INFORMATION
TECHNOLOGY JOURNAL

Soft Systems Methodology in Environment-Aware Case-Based Reasoning System Analysis

Sa'adah Hassan

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia,
43400 Serdang, Selangor, Malaysia

Abstract: The aim of this study is to examine the potential of Soft Systems Methodology (SSM) in the development of environment-aware Case-Based Reasoning (CBR) system. Having awareness of the computing system's environment is useful either for providing information or factor that can affect the system or effected by the events occur in the environment. The CBR system with environment awareness is able to employ real-time contextual information available in the environment. This information or knowledge will contribute to improve the problem descriptions as well as the solutions, in which the system not only depend on information given by user and historical cases. However, resources available in the environment need to be identified prior to the implementation of environment-aware CBR system. This requires an appropriate analysis methodology to build an explicit model of the system's environment to capture relevant environment knowledge for the system. This study presents the SSM analysis in the development of environment-aware CBR system. The discussion is illustrated with reference to the partial analysis of an environment-aware CBR system development. The deliverable of this initial analysis is an explicit model of the system's environment for environment-aware CBR system.

Key words: Case-based reasoning, environment awareness, soft systems methodology

INTRODUCTION

Today's computing systems are situated in mobile distributed computing environment. The current situation has offered a wide range of information. Therefore, it is beneficial for computing systems if aware of the available information in the environment. For example, by utilizing contextual information available in the environment to provide suggestions of relevant services or to generate recommendations to the user (Zimmermann, 2003; Lee and Lee, 2007). In contrast, if the information is based only on local sources, the ability for the computer systems to make appropriate decisions or suggestions may be limited.

Case-Based Reasoning (CBR) is an approach to problem solving in which previous problems and their solutions are stored in a case base and retrieved in response to a query describing a target problem (López de Mántaras et al., 2005). Recently, CBR has been utilized as a method for identifying situation in a dynamic environment. CBR system with environment-awareness is able to provide information about context available in the environment in real-time basis. This information or knowledge will contribute to the problem descriptions and solutions, wherein the traditional CBR system only depends on information given by user and historical cases.

Developing a system that has environment awareness requires a deeper analysis of the system as well as its relationships with its environment. Especially, to model environment-aware computing system that relies on context information as the primary source of knowledge. The analysis approach should be able to yields a collection of collaborating elements in a computing system and its links to the environment (Bustard et al., 2005). The model produced from the analysis should also be able to grasp the human understanding about the system as well as of the diversity of the information available in the environment.

Soft Systems Methodology (SSM) (Checkland, 1981; Checkland and Scholes, 1990) is a well-established approach to understanding the ill-defined problem situation. The potential role of SSM has been discussed in the development of today's computing system (i.e., autonomic computing) (Bustard et al., 2005, 2006; Bustard and Steritt, 2006). In which the SSM's models help to clarify the role of an autonomic system and to identify opportunities for improvement in the wider business context. Such models have also been shown to be beneficial for identifying where computing support is desirable in a human activity system and determining business objectives in computing system design (Bustard et al., 2005). The SSM is also used to build an

explicit model of autonomic computing systems environment in which the model is used to inform the design of autonomic system that able to adjust automatically to the changes occur in its environment (Hassan *et al.*, 2006a). Previous research by Bustard *et al.* (2005), Hassan *et al.* (2006a) have shown on how the relevant environment knowledge needed to support autonomic behavior can be identified from SSM models.

Growing research interest on research on context-awareness in CBR has motivated this study to examine the benefit of SSM to support the development of environment-aware CBR systems. Most of the work on environment-aware of CBR systems has been largely on implementation and less concern on earlier stage of the development. This study takes this opportunity to further investigate the soft systems concept in environment-aware CBR systems development. This study proposes an initial analysis approach to produce an explicit model of the system's environment for environment-aware CBR system.

BACKGROUND AND MOTIVATION

Some backgrounds that motivate this study are given in this section. First sub-section introduces the concept of environment awareness. The following sub-sections provide brief overview of CBR and the soft systems concept.

