Would the Best Candidate Please Step Forward? Reflections on the Pros and Cons of Different Support Mechanisms for Online-Questionnaire Design

Aaron Cooper, University of New Brunswick, P.O. Box 4400, Fredericton, N.B., Canada, E3B 5A3; E-mail: AaronR.Cooper@unb.ca
Jo Lumsden, NRC IIT e-Business, 46 Dineen Drive, Fredericton, N.B., Canada, E3B 9W4; E-mail: jo.lumsden@nrc-cnrc.gc.ca

1. INTRODUCTION

The popularity of the Internet makes it an attractive option for conducting surveys – it has even been predicted that online-questionnaires will replace traditional delivery methods [3]. Although online-questionnaires offer many [e.g., 7, 15, 24] advantages over traditional paper-based questionnaires, the nature of the electronic medium itself brings some unique challenges [e.g., 3, 6, 11, 15]. To support the creation of online-questionnaires that fully exploit the advantages whilst addressing the challenges, guidelines have been derived to support online-questionnaire design [22].

An environmental scan of existing online-questionnaire design tools found that little, if any, support is currently incorporated within tools to *guide* online-questionnaire design according to the essence of these guidelines [24]. A logical step towards better supporting online-questionnaire design is, therefore, to practically and effectively incorporate the online-questionnaire design guidelines into design tools. This paper reflects on the pros and cons of various candidate support mechanisms which we considered during the realization of this step. The discussion is, necessarily, biased towards our goal and constraints; it is not our intention to extrapolate generic implications from our reflections, but rather to discuss our systematic comparison process and resulting observations in the hope that both may be useful to researchers and developers faced with a similar challenge.

Section 2 provides some limited background information about online-questionnaires and the guidelines derived to support their design; readers are referred to [22-24] for more detail. Section 3 reflects on a comparison of candidate support mechanisms which we undertook to identify appropriate mechanisms for incorporating the guidelines into a commercially available online-questionnaire design tool. The paper concludes with an outline of our planned future work.

2. BACKGROUND

Generating a respondent-friendly questionnaire can pose many challengers for designers, especially inexperienced designers [23, 30]. Design is a complex activity, often involving too much information to feasibly consider without support [10]; *with* support, however, there is potential for designers to be more productive and to produce better quality artifacts [30]. One means by which to provide such support for online-questionnaire design is in the form of design guidelines.

Until recently, online-questionnaire design guidelines have, however, been somewhat scarce [6, 25, 34, 39]. In an attempt to overcome the lack of a practical reference guide, Lumsden [22] derived an extensive set of online-questionnaire design guidelines from principles in the relevant domains of website, paper-questionnaire, and online-questionnaire design. Informed by research on website design, the guidelines have a strong focus on accessibility and usability, such as issues concerning visually impaired users and the elderly [24].

Despite their relevance there are, however, several problems generally associated with the practical use of guidelines [16, 18, 26]: when manifested as a physical document, guidelines are often impractical and therefore ignored [16]; when they must be manually located from a central source they become de-contextualized, lessening their apparent applicability and/or reducing awareness of guideline violation [18]; designers may not know how to interpret and apply abstract guidelines correctly [16, 26]; and guidelines can conflict with one another – increasing the cognitive demands placed on designers [16, 26] – and some (e.g., those relating to design consistency) can be hard to manually enforce. Limiting the practical effectiveness of guidelines, these issues can be addressed by incorporating guidelines into design software and thereby making guidelines an integral and active part of the design process.

Rarely do available online-questionnaire design tools incorporate support for the essence of Lumsden's guidelines [24]. In the rare instances where guidelines are incorporated, they are typically conveyed via defaults and non-context sensitive help facilities. Where defaults are used, designers are not advised about the possible side effects of modifying the default choices. Non-context sensitive help facilities, such as documentation commonly found under a help menu, provide inadequate support because designers must discontinue their design activity and manually search for design guidance – the interruption, combined with inconvenience, making this an unlikely occurrence.

In a trial evaluation of Lumsden's guidelines, based on their current hardcopy manifestation, they were shown to add considerable value to the resulting design of an online-questionnaire [23]. For reasons already noted, it is unlikely that the guidelines will achieve widespread practical adoption and impact in their current paper form. Our aim is, therefore, to investigate how best to incorporate the guidelines within an online-questionnaire design tool such that they inform the design from a practical, active (as opposed to their current passive) perspective.

