods they remain very heterogeneous and mostly limited to each domain of application. Moreover, the most of existing adaptive applications propose specific approaches for the development of new systems, without considering the opportunity to convert existing smart systems into adaptive systems. To overcome this limitation, this paper introduces an approach able to support the designers in adapting an already existing system. To this purpose, a new design method, consisting of the three following steps, has been developed: (1) context analysis; (2) adaptive module design; and (3) system re-design. The described approach is applied, as an example, to a Wi-Fi system for the control and management of household appliances developed by a large Italian company.

Keywords: Adaptive system · Adaptive interface · Methodology
Smart environment · Universal design · IoT

1 Introduction

Nowadays, the smart systems are becoming essential into our daily life, thanks to the fast improvement of computing technology and the deployment of Internet of Things (IoT) devices. The success of the smart systems is tightly linked to features such as easy handling and user-friendly graphical interface (GUI). Primarily designing GUIs is an important step in the development of smart systems, as it allows the user to interact effectively and mutually with the intelligent system.

Adaptive Interface (AI) represents a new way to improve the GUI usability and to help the user to carry out their tasks efficiently and enjoyable [1]. Indeed, the AI adapts and customizes automatically its features and structure to the user characteristics, to facilitate users with different skill levels and impairments [2]. In order to make the user experience more enjoyable and efficient, AI is designed and employed in many different contexts: for example, in domestic environments [3, 4], in workplaces application [5, 6] and healthcare domain [7, 8]. However, all these applications propose specific approaches for the development of new systems, without considering the opportunity to convert existing smart systems into adaptive systems.

For this reason, this research work aims to provide a method able to support the designers in adapting an already existing system. In particular, starting from existing methodologies developed for new applications [9, 10], a new design method, consisting of the three following steps, has been developed: (1) context analysis; (2) adaptive module design; and (3) system re-design.

2 Research Background

Adaptive systems are entities that adapt their displays and available actions to the user's current goals and abilities, by monitoring user status, the system state and the current situation" [11]. Benyon and Murray [12] define Adaptive Systems as "systems which can automatically alter aspects of their functionalities and/or interface in order to accommodate the differing needs of individuals or groups of users and the changing user needs over time". To achieve this, such systems have to learn the context in which they evolve and to adapt their interfaces and interaction modalities to communicate with users. In detail, an adaptive interface is defined as the collection of software system, algorithms and graphical interface that can modify the data representation, the way of interaction, and the assignment of tasks to be performed as a function of the ability of the user, the state of the system, and the environment. The research literature describes many approaches that can be used to design an adaptive interface.

Dieterich et al., defines the adaptation strategy as a decision-making process, which answer to the following questions [13]: Where? Why? What? To what? How? The first attribute identifies the application areas in which the adaptive system can be helpful; the second attribute identify the reason to make an adaptive system: depending on the context and on the system type, the "adaptation goals" concern a search goal, a problem-solving or learning goal [14]; the third attribute refers to what aspects are subject to adaptation in a particular system; in detail, what features undergo a change depending on which users: "adaptation constituents" [11]; the fourth attribute refers to what aspects of the user consider to make adaptation: in general, the adaptation depends on the user's characteristics and preferences, on the user's goal, on the user's background and on the utilization context [15]; the fifth attribute identifies how to make adaptation through the definition a set of rules "adaptation rules" [14]; finally, an additional dimension has been added to those highlighted by Benyon and Murray: when to adapt? The adaptation can be carried out in run-time (during the use of the system) or statically (at the first access of the system) [12]. Stephanids et al., define these attributes as "adaptation determinants" [14]. Liu et al., design a dynamic, seamlessly personalized adaptive user interface through the use of an approach based on episodes identification and association (EIA) [16].

Namgoong et al. propose an adaptive user controller exploiting the representation of devices through semantic descriptions [17]; Yen and Acay introduce a generalized framework for the adaptation of user interface in complex environments [18]; new framework to design an AUI in the Web Application has been proposed [19–21]; Bongartz et al., define a new architecture for the AUI design in the Smart environment context [22]; Song and Cho provide a new AUI through the use of Bayesian Network

in a ubiquitous environment [23]; Wang et al., provide an adaptive user interface model based perceptual control theory [24]; in the world wide web Rebai uses learning algorithms and Bayesian network to provide a new AUIs [25].

