A Practical Method for Courseware Evaluation

Aldo de Moor
CommunitySense
Cavaleriestraat 2
5017 ET Tilburg, the Netherlands
+31-13-4564126

ademoor@communitysense.nl

ABSTRACT

As more courseware becomes available, choosing the right functionality for a particular e-learning community is becoming more problematic. Systematic methods for evaluating courseware functionality components in their context of use are required. Of many general methods for ICT evaluation it is unclear how to assess their applicability in the context of courseware. We outline a practical method for courseware evaluation. We experiment with the method by evaluating the courseware functionality used in one core e-learning activity: the making of group assignments. One interesting finding is that the usefulness of an application to a large degree depends on the particular activity being supported, much less on the particular functionality used.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics – *performance measures, product metrics, software science.*

General Terms

Measurement, Performance, Design, Economics, Experimentation, Human Factors

Keywords

Courseware, evaluation, method, information systems quality.

1. INTRODUCTION

E-learning is an increasingly important application of the Internet. Ubiquitous computing and life-long education is rapidly making electronic learning more feasible and accepted. In the educational field, courseware has become an essential means of supporting course-based e-learning, much of it Web-based. A plethora of tools and environments is available. Two main types of courseware applications exist: commercial platforms like WebCT and Blackboard, and open platforms which can be completely or partially open source. A related development is the development of online open content, such as the ambitious Open Courseware initiative in which MIT makes much of its course material available online to the world for free[1].

Courseware offers many functionalities which can be used to support communities of users in their individual and especially collaborative needs. Much information systems research concentrates on developing ever more advanced components for knowledge sharing and learning [4]. However, many design problems arise:

- *Too much (costly) functionality*: functionality goes unused, wasting resources and confusing users. Many sophisticated group file management options are never used, for instance.
- *Gaps between required and available functionality*: some needs go unsatisfied. For example, originally Blackboard permitted only lecturers to remove students from a course, whereas empowering students to manage their own accounts saves everybody much work.
- *Conflicting functionality requirements between different categories of stakeholders*: a prime example is the different requirements of students who want user-friendly and powerful functionality, and computer centers, which need to ensure security and maintainability.

The basic question is: how to make sense of the jumble of requirements, technologies and interests typical of courseware decision making in large, complex institutions like universities? In other words, how to evaluate web functionalities in their context of use? Such context-sensitive evaluation methods that work are important building blocks of the Pragmatic Web.

To answer this question, we first summarize our definitions. *E-learning* is any process of computer-mediated learning. We define *courseware* as technological environments consisting of multiple functionality components, together offering a complete system of information and communication services required for supporting course needs[2]. A functionality is a set of functions and their specified properties that satisfy stated or implied needs[3].

2nd *International Conference on the Pragmatic Web*, Oct 22-23, 2007, Tilburg, The Netherlands.
Copyright 2007 ACM 978-1-59593-859-6 $5.00.

[1] UNESCO's free software portal gives an overview of some typical open educational software, much of it open source: http://tinyurl.com/2roukq/. The MIT Open Courseware portal: http://ocw.mit.edu.

[2] A comparison of the functionalities of many well-known courseware tools: http://www.edutools.info/course/compare/all.jsp

[3] http://www.sei.cmu.edu/opensystems/glossary.html

Functionalities can be studied at different levels of granularity: at the lowest level, we distinguish *systems* of tools or services. The next level consists of the *tools* or services themselves. The next level consists of the *modules* of which the tools or services are composed. Finally, we distinguish the particular *functions* combined in a module [7]. Depending on the particular community in which they are applied, almost identical functionalities are often selected and configured very differently in practice [14]. In making strategic decisions on which courseware to select, the value of its functionality to the users thus needs to be assessed carefully.

Much research on ICT evaluation has focused on exploring the dimensions of evaluation which such methods can take into account, such as trees of quality aspects to be used in the evaluation process [5, 6]. Applying such dimensions, a wide range of methods has been developed to systematically evaluate the quality of information technologies, e.g. [3, 16]. However, these methods vary widely in their evaluation scope, outcomes, and techniques. These generic methods, although useful in theory, are not very applicable in practice, since they do not take into account the *situatedness* of courseware evaluation, determined by the context of use of the tools.

In this article, we examine some of the specific characteristics of courseware evaluation methods in Sect 2. In Sect. 3, we outline our practical method for courseware evaluation, which combines simple indicators assessing the value of the combined ICTs for particular workflow activities with the value of a particular tool functionality for the combined activities. We show how the method was validated in practice by applying it in a group assignment making experiment in Sect 4. We discuss some implications of our approach and end the article with conclusions.