Online-questionnaire design tools are complex applications and, since current tools (as already mentioned) provide minimal active design *guidance*, we investigated other genres of software design tools in order to systematically review potential candidate support mechanisms for use within the context of online-questionnaire design tools. The remainder of this paper discusses the results of this evaluative process.

3. ASSESSING CANDIDATE SUPPORT MECHANISMS

We define a *support mechanism* to be a user interface element (or set of related elements) that assists a designer in creating better products and/or in solving problems.

To allow us to *systematically* assess existing support mechanisms relative to our problem domain, we had to first establish a set of criteria to guide our comparison. We based our set of criteria on the fact that design support within an online-questionnaire design tool should [24]:

- point out sub-optimal decisions;
- be context-linked to actions;
- set up best choices by default;
- educate a designer about guidelines; and
- automate certain tasks.

We added the criterion that the support mechanism within the tool should be able to provide suggestions to the designer, as well as a number of other criteria as

Table 1. A consequence table

	Primary Supports			
Criteria	critic	wizard	knowledge repository	automated design creator
Increases chances of creating respondent-friendly questionnaires	1st	2nd	4th	3rd
can point out sub-optimal decisions	Y			
can be context-linked to actions	Y	Y		
can set up best choices by default		Y		Y
can educate user about guidelines	Y		Y	
can automate tasks		Y		Y
can provide suggestions	Y			
Approach meets desirable implementation criteria	1st	3rd	4th	1st
been implemented in a web environment	Y	Y	Y	Y
could encode all guidelines (All, Most, Some, None)	Most	Some	None	Most
could be implemented within my time frame	Y	Y	Y	Y
Supported by others' research	1st	3rd	4th	2nd
positive author comments	1st	3rd	4th	2nd
negative author comments	1st	3rd	4th	1st
Overall:	1st	3rd	4th	2nd

shown in Table 1. We used *consequence tables* [17] (see Table 1) to support our goal-specific comparison process.

3.1 Support Mechanisms

Via a systematic literature review, we identified several potentially appropriate support mechanisms – originating in critiquing systems [31, 36, 38], learning systems [21, 29, 40], help systems [1, 2, 5], and automated systems [4, 20, 35] – and categorized the mechanisms as follows:

- Critics;
- Wizards;
- Knowledge Repositories;
- Automated Design Creators; and
- Scaffolds and Supplementary Supports.

Critics

A *critic* is an intelligent agent that provides assistance during the design process [31] by analyzing a designer's solution [36]. A critic can analyze a design according to various criteria, such as guideline adherence, completeness, design consistency, and alternative design choices [37]. Feedback from the analysis is then presented to the designer (often as a list of messages); once the feedback is considered, appropriate design modifications can be made. Critics are found in critiquing systems – these systems are responsible for generating and presenting critic feedback, activating and de-activating critics, and updating internal design representations.

Wizards

A *wizard* guides a user through a process on a step-by-step basis [1]. A user is typically presented with options for selecting/entering desired preferences and properties; the wizard then automatically performs corresponding tasks based on the user's specifications.

Knowledge Repositories

A *knowledge repository* stores and displays information that can assist a designer in making educated design decisions. Some systems have included design guidelines as hypertext references so that a designer is not required to laboriously reference a paper document [18, 19]. Others have included statistics about past design decisions [16] so that such knowledge can provide additional context.

Automated Design Creators

Some researchers are concentrating on *automated generation techniques*, such as methods found in the fields of expert systems [20, 35], generative programming [4, 33], and model-driven development [9, 14]. A common aim among these systems is to provide a designer with the ability to specify design requirements and/or objectives after which the system produces a corresponding design. Some systems allow a designer to manually modify the solution once generated [9].

Scaffolds and Supplementary Supports

Scaffolds are defined as "*software features that address the cognitive obstacles learners face so they can engage in the work in an educationally productive manner*" [29, p.81]. With the support of scaffolds, students are often able to successfully perform work which likely would not otherwise have been possible [29].

A natural bi-product of providing practical guidance to designers during the creation of an online-questionnaire – and an important anticipated outcome of exposure to appropriate support mechanisms during design activities – is furthering the learning of online-questionnaire designers. As such, although our educational objective is not as explicit as it is in e-Learning systems, we chose to include scaffolds in our evaluation because we believe scaffolds may be a useful approach for conveying the information contained in guidelines.