There are several contexts in which the adaptive systems have been used. In the health context, for example, the adaptive user interface has been used in a hypermedia data browsing to develop a clinical workstation prototype for the cardiology domain [26], in cancer domains to develop an interactive system which provide patients information [27] in e-health domain for tracking and monitoring health data for patients [7]. In smart environments context, the adaptive user interface has been used to manage and control various devices simultaneously in a modern home theater system [23]. Yen et al. provide a generalized framework for the UI adaptation in complex environments [18]. El-Bakry et al. present a new adaptive interface design for web application [20]. In addition, the adaptive user interfaces have been used in a work smart environments context, for a warehouse picking system [22].

In the mobile device context, the adaptive user interface has been used to adapt a graphical interface to a various type of user [24, 28, 29]. The literature on the adaptive interfaces research context is very heterogeneous and closely linked with each domain of application; however, all these experimental applications propose methods and specific approaches for the creation of new adaptive systems, without considering the adaptation of existing systems.

3 The Design Method

Not every system can just be turned into an adaptive one, so it is worth to specify some requirements that the existing system should meet for the proposed method to be applicable. They are the following:

- Communication and storing capabilities should be supported by the system to ensure the proper data flow;
- Capabilities of sensing contextual data from the user, the environment, the system itself. The system should be able to recognize user and human-system interaction information;
- Adaptable elements and actions should be supported by the system. They allow introducing variability in system behavior based on the changeable requirements.

In addition, the economic feasibility to make a system adaptive should be verified.

The method consists of three main steps: (1) context analysis; (2) adaptive module design; and (3) system re-design (Fig. 1).

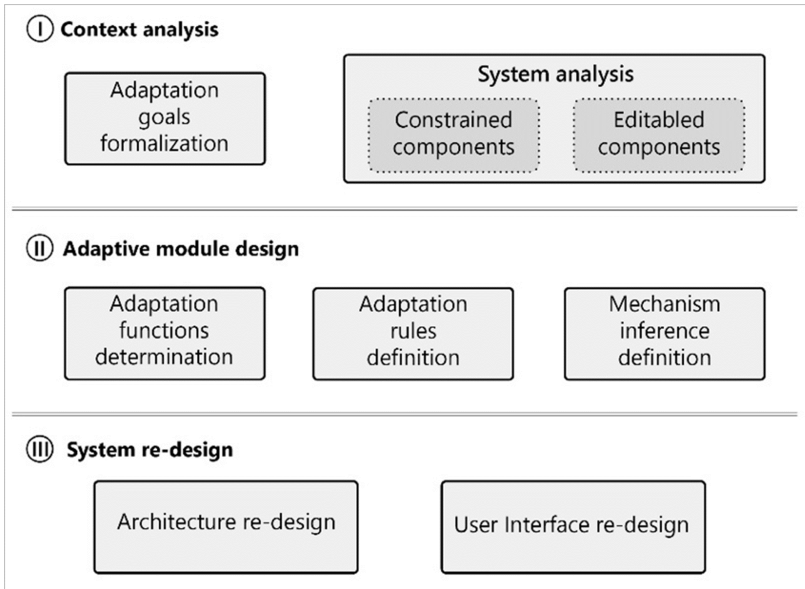


Fig. 1. Design method

The context analysis consists of two interdependent activities. The former identifies the reason to make a system adaptive by defining the “adaptation goals”, which could refer to search goals, problem-solving or learning goals. On the basis of the final target, there are different type of adaptations such as user-centered adaptation, target-oriented adaptation, software architecture adaptation. In general, the profile of the target user plays a key role. Therefore, it is essential to understand his/her characteristics (e.g. goals, needs, moods, preferences, intentions, etc.) and investigate how to acquire the missing information. In case of potential conflicting system goals, it is suggested to perform a trade-off analysis. At the end of this activity, the main functionalities (both existing and new) of the adaptive system must be clear. The latter involves the study of the main system features such as architecture, framework, database, script code and layout, etc. It includes the characteristics of the application, the devices, and the physical features of a system as well as the relationships between elements. According to the adaptation purpose and the company’s needs, the following components have to be identified: existing elements or features that cannot be modified (i.e., constrained component) and existing requirements that can be changed (i.e., editable components). In particular, the main constraints concern technical issues, marketing requirements and economic strategies.