2. COURSEWARE EVALUATION METHODS

A new paradigm of market-based information systems development is emerging. In this model, vendors focus on developing products, not systems. New forms of analysis, roles in software development, and the meaning of use and maintenance therefore need to evolve [20]. To design useful information systems by selecting the right components, available functionalities in the form of modules and services need to be evaluated in their context of use by their communities of use. Choosing the right courseware is not simply a technological decision, however, but a strategic IS development choice. Since the choice of courseware has far-reaching implications for the primary process of an educational institution, its selection requires a careful balancing of the multiple educational, political, and social requirements with the affordances and constraints inherent in the technology [21, 23]. If proper analysis does not take place, the socio-technical gap between work practices and supporting groupware may easily become too large, impeding use and collaboration [1].

When assessing the value of courseware in the e-learning process, we are especially interested in the role that these technologies play in supporting work activities, in other words, finding out whether they are used effectively. Effective use is the capacity and opportunity to successfully integrate ICTs into the accomplishment of self or collaboratively identified goals [10]. Evaluating the usefulness of a functionality can be defined as the evaluation of the extent to which users can translate their intentions into effective actions to access the functionality [8]. Such evaluation should ultimately contribute the purpose of the community of use [18], so that somehow individual assessments must be aggregated and interpreted by relevant stakeholders.

Courseware selection is still too often seen as only a process of software selection based on some simple list of technical features. Many software evaluation methods outline the order of process steps to be taken, such as defining software requirements, investigating the options provided by the various tools, making a shortlist and selecting the final candidate [11]. However, these methods do not provide much help in providing material criteria and procedures for the communal, systemic evaluation of the usefulness of software. In other words, how to assess whether the courseware-in-use serves the purposes of the community?

An interesting class of evaluation methods are so-called portfolio approaches. These take into account the existing infrastructure and architecture of the organization in their weighing and scoring of aspects. One prime example is Bedell's method for ICT investment selection [3]. In this method, ICT functionalities are scored on both their effectiveness and importance for the activities they are to support. The activities themselves are also scored on their importance to the organization. The importance scores act as weights for the effectiveness scores. Several higher-level indicators can be calculated, such as for making decisions on whether the organization should invest in ICT at all. Thus, Bedell's method provides a powerful instrument for aggregating many diffuse evaluations and preparing for decision making

Summing up: courseware evaluation requires evaluation criteria of the right scope and granularity, procedures to calculate more complex metrics out of simpler ones, and approaches to allow for human interpretation to play its appropriate role in the evaluation process. However, most generic methods, such as multi-criteria and portfolio approaches, do not pay enough attention to the larger evaluation process and context in which the (human) decision makers interact with the data [2]. In Figure 1, we present a context model of courseware evaluation, which is at the heart of our evaluation method. Space is lacking here to discuss the model in detail. The reader is referred to [7] for an extended version of this model and the method it informs, which we outline next.

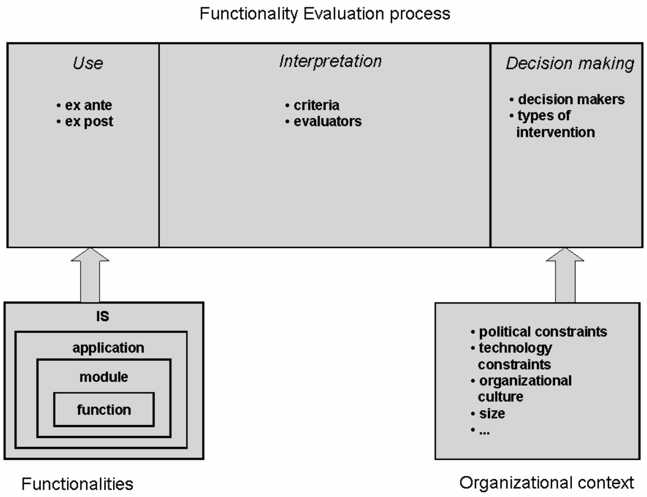

Figure 1. A context model of courseware evaluation

3. A PRACTICAL METHOD FOR COURSEWARE EVALUATION

Our main objective was to design a practical method that - with a minimum of intrusion and effort – could assess the usefulness of functionality modules to courseware users. It should allow various stakeholders to weigh and reach agreement on possibly conflicting evaluations. One application of this method is performing an initial analysis of which functionality modules to acquire, develop or divest from in building a comprehensive courseware information system.