We classify a range of other mechanisms as *supplementary supports* – e.g., templates, hypertext, defaults, and examples – which can be used by any of the primary mechanisms previously described. For example, a wizard can use hypertext to link to other actions in an interface.

3.2 Candidate Comparison

We believe that the nature of scaffolds and supplementary mechanisms makes them better suited, and more powerful, as complementary supports in our context; for instance, a critic that alerts a designer to an ill-advised modification of a default property offers more support than a system that only includes defaults with no advising capabilities. With this in mind, we systematically compared the relative merits of the other mechanisms to act as a primary support. Table 1 shows our primary comparison criteria, divided into sub-criteria. The "positive/negative author comments" sub-criteria are further divided as shown in Table 2; we evaluated each on the three point scale shown. The following discussion is from the perspective of comparing a critiquing system to the remaining three candidates relative to our specific goal; this standpoint was adopted both for ease of discussion and because, upon initial review, the critiquing approach seemed potentially the most capable of meeting our specific needs. In essence, the discussion serves to confirm this observation.

3.2.1 Favorable Characteristics

Since many critiquing systems demonstrate the potential to include a large number of guidelines, it seems that a critic has the capability of supporting the majority of Lumsden's extensive guidelines. This is important because it maximizes the comprehensiveness of active support that can be offered to a designer. Consider this capability in comparison with the other noted mechanisms. Since a wizard is primarily suited towards procedural tasks, it is likely that a wizard could only provide active support for very specific guidelines – for instance, those guidelines

Table 2. Assessed positive and negative criteria – Yes = Y; Sometimes/Somewhat = S; No = N

		Primary Supports			
	Criteria	critic	wizard	knowledge repository	automated design creator
Positive	promote reflection on decisions	Y	N	N	S
	reduce design error costs	Y	Y	S	Y
	support during design context	Y	S	N	S
	novice considers expert issues	Y	N	S	N
	designed for real work environments	Y	Y	Y	S
	part of natural design process	Y	N	S	N
	ideal for structured tasks	S	Y	N	Y
	no need to search for guidance	Y	Y	N	Y
	can link to other UI services	Y	Y	Y	Y
	ideal for novice users	N	Y	S	S
	user acquires knowledge	Y	S	S	S
	guidance is searchable and/or explorable	S	N	Y	N
	good for consistency & low-level issues	Y	Y	N	Y
	decisions can be made for user	N	S	N	Y
	user can guide system to desired solution	N	N	N	S
	speed up development	S	S	S	S
	improve quality of work	S	S	S	S
Negative	potential for designer resistance/reluctance	N	S	Y	N
	feedback may not be viewed	S	Y	Y	Y
	designer may be left to perform many manual tasks	N	S	N	Y
	feedback may be irrelevant	S	Y	S	Y
	creativity may be stifled	S	S	Y	N
	concentration may be interrupted	S	Y	N	Y
	designer may not be encouraged to reflect on decisions	S	S	Y	N
	mechanism is often explicitly invoked	Y	N	N	S
	cannot run concurrently while designer modifies design	Y	N	Y	Y
	assistance opportunities typically designer responsibility	Y	N	N	Y
	mechanism not well integrated into design environment	Y	Y	N	Y
	feedback may be overwhelming	S	Y	N	Y
	designer may be forced to read documentation	S	Y	N	Y
	designer may lose feeling of being in control	Y	S	Y	S
	designer's knowledge may not contribute to design	Y	S	Y	S
	resulting interfaces tend to look similar to one another	Y	N	Y	N

that pertain to the suggested structure (i.e., page breakdown) of an online-questionnaire. A knowledge repository is a passive form of support that is likely to manifest as a de-contextualized hyperlinked manual of guidelines, which would have to be manually searched by the designer. Finally, although an automated design creator could also encode a large portion of the guidelines, Eisenstein and Puerta have argued that "*the only way to build a system that is usable in real-world design projects is to focus on those areas of design that are particularly amenable to automation and leave other areas of design in human hands*" [8, p.75]. On this basis, it is likely that automating all of Lumsden's guidelines is not appropriate, albeit a semi-automated approach might be suitable. This is, in fact, the approach of critiquing systems: they provide an automated analysis tool that focuses on tedious and low-level details (a task ideally suited to computers), while leaving the designer in charge of making final design decisions.