In order to satisfy the adaptation goals, all possible design variabilities should be explored by constructing a design space with the help of experts’ knowledge. The design of the adaptive module consists of three sub-modules. The first one refers to the determination of what aspects of the system are subject to adaptation or what new features, which undergo a change depending on users interaction, should be implemented. They may refer to semantics, syntactic (i.e. sequencing of the interaction) or lexical aspects

of interaction (i.e. interface objects, interaction techniques, media and modalities). They include generic functions (error correction, help, etc.) and interaction (task simplification, improved user experience, etc.). At the same time, it is necessary to establish what aspects of the user consider to make adaptation. The second sub-module identifies how to make adaptation. It consists on the definition of a set of rules that involves the logics formalization for a proper adaptive behavior [30]. The last sub-module concerns the mechanism inference definition that consists in the identification of the domain-independent algorithm used to draw conclusions or perform actions using the knowledge base and the user's responses. It also defines the right time for adaptation, which is a crucial point in the adaptive process. The adaptation can be carried out in run-time (during the use of the system) or statically (at the first access of the system).

Finally, the system re-design consists of modifying the current architecture by inserting the adaptive module and editing the graphical interface. The system architecture redesign includes the database restructuring, the addition of the adaptation module and possible sensors, the data flow optimization, and the enabling of new system actions. The method implications on the user interface redesign could refer to graphics (e.g. colors, brightness, contrast, font, etc.), navigation (e.g. wizard mode), interaction mode (e.g. voice, gesture, etc.), and content (e.g. pre-selected function, help, notify, etc.).

4 The Case Study

The described method is applied, as an example, to a Wi-Fi system for the control and management of household appliances developed by a large Italian company.

4.1 Context Analysis

First of all, the final adaptation goals have been defined: make the system adaptive on the basis of the user behavior and needs. Subsequently, a system analysis to identify the specifics of the smart system has been carried out. Three different modules make the current system architecture: the mobile application, the household appliances and the Cloud (Server-DB) (Fig. 2).

The user-household appliance interaction takes place in two different ways: Direct Interaction (performed on the household appliance) or Mobile Interaction (performed on the mobile application). The application allows the user-household appliance communication both in the home (through Local Network) and outside (through Remote Control). In the first case queries are sent directly to the household appliances and saved on the Cloud DB (View HAs status and HAs Programming). In the case of Remote Control use, however, the queries are sent to the Household Appliances through the Server on Cloud. In detail, the HAs programming queries will be sent through the Appliance Actuator in Active Remote Control mode, while the status will be displayed on Cloud Database. The status update on the Database takes place every 30 s (polling).

On the basis of the defined adaptation goals, the constrained components (CC) and editable components (EC) have been identified. Specifically, the defined CC are: User-Household appliances communication modality, Home-appliance Program Temperature

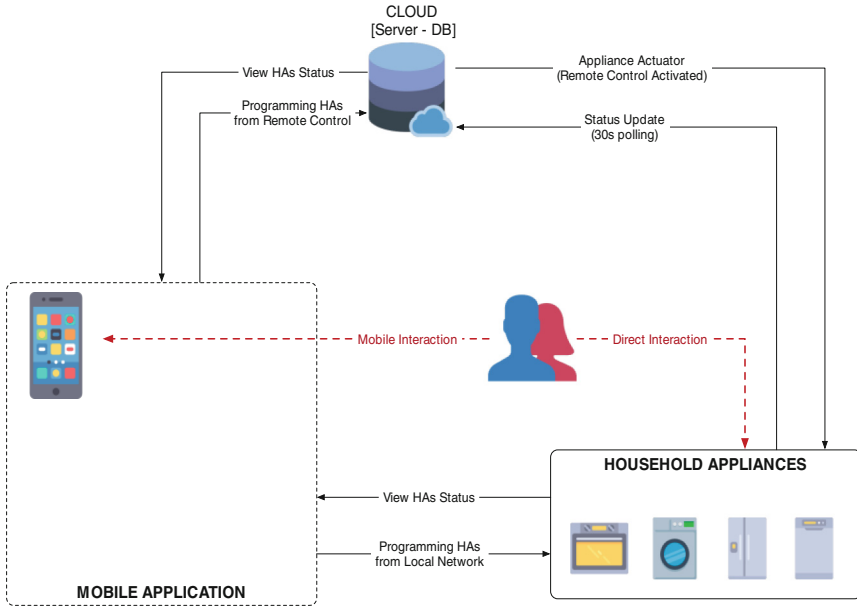


Fig. 2. The architecture design

and Cycle, Mobile Application Interface color and font. Instead, the defined EC are: Database, Mobile Application Interface contrast and layout, mobile application browsing mode, adaptive functions addition (i.e., programs or temperature suggestion).