Environment awareness: Basically, the concept of environment awareness is aligned to context awareness (Want *et al.*, 1992, 1995). Context is any information about the environment that characterizes the situation of an entity, which can be any person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves (Dey *et al.*, 1999, 2000). The goal of environment awareness is to exploit the contextual information in order to provide appropriate knowledge to the user.

Environment awareness is crucial for making decisions and quickly reacting to changing environment. Particularly, for today's computing systems that are ubiquitous and autonomic. The ability to take account of contextual information in the environment is of course very important in interactive and mobile applications. A system with environment awareness stand to benefit from awareness of the context in which they operate (e.g., physical location, availability of other services) to modify the behaviour of applications as well as to guide policy decisions (Want *et al.*, 2003). For example, environment awareness concept is used to locate

members of staff in an office environment (Want *et al.*, 1992) or to discover other devices and services in a network (e.g., network printers, databases).

The increasing importance of environment awareness is also recognized for monitoring the environment of self-managed systems. It is essential to continuous monitoring of changes in the environment to ensure that when abnormal events occur, the system is able to detect the abnormality and adjust its behaviour accordingly to restore normal operation. Furthermore, today's computing systems must share information and resources to ensure continuity of services (e.g., by making use of alternative resources when problems occur). For example, by having knowledge of available resources in the environment are useful for a system to effectively guide user recovery from service failures (Hassan *et al.*, 2006b).

Environment awareness is also needed to build cognitive systems that understand their own goals and the goals of other systems they communicate with and are able to adapt their output to suit their associate systems and the situation (Brachman, 2002). If actions are based only on local information sources, the ability to make accurate decisions may be limited, resulting in perceived conditions that are incorrectly identified (Durham *et al.*, 2006).

Figure 1 shows an example of a problem-solving dialogue in the SHRIEK's printer-troubleshooting domain (Hassan *et al.*, 2006b). The example dialogue is based on an early prototype to illustrate the SHRIEK approach to self-healing and recovery informed by environment knowledge. In this example, some of the information in the problem description is contextual information, such as toner level, power light and printer status, which are obtained from the printer device. The knowledge in the

Default printer: Printer 1
print quality? blank
Problem description: Power light on = yes, able to print = yes,
print quality = blank, toner level = high, printer status = ready
Most similar case:
Case 4: Toner cartridge sealed
able to print = yes
print quality = blank
toner level = high
printer status = ready
Similarity: 1.0
Diagnosis: Toner cartridge sealed
Remedial action:
• Technician notified that toner cartridge in printer-1 is sealed
Recovery strategies:
• Redirect your print job to printer-2 in room 11
• Wait for technician to remove seal from toner cartridge
• Remove seal from toner cartridge and replace cartridge in printer

Fig. 1: A Problem-Solving Dialogue in the Printer Troubleshooting adapted from (Hassan *et al.*, 2006b)

remedial action and possible recovery strategies are also based on information obtained from other systems or applications related to the printing service.

Case-based reasoning: The CBR is an approach to problem solving based on previous experiences or cases (Leake, 1996) a concept inspired by the way that humans solve problems by remembering previous similar problems. In much the same way, a CBR system solves problems by remembering a previous similar problem and reusing its solution. CBR systems solve new problems by utilizing specific knowledge from past experiences (López de Mántaras *et al.*, 2005).

In a CBR system, descriptions of previous problems and their solutions are stored in a case base. In the first step of the CBR cycle, one or more similar cases are retrieved from the case base in response to a query describing the target problem (López de Mántaras *et al.*, 2005). The solution from the most similar case is then reused after being adapted, if necessary, to account for any differences between the target problem and the revised case. Following evaluation of the proposed solution and revision of the solution, if necessary, the problem and its solution may be used to create a new case to be retained for future use. The retention of newly identified problems and their solutions as new cases which can be used to solve future problems enables a CBR system to learn from experience (López de Mántaras *et al.*, 2005). As the system learns to solve new problems, the case base continues to grow.