A second attractive feature of a critic is its ability to encourage a novice designer to think like an expert [30]; critics actively bring relevant issues and possible trade-offs to the designer's attention. In comparison, wizards seldom allow such opportunities to present themselves since they are automated tools that aim to *hide* details from a user. Despite the expert knowledge contained within a knowledge repository, the process of obtaining this information is far too passive and laborious to make it a viable means by which to encourage expert thinking. Although an automated design creator can also encourage expert thinking (e.g., [8, 28]), the process is not as explicit as in a critiquing system. Further criteria favoring a critic include its ability to explicitly educate a designer about guidelines, as well as actively pointing out sub-optimal design decisions and providing suggestions on possible fixes [37]. Using a wizard can lead to some transfer of knowledge, but not to the extent that comes from interacting with a critic, and very rarely are explicit educational opportunities provided in an automated design creator. A knowledge repository can promote learning, but in a reactive fashion. None of the other three approaches offer analysis and advice provision capabilities.

Finally, the concept of critiquing is a *natural* part of design [31]; to improve an artifact, a designer must determine whether aspects of the design need improving and if so, the means by which to make the corresponding improvements.

3.2.2 Unfavorable Characteristics

While there are many favorable characteristics to support the critic approach, we must equally consider the disadvantages or unfavorable characteristics of this approach. Principal amongst these is the potential for user reluctance; one study showed that critics were seen to challenge designers' authority and the critiquing was perceived as antagonistic [37]. The study viewed negatively the fact that designers anticipated the criticism and performed design actions in an effort to avoid the firing of feedback (although some might argue it to be a positive outcome). Whatever one's stance, a subsequent study [30] suggests that, unsurprisingly, it is the design of the critic feedback and corresponding interaction, as opposed to the critic metaphor itself, that determines designer response to the support mechanism.

While the possibility exists for a designer to resist a critic, the same risk is equally real for the other three support mechanisms. Novice users prefer lots of guidance and limited choices, while expert users tend to favor less guidance and the availability of many options [27]; as such, it is highly likely that a wizard could be viewed as restrictive by experienced designers. Users rarely consult passive documentation and online-help [2] and so although, being easily ignored, knowledge repositories are less likely to meet explicit resistance, as a solution in our context they are less likely than a critic to be effective *because* they can so easily be disregarded. Finally, automated design creators are often criticized on account of the limitations they place on the extent to which a designer can actively bring his/her knowledge to the design process [38] and users of expert systems typically resent being placed in a passive human-computer interaction role [35]. In contrast, researchers have argued in favor of critics because such mechanisms encourage collaborative problem solving [12, 31, 35, 38]. Instead of a one-sided problem solving approach, both the designer's and system's knowledge play a key role in the design process [38].

Robbins [31] asserts that existing critiquing systems force the designer to engage in too many manual tasks, namely: modifying a design in response to critic feedback; enabling/disabling relevant/irrelevant critics; and instructing the system in how a critique was resolved. Although theses issues are important, they are essentially facets of the interaction design (and hence, usability) of a critic as opposed to concerns about its inherent ability as a concept to effectively support a designer. Furthermore, leaving the designer in control is a *strength* of critiquing systems; the ability to, for example, manually modify a design is a desirable characteristic in such software. A wizard can be used to start a design session from a default template, but complete construction of a questionnaire is unlikely to be possible without some manual designer intervention. Indeed, only certain aspects of questionnaire creation are amenable to automation and manual design decisions and modifications are unavoidable. Of all four mechanisms, knowledge repositories place the heaviest manual burden on the designer; guidelines must be manually searched, read, and interpreted. At the other end of the spectrum, as already mentioned, an automated design creator typically does not provide *enough* opportunities for manual intervention. Although some automated design creators (e.g., [33]) allow a designer to actively engage with the system and manually select components for the later automated generation phase, we believe that having to choose a multitude of low-level design components places no fewer demands on the designer than a critiquing system.

Finally, critiquing systems are sometimes criticized on account of the degree to which they can potentially interrupt a designer's task concentration [13]. While certainly a valid concern, interruption is not always a negative issue; a critic educates a designer by posing appropriate questions in real time and reflection is encouraged. Reflection is a part of the mental process during design and researchers [13, 32, 37] often cite psychological theories to support the use of the critic mechanism on these grounds.