4.2 Adaptive Module Design

Starting from the current system architecture and based on the defined requirements, the adaptive module has been designed. Specifically, the Adaptive module consists of:

- **Adaptation Rules**, which involves the:
 - **Back-end Translation for Bayesian Network.** This module is responsible for reading the updated Database, filtering and translating it. Specifically, the filtering action selects from the DB the states needed to perform the adaptation. Instead, the translation action consists of translating the states, previously filtered, into a compatible language to be sent to the Adaptive Middleware module.
 - **Adaptive Middleware.** This second module consists of Adaptive Engine, characterized by the Decision-Making Algorithm (DM) [12]. The adaptive middleware is responsible for Bayesian network learning; through this module data received from the Middleware Translation are processed and sent to the network. Once trained the network and obtained the specific conditional probabilities, these will be elaborated through the Decision-Making algorithm. The DM will identify the appropriate adaptation actions to be taken on the system by upgrading network node dependencies on the mobile application.

- Mechanism Inference**, this module consists of the **Bayesian Network** customized for the System. The nodes number constituting the network is affected by the devices number involved and by the user’s monitoring actions. The network learning is made via the adaptive middleware.

4.3 System Re-design

4.3.1 Architecture Re-design

Finally, the system architecture has been re-designed. In details, the Adaptive module introduction has been carried out on two different levels: the first level involves the DB Server on Cloud, the second level involves the mobile application. The adaptation actions suggested by the adaptive module will be shown on the mobile device through the user interface.

Figure 3 shows the adaptive module addition in the system architecture.

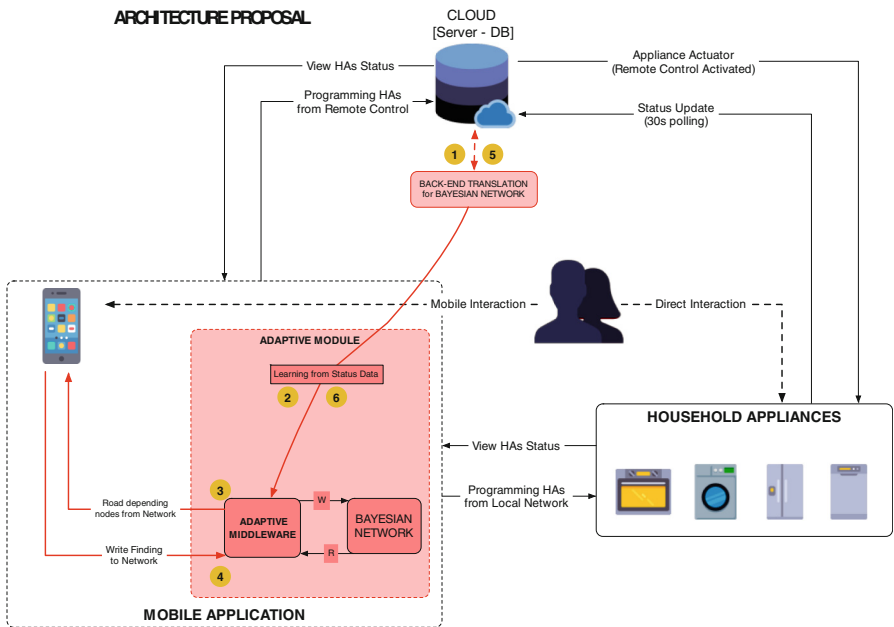


Fig. 3. The architecture re-design (in red the adaptive module addition) (Color figure online)

4.3.2 User Interface Re-design

In order to simplify the control and the setting of household appliances, the mobile application UI has been redesigned. The UI re-designing consists in removing features that are not required by the user and adapting others to his preferences. Based on the user profile, the previously interaction and the monitored actions, the adaptive module introduction allows finding the optimal features combination for the interface (Table 1).

Table 1. User interface re-design.