Research interest in employing CBR as a method for identifying situations in a dynamic environment to the development of context-aware systems is apparently increased. For example, Zimmermann (2003) demonstrated how CBR could be used to generate recommendations for users in a mobile environment. The recommendations were used to augment a museum exhibition with audio information. Kofod-Petersen and Aamodt (2003) incorporated context information as cases in CBR for user situation assessment in a mobile context-aware system. Kumar *et al.* (2005) proposed a context enabled Multi-CBR approach for aiding the recommendation engine in retrieving appropriate information for e-commerce applications. In addition, Leake (1996) and Leake *et al.* (2006) has presented three potential areas in which the use of CBR may provide benefits for context-aware applications, especially in smart homes. Lee and Lee (2007) have developed a context-aware music recommendation system using case-based reasoning, which is called C^2_Music. Their system utilizes user's demographics, behavioral patterns as well as user's context at the time of making recommendation. The system is able to retrieve the similar contexts and similar users.

Moreover, the conversational CBR in SHRIEK (Hassan *et al.*, 2006b; McSherry *et al.*, 2008) improving the solution by providing recovery strategies for the suggested solution in which the remedial actions and fault recovery strategies are informed by environment knowledge. A part from that, the conversational CBR in SHRIEK uses contextual information of the computer devices and services for the problem descriptions.

Soft systems methodology: Soft Systems Methodology (SSM) (Checkland, 1981; Checkland and Scholes, 1990) is a systemic approach to clarifying objectives in complex and dynamic problem situations. The SSM models produced are called holons (Checkland, 1995) (i.e., constructed abstract wholes) that can be used to explore actual and future needs. A conceptual distinction is made between what a system is and what it does and a practical distinction is made between defining the system and modeling its activities as an integrated entity. The four main activities in SSM (Checkland and Scholes, 1990) are:

- Analyzing a problem situation
- Building purposeful activity models by formulating relevant descriptions of activity as root definitions and conceptual models. The elements of the root definition, as defined below, are often referred to by the mnemonic CATWOE:

  - **Customer:** A beneficiary of the system or anyone who is affected by the system
  - **Actor:** Person who performs the transformation
  - **Transformation:** The conversion of input to output activity carried out by the system
  - **Weltanschauung:** The world view which makes the transformation meaningful
  - **Owner:** The person or group with the authority to modify the system
  - **Environment:** Elements outside the system that influence or constrain its activities

- Using the models for debating the situation and seeking from that debate both changes for improvements and action-to-improve to be taken
- Taking action to improve the situation

Understanding the environment is part of the process of determining the requirements of any computing system prior to its creation (Jackson, 1995). In addition to activities such as analyzing, designing, implementing and validating, a systems development approach should also be concerned with the system's interactions to its environment (Sommerville, 2004). In particular for today's computing systems where it should be able to sense and

recognize relevant information in the operating environment to determine what actions should be taken.

The physical world of computing systems must be modeled so that computer systems can have direct and intimate knowledge of their environment, ultimately using that knowledge to effect change (Intel, 2005). The potential benefits of SSM as a basis for developing an environment model for an autonomic computing system has been mentioned in previous work (i.e., Hassan *et al.*, 2006a). Similar to environment-aware CBR, related resources available in the environment need to be identified prior to the implementation of environment-aware CBR system. This requires an appropriate analysis methodology to support the creation of model of the system and its environment. The analysis model should also take account of the dynamic context when dealing with ubiquitous and heterogeneous resources in the computing environment (Mainsah, 2002).

ENVIRONMENT AWARENESS CBR SYSTEM ANALYSIS USING SSM: A CASE STUDY

The development of environment-aware CBR system must begin with an appropriate analysis methodology to identify the knowledge about resources available in the environment that can be used to improve the services of computing system. The analysis is essential to effectively modelled and create understanding of the role of the system environment. Here, it is described how Soft Systems Methodology is used in the analysis stage of environment-aware CBR systems development. The discussion is illustrated with a case study of IT service system.