We used Bedell's method [3] as the basis of our own courseware evaluation method. However, we simplified it in that we did not perform some of the higher-order analyses on, for instance, the effect of all tools on the organization level, as such interpretations can be quite complex and abstract to first-time evaluators. Instead, we focused on developing simple measures for the usefulness of courseware.

We were interested in two basic questions: (1) how well are our course activities supported by the various courseware functionality components? (2) how much are the various functionality modules used? To this purpose, we developed two indicators: *activity* and *functionality scores*.

3.1 Activity and Functionality Scores

Activity scores show how useful the combined functionality components are for the support of a particular activity. Functionality scores represent the usefulness of a particular functionality component in supporting the combined activities of a community.

The basic elements to be scored by the users are:

- *I(a)* = importance of an activity
- *I(f,a)* = importance of a functionality in supporting a particular activity
- *Q(f,a)* = quality of a functionality in supporting a particular activity. This measure can include aspects such as efficiency, user-friendliness, to look-and-feel, etc.

The elements are scored on a ten point scale, with 10 being highest. Examples of these indicators could be, respectively, I(Information Collection) = 9, meaning that Information Collection is very important to the group, I(Virtual Chat, Information Collection) = 4, implying that chat support is not so important to the users for collecting information, and Q(File Transfer, Submission of Results) = 8, which suggests that the particular file transfer module used works well for result submission.

Activity scores

A-Score = Σ I(f_i,a) * Q(f_i,a), for all functionalities 1..i.

For all functionality components f_i supporting a particular activity *a*, the experienced quality of the support they provide is multiplied by their importance in supporting this activity. The sum of these values measures the usefulness of the combined technologies for a particular activity to the scoring user. This measure is especially useful for technology users, such as lecturers and students.

Functionality scores

F-Score = Σ I(a_j) * I(f,a_j) * Q(f,a_j), for all activities 1..j.

For all activities a_i supported by a particular functionality component *f*, the quality of support provided is multiplied by its importance for this activity *and* by the importance of the activity. This last multiplication is necessary, as support provided by the component to more important activities should weigh more than to less relevant activities. The sum of these values measures the usefulness of a particular functionality component for the combined activities of the scoring user(s). This is especially useful for technology maintainers and developers, such as computer centers and software vendors, in order to determine which components to acquire, develop, or to remove.

4. AN EXPERIMENT: THE EVALUATION OF GROUP ASSIGNMENT FUNCTIONALITY

Tilburg University heavily promotes the use of Blackboard: almost all courses are mandatorily supported by this software. However, a few years ago, the university computer center received many complaints about the functionality from students and lecturers, ranging from awkward interfaces to very low performance. Furthermore, given the rapidly increasing prices of commercial courseware software, the center was investigating the possibilities of (partially) open source courseware at the time. Moreover, one of their strategic objectives was to find better ways to collect, classify, and handle user requirements. The computer center was therefore very interested in experimenting with a practical evaluation method.

4.1 The Design of the Experiment

An initial experiment was set up in which students taking the 2002 course Quality of Information Systems would use the method to evaluate the quality of Blackboard for the making of group assignments. This experiment was repeated in the 2003 course, to compare results and see if discovered patterns would hold over time with different groups of students. In 2003, a new version of Blackboard was used, but most of the new functionality was for better course management by the lecturer and did not affect the functionality experienced by students. Furthermore, to compare not just the usefulness of one tool over time, but also the similarities and differences between tools, in the 2003 course another courseware tool, CourseFlow, was analyzed as well. CourseFlow was custom-developed earlier by one of the students for another institute of higher education and contained most of the basic functionalities provided by Blackboard. To ensure a valid comparison, the functionality modules evaluated in the 2002 experiment were also tested in the 2003 experiment, both in Blackboard and CourseFlow. To allow students to develop sufficient experience with CourseFlow, this tool was used as the only courseware in the 2nd half of the course, prior to the evaluation assignment.