Current UI	Adaptation goal	UI adaptation action
UI contrast and zoom: fixed	Increase UI accessibility and readability	Adapt UI contrast and zoom at the User Login
–	Improve user experience	Add/replace image with text to help user in programming the device
Single page layout	Optimize the layout (to highlight relevant information and minimize irrelevant and potentially distracting information)	Change UI layout at the User Login (i.e. propose the most fitting option for the user to provide an easy interaction and usage)
No Program suggestion	Best Program Suggestion	Depending on the user login, day and time range, the adaptive network suggests the best Program to the end user. The application will set as the first choice the Program suggested by the network, as a second choice the Program suggested by the Company and in cascade all the others
No Temperature suggestion	Best Temperature Suggestion	Depending on the user login, day, time range and suggested Program, the adaptive network highlights the best Temperature to the end user. The application will set as the first choice the Temperature suggested by the network, as a second choice the Temperature suggested by the Company and in cascade all the others
No Delay Start suggestion	Delay Start and cooking duration Suggestion	Depending on the user login, day, time range and suggested Program, the adaptive network suggest the delay start and cooking duration (oven case). The application will set as the first choice the most used delay start time and cooking duration

The UI adaptations concerned two main features: the first one consists of changing the interface’s characteristics, and the last consists of changing the content displayed by the interface. The first UI adaptations is based on varying the graphical features (contrast, zoom, etc.,) or its layout configuration. In this case, the platform is aware of the user

experience with the interface and it makes the corresponding adaptation based on these interactions. The second UI adaptation deals with the interface contents. Contents represent all interface items editable according to uses and actions the user acts on interface, own preferences and needs.

As a result, the user interface has been split into two presentation modalities: Default (i.e. the current UI) and New UI. The New UI has been designed by creating separate user interfaces (starting from Default UI) and by dynamically adapting the graphical features and the content according to the adaptation rules.

5 Conclusion and Future Works

In order to support the development of smart systems able to ensure a better user experience, this paper proposes a design methodology for making an existing system adaptive. It consists of three steps: from the context analysis in terms of goals, constraints, and opportunities to the redesign of the system architecture and GUI, through the definition of adaptation algorithms.

The methodology has been experimented by an Italian company, which aimed at improving its Wi-Fi system for the control and management of household appliances. It led designers in their activity, supporting the rules definition and reducing the overall efforts. Such approach allowed performing adaptation activities (i.e., monitoring, analysis, decision-making, execution, etc.) without affecting system performance and availability.

Different applications will be carried out to highlight the benefits of using the proposed methodology as opposed to the traditional ones and to validate it. Aspects related to usability, technology, efficiency and costs will be addressed. The final aim is the development of a set of guidelines and a user-friendly tool that effectively support the universal design.

References

1. Lavie, T., Meyer, J.: Benefits and costs of adaptive user interfaces. *Int. J. Hum. Comput. Stud.* **68**(8), 508–524 (2010)
2. Haas, M.W., Hettinger, L.J.: Current research in adaptive interfaces. *Int. J. Aviat. Psychol.* **11**(2), 119–121 (2001)
3. Kushwaha, N., Kim, M., Kim, D.Y., Cho, W.D.: An intelligent agent for ubiquitous computing environments: smart home UT-AGENT. In: *Proceedings of the Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, pp. 157–159. IEEE, May 2004
4. Gullà, F., Ceccacci, S., Menghi, R., Cavalieri, L., Germani, M.: Adaptive interface for smart home: a new design approach. In: Cavallo, F., Marletta, V., Monteriù, A., Siciliano, P. (eds.) *ForItAAL 2016*. LNEE, vol. 426, pp. 107–115. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54283-6_8
5. Letsu-Dake, E., Ntuen, C.A.: A case study of experimental evaluation of adaptive interfaces. *Int. J. Ind. Ergon.* **40**(1), 34–40 (2010)