4. SUMMARY AND FUTURE WORK

As the first step towards incorporating guideline support within an online-questionnaire design tool, we have conducted an environmental scan of appropriate design support mechanisms used in other genres of software design tools. We have concluded that the advantages of a critic outweigh its potential disadvantages within our specific context. We believe that this natural user interface metaphor, which encourages self-critiquing of a work-in-progress, is the most appropriate mechanism for supporting an online-questionnaire designer.

We observed that critiquing systems normally focus on visual feedback – typically relying on the presentation of textual messages to convey critiques. Alternative modalities (such as audio feedback) could prove to be beneficial in better supporting designer-critic dialogue; indeed, Ericsson suggests that "*better cues are needed*" to signal comments needing immediate attention [10]. We therefore propose to investigate alternative means of presenting critic feedback to a designer. Once we have completed our development phase, we intend to conduct extensive usability studies to determine the precise aspects of our design that do and do not work in the context of online-questionnaire design; on the basis of the findings of such studies, we hope to improve on our initial design and identify which components of support benefit from automation and which should be left in the control of the designer.

As noted in the introduction, we acknowledge that the discussion presented here is biased towards our goal and constraints. Although we have not, therefore, extrapolated generic implications from our reflections, we hope that this discussion concerning our comparative observations will prove useful to researchers and developers faced with a similar challenge. Furthermore, we hope our observations might also prove useful in terms of knowing what to look for when selecting a well-supported online-questionnaire design tool in the future!

REFERENCES

1. Ames, A.L., Just What They Need, Just When They Need It: An Introduction to Embedded Assistance. in *19th Annual International Conference on Computer Documentation*, (Santa Fe, USA, 2001), 111-115.
2. Baecker, R., Showing Instead of Telling. in *20th Annual International Conference on Computer Documentation*, (Toronto, Canada, 2002), 10-16.
3. Couper, M.P. Web Surveys: A Review of Issues and Approaches. *Public Opinion Quarterly*, **64**(4). (2000). 464-494.
4. Czarnecki, K., Overview of Generative Software Development. in *International Workshop on Unconventional Programming Paradigms*, (Le Mont Saint Michel, France, 2005), 326-341.
5. Delisle, S. and Moulin, B. User Interfaces and Help Systems: From Helplessness to Intelligent Assistance. *Artificial Intelligence Review*, **18**(2). (2002). 117-157.
6. Dillman, D.A. *Mail and Internet Surveys: The Tailored Design Method*. John Wiley & Sons Inc., New York, 2000.
7. Dillman, D.A., Tortora, R.D. and Bowker, D. *Principles for Constructing Web Surveys*, Washington State University, Pullman, Washington, 1998.
8. Eisenstein, J. and Puerta, A., Adaptation in Automated User-Interface Design. in *5th International Conference on Intelligent User Interfaces*, (New Orleans, USA, 2000), 74 - 81.
9. Eisenstein, J. and Puerta, A., Towards a General Computational Framework for Model-Based Interface Development Systems. in *4th International Conference on Intelligent User Interfaces*, (Los Angeles, USA, 1999), 171-178.
10. Ericsson, M. *Commenting Systems as Design Support*, Department of Computer and Information Science, Linköping University, Linköping, Sweden, 1996, 193.
11. Evans, J.R. and Mathur, A. The Value of Online Surveys. *Internet Research: Electronic Networking Applications and Policy*, **15**(2). (2005). 195-219.
12. Fischer, G., Lemke, A.C., Mastaglio, T. and Morch, A.I. The Role of Critiquing in Cooperative Problem Solving. *ACM Transactions on Information Systems*, **9**(2). (1991). 123-151.
13. Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G. and Sumner, T., Embedding Computer-Based Critics in the Contexts of Design. in *SIGCHI Conference on Human Factors in Computing Systems*, (Amsterdam, The Netherlands, 1993), 157-164.
14. Fraternali, P. and Paolini, P., A Conceptual Model and a Tool Environment for Developing More Scalable, Dynamic, And Customizable Web Applications. in *6th International Conference on Extending Database Technology*, (Valencia, Spain, 1998), 421-436.
15. Fricker, R.D., Jr. and Schonlau, M. Advantages and Disadvantages of Internet Research Surveys: Evidence from the Literature. *Field Methods*, **14**(4). (2002). 347-367.
16. Grammenos, D., Akoumianakis, D. and Stephanidis, C. Integrated Support for Working with Guidelines: The Sherlock Guideline Management System. *Interacting With Computers*, **12**(3). (2000). 281-311.
17. Hammond, J.S., Keeney, R.L. and Raiffa, H. *Smart Choices: A Practical Guide to Making Better Decisions*. Broadway Books, New York, 2002.
18. Henninger, S. A Methodology and Tools for Applying Context-Specific Usability Guidelines to Interface Design. *Interacting With Computers*, **12**(3). (2000). 225-243.
19. Iannella, R. HyperSAM: A Management Tool for Large User Interface Guideline Sets. *ACM SIGCHI Bulletin*, **27**(2). (2002). 42 - 45.
20. Liao, S.H. Expert System: Next Term Methodologies and Applications—A Decade Review from 1995 to 2004. *Expert Systems with Applications*, **28**(1). (2005). 93-103.
21. Luchini, K., Quintana, C. and Soloway, E., Design Guidelines for Learner-Centered Handheld Tools. in *CHI'04*, (Vienna, Austria, 2004), 135-142.
22. Lumsden, J. *Guidelines for the Design of Online-Questionnaires*, NRC, Fredericton, NB, 2005.
23. Lumsden, J., Flinn, S., Anderson, M. and Morgan, W., What Difference Do Guidelines Make? An Observational Study of Online-Questionnaire Design Guidelines Put to Practical Use. in *19th British HCI Conference*, (Edinburgh, UK, 2005), 69-83.
24. Lumsden, J. and Morgan, W., Online-Questionnaire Design: Establishing Guidelines and Evaluating Existing Support. in *IRMA'2005*, (San Diego, USA, 2005), 407-410.
25. Norman, K.L., Lee, S., Moore, P., Murray, G.C., Rivadeneira, W., Smith, B.K. and Verdines, P. *Online Survey Design Guide*, University of Maryland, 2004.
26. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T. *Human-Computer Interaction*. Addison-Wesley, Boston, MA, 1994.
27. Priestley, M., A Wizard for Wizards: Decision Support for the New or Despairing User. in *16th Annual International Conference on Computer Documentation*, (Quebec, Canada, 1998), 98 - 102.