The population consisted of 2nd year Information Management students, most of whom had at least one year experience using Blackboard prior to this course. Information management students are good candidates for such an evaluation experiment, as they are used to thinking in terms of assessing the organizational usefulness of technical functionality. For these future architects of information systems, it was motivating that their assessments were taken seriously in strategic software acquisition and development planning. At the beginning of each course, the students were divided into groups of 4 persons each (2 groups in each year consisted of 3 students. This, however, should not significantly have affected the quality of group judgment, as the students in all groups had a similar background and experience). The groups had to do several assignments during the course, one of which was this evaluation experiment. The 2002 experiment counted 62 students in 16 groups, the 2003 course 46 students in 12 groups. In the 2002 course, all 16 groups had to score Blackboard, in the 2003 course, 6 of the randomly assigned groups analyzed Blackboard, the other 6 evaluated CourseFlow. The software manager of the computer center was involved from the start. He promoted the importance of the project to the students, gave feedback on the results, and promised to use the results in strategic software planning.

One of the main – and complex - workflows to be supported by the courseware, was the making of group assignments. The purpose of the experiment was to find out (1) how well the various activities that comprise the group assignment making process were supported by the technical system as a whole, and (2) how useful the various functionality components were considered to be.

The group assignment making process was subdivided into four *activities*:

- *Information collection*: the retrieval of the assignment, relevant literature, and the standard answer sheet.
- *Group discussion*: the communication within the group about the planning of the assignment and the division of tasks.
- *Submission of results*: the submission of the finished assignment, including notification of teaching assistant.
- *Feedback from peers*: the review of each submitted assignment by at least one other group and related comments obtained from other students.

The following 11 *functionality modules* were scored:

- *General communication*: Send E-Mail, Discussion Board, Virtual Chat, Student Roster
- *General information*: Announcements, Course Information, Course Documents, Assignments
- *Group*: Discussion Board, Virtual Chat, File Transfer

Student groups had to score the importance of the various activities, and both the importance and quality of a particular module in supporting a particular activity. Additionally, they were to give short textual motivations (given their diversity and the lack of space, these are not included here, but have been used in the interpretation of the quantitative data by the software manager). Participation by students was high, and resulted in detailed responses. Informal feedback showed that many students thought it was a useful and valuable exercise.

4.2 Results

The data obtained from the evaluation are summarized in Figures 1-4. Note that the 2003 scores in Figure 2 and 3 are averaged over both tools. The comparison between Blackboard and CourseFlow in 2003 is given in Figures 4 and 5. The scores for both years and tools are remarkably similar. An explanation of interesting differences will be given below.

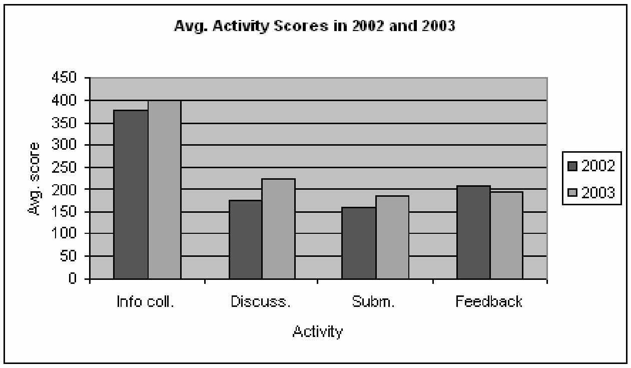

Figure 2. Average activity scores for the group assignment making process in 2002 and 2003

Figure 2 shows the average activity scores for the 2002 and 2003 experiments. These activity scores indicate that both cases, overall functionality was perceived as most useful for information collection. In their textual explanations, most student groups indicate that easy, portal-style, access to the various materials needed to properly make the assignment was most important to them. This was also advertised by the vendor as one the key strengths of the Blackboard functionality. The relative usefulness of Blackboard for information collection can further be explained by discussion and feedback being relatively unimportant in the process model of group assignment making used, and the fact that assignment submission could also easily be done by ordinary e-mail.

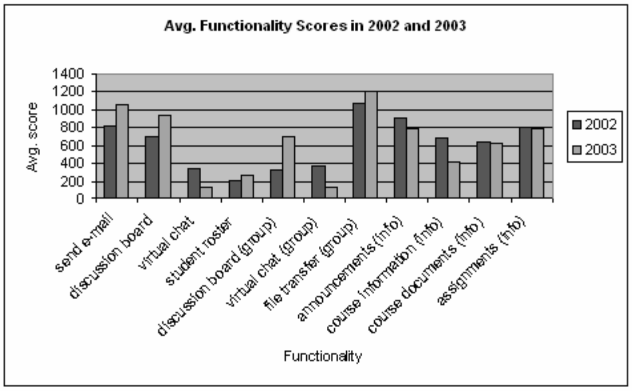

Figure 3. Average functionality scores for the group assignment making process in 2002 and 2003