6. Evans, D., Arasu, V.A., Fendley, M.: A model-based framework for an adaptive user interface in supervisory control. *IFAC Proc. Vol.* **46**(15), 37–43 (2013)
7. Vasilyeva, E., Pechenizkiy, M., Puuronen, S.: Towards the framework of adaptive user interfaces for eHealth. In: 2005 Proceedings 18th IEEE Symposium on Computer-Based Medical Systems, pp. 139–144. IEEE, June 2005
8. Shakshuki, E.M., Reid, M., Sheltami, T.R.: An adaptive user interface in healthcare. *Procedia Comput. Sci.* **56**, 49–58 (2015)
9. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. *Adapt. Hypertext Hypermedia*, 1–43 (1998)
10. Gullà, F., Cavalieri, L., Ceccacci, S., Germani, M., Bevilacqua, R.: Method to design adaptable and adaptive user interfaces. In: Stephanidis, C. (ed.) *HCI 2015. CCIS*, vol. 528, pp. 19–24. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21380-4_4
11. Rothrock, L., Koubek, R., Fuchs, F., Haas, M., Salvendy, G.: Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms. *Theor. Issues Ergon. Sci.* **3**(1), 47–84 (2002)
12. Benyon, D., Murray, D.: Applying user modeling to human-computer interaction design. *Artif. Intell. Rev.* **7**(3), 199–225 (1993)
13. Dieterich, H., Malinowski, U., Kühme, T., Schneider-Hufschmidt, M.: State of the art in adaptive user interfaces. *Hum. Factors Inf. Technol.* **10**, 13 (1993)
14. Stephanidis, C., Karagiannidis, C., Koumpis, A.: Decision making in intelligent user interfaces. In: *Proceedings of the 2nd International Conference on Intelligent User Interfaces*, pp. 195–202. ACM (1997)
15. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. *User Model. User Adap. Inter.* **6**(2–3), 87–129 (1996)
16. Liu, J., Wong, C.K., Hui, K.K.: An adaptive user interface based on personalized learning. *IEEE Intell. Syst.* **18**(2), 52–55 (2003)
17. Namgoong, H., Sohn, J.C., Cho, Y.J., Chung, Y.K.: An adaptive user interface in smart environment exploiting semantic descriptions. In: 2006 IEEE Tenth International Symposium on Consumer Electronics, ISCE 2006, pp. 1–6. IEEE (2006)
18. Yen, G.G., Acay, D.: Adaptive user interfaces in complex supervisory tasks. *ISA Trans.* **48**(2), 196–205 (2009)
19. He, J., Yen, I.L., Peng, T., Dong, J., Bastani, F.: An adaptive user interface generation framework for web services. In: *Congress on Services Part II, SERVICES-2*, pp. 175–182. IEEE (2008)
20. El-Bakry, H.M., Riad, A.M., Abu-Elsoud, M., Mohamed, S., Hassan, A.E., Kandel, M.S., Mastorakis, N.: Adaptive user interface for web applications. In: *Proceedings of the 4th WSEAS International Conference on Business Administration Recent Advances in Business Administration (ICBA 2010)*, pp. 20–22 (2010)
21. Hervás, R., Bravo, J.: Towards the ubiquitous visualization: adaptive user-interfaces based on the semantic web. *Interact. Comput.* **23**(1), 40–45 (2010)
22. Bongartz, S., Jin, Y., Paternò, F., Rett, J., Santoro, C., Spano, L.D.: Adaptive user interfaces for smart environments with the support of model-based languages. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) *AmI 2012. LNCS*, vol. 7683, pp. 33–48. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34898-3_3
23. Song, I.J., Cho, S.B.: Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment. *Expert Syst. Appl.* **40**(5), 1827–1838 (2013)
24. Wang, X., Hua, Q.Y., Zou, F., Guo, L.: An adaptive user interface model for mobile devices based on perceptual control theory. In: 2014 5th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 908–911. IEEE (2014)

25. Rebai, R., Maalej, M.A., Mahfoudhi, A., Abid, M.: Building and evaluating an adaptive user interface using a Bayesian network approach. *Int. J. Comput. Sci. Inf. Secur. (IJCSIS)* **14**(7), 548–565 (2016)
26. Ghédira, C., Maret, P., Fayn, J., Rubel, P.: Adaptive user interface customization through browsing knowledge capitalization. *Int. J. Med. Informatics* **68**(1), 219–228 (2002)
27. Bental, D., Cawsey, A., Pearson, J., Jones, R.: Adapting web-based information to the needs of patients with Cancer. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) *AH 2000. LNCS*, vol. 1892, pp. 27–37. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44595-1_4
28. Mitrović, N., Mena, E.: Adaptive user interface for mobile devices. In: Forbrig, P., Limbourg, Q., Vanderdonckt, J., Urban, B. (eds.) *DSV-IS 2002. LNCS*, vol. 2545, pp. 29–43. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36235-5_3
29. Nivethika, M., Vithiya, I., Anntharshika, S., Deegalla, S.: Personalized and adaptive user interface framework for mobile application. In: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1913–1918. IEEE (2013)
30. Gullà, F., Cavalieri, L., Ceccacci, S., Germani, M.: A BBN-based method to manage adaptive behavior of a smart user interface. *Procedia CIRP* **50**, 535–540 (2016)