28. Puerta, A. A Model-Based Interface Development Environment. *IEEE Software*, **14**(4). (1997). 40-47.
29. Quintana, C., Krajcik, J. and Soloway, E., A Case Study to Distill Structural Scaffolding Guidelines for Scaffolded Software Environments. in *CHI'02,* (Minneapolis, USA, 2002), 81-88.
30. Robbins, J.E. *Cognitive Support Features for Software Development Tools*, University of California, Irvine, 1999, 231.
31. Robbins, J.E. *Design Critiquing Systems*, University of California, Irvine, California, 1998.
32. Robbins, J.E. and Redmiles, D.F. Cognitive Support, UML Adherence, and XMI Interchange in Argo/UML. *Information and Software Technology*, **42**(2). (2000). 79-89.
33. Schlee, M. and Vanderdonckt, J., Generative Programming of Graphical User Interfaces. in *Working Conference on Advanced Visual Interfaces*, (Gallipoli, Italy, 2004), 403-406.
34. Schonlau, M., Jr., R.D.F. and Elliott, M.N. *Conducting Research Surveys via E-mail and the Web*. RAND, 2002.
35. Shepherd, A. Knowledge-Based Expert Systems: Critiquing versus Conventional Approaches. *Expert Systems with Applications*, **14**(4). (1998). 433-441.
36. Silverman, B.G. Survey of Expert Critiquing Systems: Practical and Theoretical Frontiers. *Communications of the ACM*, **35**(4). (1992). 106-127.
37. Sumner, T., Bonnardel, N. and Kallak, B.H., The Cognitive Ergonomics of Knowledge-Based Design Support Systems. in *CHI'97*, (Atlanta, USA, 1997), 83-90.
38. Tianfield, H. and Wang, R. Critic Systems - Towards Human-Computer Collaborative Problem Solving. *Artificial Intelligence Review*, **22**(4). (2004). 271-295.
39. Umbach, P.D. Web Surveys: Best Practices. *New Directions for Institutional Research*, 2004 (121). 23-38.
40. Winnips, J.C. Support of Active, Self-Reliant Learning via the WWW by Means of Cue Scaffolding. *International Journal of Continuing Engineering Education and Life-Long Learning*, **10**(1-4). (2000). 296-313.

Related Content