Analyzing the problem situation: The first step in SSM is to analyse the current problematical situation of the system. A rich picture diagram (SSM) (Checkland, 1981; Checkland and Scholes, 1990) is often useful for depicting a shared understanding of the problem under investigation and identifying significant external dependencies, issues and possible innovations.

This initial analysis provides an understanding of the computing environment and insights into how systems can interact with their changing environment. The first two activities of SSM are involved in this initial analysis. The results of the initial analysis, the rich picture and conceptual model of SSM, are used to provide an understanding of the processes, working scenarios and potentially relationships of the major information likely to affect the system.

An IT service system is used to demonstrate the use of SSM analysis to model the system and its environment.
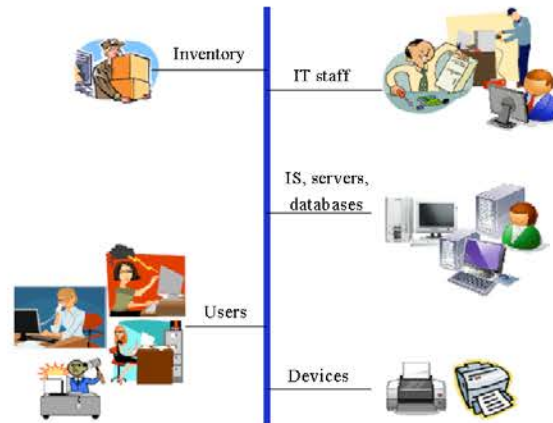

Fig. 2: A rich picture for computing service system

Figure 2 shows a rich picture developed to illustrate the problem situation in the computing service system. A computing service involves a variety of elements that are useful as sources of environment knowledge, for example, network printers, technicians, users, inventory systems and personnel information systems. In common situation, an inventory system (e.g., printer toner cartridges and paper) and IT staff are involved in maintaining the computing services.

The IT service system is meant to provide a continuous computing service to users. However, users may sometimes experience difficulties with the service, for example when an urgent or critical job is stuck in a long queue or the paper is jammed in a network printer. Typically, the solutions available to the user include sending her job to another printer or seeking help from a technician to resolve the problem. However, the situation becomes more difficult if the technician is not available or busy. Such disruptions may sometimes prevent the user from completing an urgent task on time, thus causing her to feel dissatisfied and frustrated with the service. Sometimes, it is quicker for the user to see if she can troubleshoot the problem herself. Or else, it is more efficient if the problem can be detected earlier and preventive action taken before a service failure occurs.

Building purposeful activity models: The rich picture shown in Fig. 2 helps to identify possible viewpoints. One of the possible viewpoints revealed is a system to provide a continuous computing service for users. A root definition based on CATWOE elements can be identified from this viewpoint. The CATWOE elements for the viewpoint are:

C : Users
A : IT staff (e.g., technician)
T : response and handle computing service failures
W : Provide a continuous computing service
O : IT department
E : Available resources and information in the computing system environment

The root definition is formulated from the CATWOE analysis and combined in a simple statement, namely:

An IT department owned system to provide a continuous computing service for users by using IT staff to response and handle computing service failures while taking account of available resources and information in the computing system environment.

The transformation (T) of the system is to 'response and handle computing service failures, which implies the need for a system that is capable of avoiding faults that may result in service failures and able to transform an erroneous system state into one in which the system can continue its normal operation. Accompanying this transformation is a weltanschauung (W) or world view, provide a continuous computing service, which defines the purpose of the transformation. The environment (E) consists of elements that can affect the service. The diagram in Fig. 3 is a conceptual model of the system created from the root definition. The conceptual model demonstrates an ideal view of the activities and their logical dependencies as described by the root definition.