The functionality scores for 2002 and 2003 are summarized in Figure 3. In 2002, these scores and their textual motivations were interpreted by the software manager of the university computer center in the following way:

- Especially the basic functionality of Blackboard (File Transfer, Announcements, and Send E-mail) was considered important by students.
- File Transfer, however, was not implemented well (according to textual comments. Its high score is thus due to importance, not quality). An alternative would be looked for by the computer center. The E-Mail functionality provided by Blackboard was only very basic, external e-mail applications were better suited, but not (yet) integrated in the platform.
- The Student Roster and Virtual Chat components were not considered important at all. Explanations were, respectively, that there already was an electronic study guide with better functionality, and that external chat-tools such as MSN were much preferred by students as chat functionality.
- Applications like Blackboard and external chat tools should be more open in their integration of functionality modules.
- Open source courseware could be a valuable addition in the future. Given that the university already made a large investment in Blackboard licenses and training, a major transition was not likely to take place soon. Still, experimenting with specific components in order to allow a possible partial migration in the future was supported. For example, Tilburg University was developing its own suite of survey-tools, called UvTLAB, as the survey-module provided by Blackboard was considered insufficient.
- The software manager made a similar interpretation of the 2003 results. Two relatively large differences can be seen in the scores for the virtual chat and the group discussion board. The virtual chat scored even lower in usefulness in 2003 than in 2002, probably because the use of chat tools had become even more widespread in the meantime. The group discussion board, on the other hand, scored significantly higher in 2003. One reason was that students perceived the quality of the group discussion board functionality of CourseFlow to be much higher than that of Blackboard. This was in line with the development goals of CourseFlow, which stated offering advanced and user-friendly discussion functionality as one of its prime objectives.

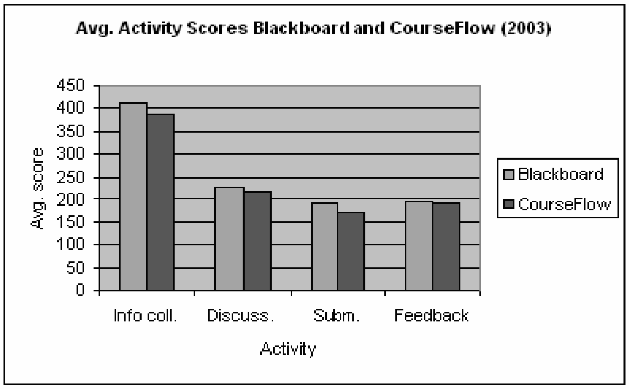

Figure 4. Average activity scores for the group assignment making process for Blackboard and CourseFlow (2003)

When comparing the 2003 data by the applications Blackboard and CourseFlow, we see a very close similarity between both tools. The activity scores (Figure 4) are almost similar and follow the same overall pattern as the comparison between the 2002 and 2003 results, with information collection by far being the activity for which the provided support is considered to be most useful. The similarity in scores suggests that activity is a much more important determinant of usefulness than the specific version or tool used to support it.

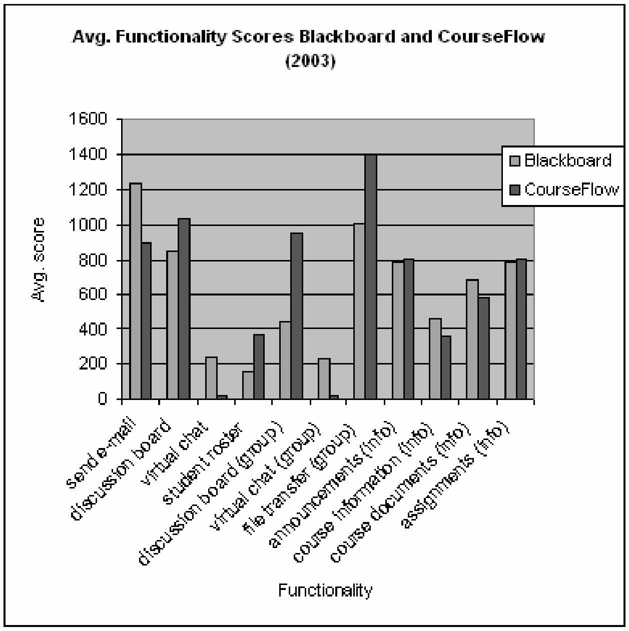

Figure 5. Average functionality scores for the group assignment making process for Blackboard and CourseFlow (2003)

The functionality scores of both tools show somewhat more variation than the activity scores (Figure 5). As mentioned before, CourseFlow scored considerably higher on the group discussion board functionality. It also scored considerably higher on student roster and file transfer functionalities, which were other design goals of CourseFlow, as it included more advanced knowledge management functionalities than Blackboard. Only on the virtual chat, CourseFlow scored even lower than Blackboard.