The activities identified in the conceptual model become a source of ideas for change and enquiry for improvement in the problem situation. The activities of 2,

3 and 4 are normally performed by the devices themselves as smart devices. Other activities are performed by technicians and users of the IT service. Activities in the area enclosed by the dotted line are those involved in the transformation (T). The purpose of Activity 10 is to monitor the service to ensure that the defined weltanshauung (W) is achieved, Taking Control Action (TCA) if necessary.

Analysis for environment knowledge in conceptual model: In the light of SSM, all relevant parties and their roles can be clearly identified. The connectivity of human resources and computing resources (e.g., software tools and hardware devices) indicates the information flow and operation sequence among different system activities or components.

The conceptual model provides a useful framework for analyzing the system and its activities in order to identify opportunities for improvement. All the activities in the conceptual model were analyzed to discover contextual information and environment knowledge required to accomplish its activity. Each activity is queried and its interaction with other activities is examined to identify environment knowledge required to accomplish its task.

The focus of this study is in the development of environment-aware CBR system. Therefore, based on the activities in the conceptual model, the Activity 6 (troubleshoot the problem) is the most relevant activity for CBR system. In addition, the dependencies between activities in the conceptual model can be used to trace ripple effects of change. For example, if problem occur, then it can be seen by examining the activities that follow is activity 5 (inform IT staff) in the conceptual model.

The activities in the conceptual model were analyzed to discover the contextual information and environment knowledge required to accomplish its task (Table 1). For example, to troubleshoot a problem (activity 6), the system needs to know the device details and status. It also needs to know which person in charge is responsible for the
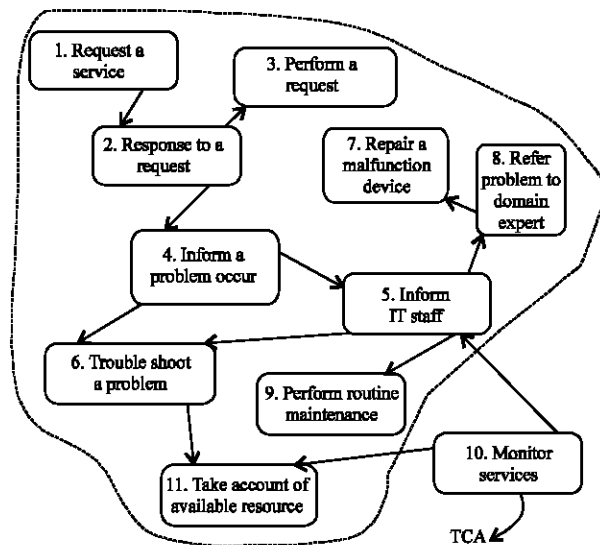

Fig. 3: A conceptual model of computing service system

Table 1: Related knowledge in conceptual model

| Activity	Contextual information	Related knowledge
Inform a problem occur	• User details • Device details • Device status	
Inform IT staff	• Staff responsibilities • Device details • Device status	• Available staff who can be contacted
Troubleshoot a problem	• Device details • Device status • Staff responsibilities and availability	• Available staff who can be contacted • Available devices
Take account of available resources	• Device details • Device status • Staff responsibilities	• Available devices in local network • Relevant databases in local network (e.g., inventory, staff)
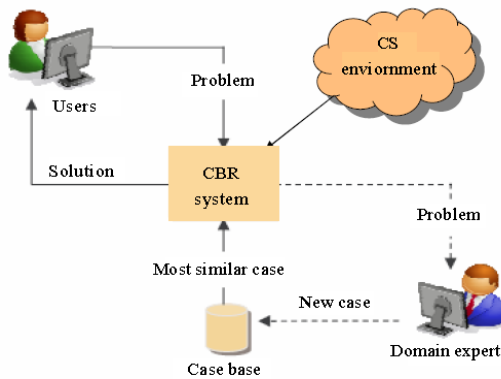
Fig. 4: Environment-aware CBR system architecture

device and whether he or she is available or not at that moment. This information is also important for the user if the diagnosed problem (e.g., out of toner) requires the attention of the technician to repair the fault (e.g., install a new cartridge). Knowledge about what alternatives are available is also required, for example in SHRIEK (Hassan *et al*., 2006b; McSherry *et al*., 2008) for identifying recovery strategies.

Conceptual architecture of environment-aware CBR system: The architecture of environment aware CBR system is shown in Fig. 4 in order to get an overview of how the environment-aware CBR system will operate.

As shown in Fig. 4, the computer system's environment will provide contextual information to CBR system. The CBR system diagnoses failure based on problem description given by the user as well as obtained from the contextual information. The solution from the most similar case of the target problem is then retrieved from the case base. The solution is then improved by the system with related knowledge generated from the contextual information obtained from the environment. If possible, the system can also suggest alternative solutions to the user.

The CBR system utilized the information from the computing system environment to help users to be able to use computing services in the occurrence of failures when possible (e.g., information of the contact person who is responsible for maintaining the printing service, information of nearby printers).

DISCUSSION

Identifying users' requirements of any computing system is crucial in the system development. Particularly for an environment-aware system in which it is not only expected to meeting the needs of users but also to be closely aligned to its environment that it dealing with.

This study has shown the practicability of SSM for environment-aware CBR systems development in which, environment awareness is properly reviewed at the analysis level. The SSM analysis model produced is concerned about the system's interactions to its environment.

Applying the SSM on environment-aware CBR development can be useful way of thinking about the overall situation of the development of the CBR system. It provides a model for visualizing not only internal factors of the system, but also external factors as an integrated entity. Through the SSM analysis, the environment awareness that need to be considered in the activities can be encountered from the analysis model.

The preceding section has also presented the conceptual architecture of environment aware CBR system. It elaborates how the environment awareness can be applied to improve in diagnosing problem and generating solution based on the information of available resources in environment.

Nevertheless, a few limited efforts to cover in details in designing and implementing the environment-aware CBR system. Issue such as to ensure the validity of knowledge due to dynamic changes in the environment and to ensure that the knowledge base is continually updated are some of the issues needed to be considered in future work. Moreover, the representation of environment knowledge should be in the form that enables the knowledge base to accommodate changes in the environment. This issues need to be investigated further for evaluating the effectiveness of the model.

CONCLUSION

Research work on applying environment awareness in case-based reasoning systems are emerging. However, very little attention has been given at the development process, particularly in analysis stage. This study has taken this opportunity to introduce soft systems concept in the analysis of environment-aware CBR system development. The analysis approach introduced in this study can be regarded as a strategic analysis approach to the development of the environment-aware CBR system.

REFERENCES

Brachman, R.J., 2002. Systems that know what they are doing. IEEE Intell. Syst., 17: 67-71.

Bustard, D.W., R. Sterritt, A. Taleb-Bendiab, A. Laws, M. Randles and F. Keenan, 2005. Towards a systemic approach to autonomic systems engineering. Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, April 4-7, IEEE Computer Society, Greenbelt, Maryland, pp: 465-472.

Bustard, D.W., R. Sterritt, A. Taleb-Bendiab and A. Laws, 2006. Autonomic system design based on the integrated use of SSM and VSM: A review. Artificial Intell. Rev., 25: 313-327.

Bustard, D.W. and R. Steritt, 2006. A Requirements Engineering Perspective on Autonomic Systems Development. In: Handbook of Autonomic Computing, Parashar, M. (Ed.). CRC Press, Boca Raton.

Checkland, P.B., 1981. Systems Thinking, Systems Practice. John Wiley and Sons, Chichester.

Checkland, P. and J. Scholes, 1990. Soft Systems Methodology in Action. John Wiley and Sons, Chichester.

Checkland, P.B., 1995. Soft Systems Methodology and its Relevance to the Development of Information Systems. In: Information Systems Provision: The Contribution of Soft Systems Methodology, Stowell, F.A., (Ed.). McGraw-Hill, England, pp: 1-15.

Dey, A.K., D. Salber, G.D. Abowd and M. Futakawa, 1999. The conference assistant: Combining context-awareness with wearable computing. Proceedings of the 3rd International Symposium on Wearable Computers, Oct. 18-19, San Francisco, CA, pp: 21-28.

Dey, A.K. and G.D. Abowd, 2000. CybreMinder: A context-aware system for supporting reminders. Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing, Sept. 25-27, Springer-Verlag, London, pp: 172-186.

Durham, L.M., J. van de Groenendaal, J. He, J. Hobbs, M. Milenkovic and M. Yousif, 2006. Platform support of autonomic computing: An evolution of manageability architecture. Int. Technol. J., 10: 253-263.

Hassan, S., D. McSherry and D.W. Bustard, 2006a. Autonomic self-healing and recovery informed by environment knowledge. Artificial Intell. Rev., 26: 89-101.

Hassan, S., D.W. Bustard and D. McSherry, 2006b. Soft systems methodology in autonomic computing analysis. Proceedings of the UKSS International Conference, Sept. 11-13, Oxford, pp: 106-115.

Intel, 2005. Built-in manageability and proactive security for business desktop PCS. Intel vPro technology white paper.

Jackson, M., 1995. Software Requirements and Specification: A Lexicon of Software Practice, Principles and Prejudices. Addison-Wesley, London, ISBN-10: 0201877120.

Kofod-Petersen, A. and A. Aamodt, 2003. Case-based situation assessment in a mobile context-aware systems. Workshop on artificial intelligence for mobile systems, Seattle.

Kumar, P., S. Gopalan and V. Sridhar, 2005. Context enabled multi-CBR based recommendation engine for E-commerce. Proceedings of IEEE International Conference on E-Business Engineering, (ICE-BE' 05), IEEE Computer Society Press, Los Alamitos, pp: 237-244.

Leake, D.B., 1996. Case-based Reasoning: Experiences, Lessons and Future Directions. AAAI Press, Menlo Park, CA.

Leake, D., A. Maguitman and T. Reichherzer, 2006. Cases, Context and Comfort: Opportunities for Case-Based Reasoning in Smart Homes. In: Designing Smart Homes, Augusto, J.C. and C.D. Nugent (Eds.). Springer, Heidelberg, pp: 109-131.

Lee, J.S. and J.C. Lee, 2007. Context Awareness by Case-Based Reasoning in a Music Recommendation System. Springer, Heidelberg, pp: 45-58.

López de Mántaras, R., D. McSherry, D. Bridge, D. Leake and B. Smyth et al., 2005. Retrieval, reuse, revision and retention in case-based reasoning. Knowledge Eng. Rev., 20: 215-240.

Mainsah, E., 2002. Autonomic computing: The next era of computing. Electronics Commun. Eng. J., 14: 2-3.

McSherry, D., S. Hassan and D. Bustard, 2008. Conversational Case-Based Reasoning in Self-Healing and Recovery. Springer-Verlag, Heidelberg, pp: 340-354.

Sommerville, I., 2004. Software Engineering. 7th Edn., Addison-Wesley, London.

Want, R., A. Hopper, V. Falcão and J. Gibbons, 1992. The active badge location system. ACM Trans. Inform. Syst., 10: 91-102.

Want, R., B.N. Schilit, N.I. Adams, R. Gold and K. Petersen et al., 1995. An overview of the PARCTAB ubiquitous computing experiment. IEEE Personal Commun., 2: 28-43.

Want, R., T. Pering and D. Tennenhouse, 2003. Comparing autonomic and proactive computing. IBM syst. J., 42: 129-135.

Zimmermann, A., 2003. Context-Awareness in User Modelling: Requirements Analysis for Case-Based Reasoning Application. In: Case-Based Reasoning Research and Development, Ashley, K.D. and D.G. Bridge (Eds.). Springer-Verlag, Heidelberg, pp: 718-732.