5. DISCUSSION

Courseware is a key technological infrastructure for enabling e-learning. However, much courseware functionality does not satisfy user requirements. One solution are component methodologies, in which tailored educational applications are constructed out of many small components of functionality [19]. Still, to make a relevant composition of modular applications, one must assess their experienced usefulness [12].

Our approach can be seen as a form of context-bound evaluation. Many traditional evaluation approaches are based on standard checklists of features that have little to do with the particular characteristics of courseware-in-use. Furthermore, such approaches make unreasonable assumptions about the level of detail at which decision makers are able to make their preferences explicit in practice [15]. Only context-bound evaluation, in which the experiences of the audience with the system in actual use are assessed *in toto*, can do full justice to the complexity of interactions of learners and their courseware [13]. Thus, for example, students can be asked to discuss in their group their joint overall experience with a particular functionality for a particular activity, giving them the freedom to define (or keep implicit) and weigh their own criteria, instead of having to check off an artificial list of characteristics that may not be relevant to their own subtle and complex work practices.

A combined use of quantitative (i.e. the scores) and qualitative indicators (i.e. the textual comments) aids complex decision making processes [17]. We found that simple measures are more insightful for initial quick scan and discussion purposes than the sophisticated measures proposed by Bedell [3].

E-learning being such a complex and core activity requires the organization itself to become adaptive, and to develop not only its technical skills, but also aim for increasing its *process learning* capabilities [22]. Such a learning organization is never finished, but continuously improves its way of working [9]. Practical courseware evaluation can be instrumental in assessing the value of learning environments, identifying functionality usefulness issues, developing solutions across applications, and helping to align the evaluation and decision making positions of the various stakeholders. Courseware evaluation can thus become a powerful catalyst in the creation of a true ‘e-learning organization’.

6. CONCLUSIONS

We proposed a practical method for evaluating the usefulness of functionality components by various groups of stakeholders. An experiment was done by evaluating a core course process, the making of group assignments. The obtained activity and functionality scores were useful for initial courseware functionality selection.

One interesting finding is that the usefulness of a courseware application heavily depends on the particular activity being supported, although by which specific version or tool hardly seems to matter. This suggests that, contrary to the claims of commercial courseware providers, open source software might do just as well (or badly), but at a fraction of the cost. Also, there is great variation in the perceived usefulness of individual functionality components, so that a precise selection of which components to acquire or develop should be possible.

In sum, practical courseware evaluation is important, but not trivial. In this article, we have explored some of the many design choices to be made in developing simple methods for complex evaluations. We ignored many of the evaluation complexities and methodological tradeoffs still to be made, but have tried to make a case for methods that matter.

7. ACKNOWLEDGMENTS

The author would like to thank Corno Vromans and Jaap Wagenvoort for their enthusiastic participation in and assistance during the experiments

8. REFERENCES

[1] Ackerman, M. S. (2000). The Intellectual Challenge of CSCW: the Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction, 15*(2), 179-203

[2] Bannister, F., & Remenyi, D. (2000). Acts of Faith: Instinct, Value, and IT Investment Decisions. *Journal of Information Technology, 15*, 231-241.

[3] Bedell, E. (1985). *The Computer Solution: Strategies for Success in the Information Age*. Homewood, Ill.: Dow Jones-Irwin.

[4] Bieber, M. et al. (2002). Towards Knowledge-Sharing and Learning in Virtual Professional Communities. In *Proceedings of the 35th Hawaii International Conference on System Sciences* (Hawaii, January 5-7, 2002).

[5] Delen, G. P. A. J., & Rijsenbrij, D. B. B. (1992). The Specification, Engineering, and Measurement of Information Systems Quality. *Journal of Systems and Software, 17*, 205-217.

[6] DeLone, W. H., & McLean, E. R. (1992). Information Systems Success: The Quest for the Dependent Variable. *Information Systems Research, 3*(1), 60-95.

[7] De Moor, A. (2007). The Pragmatic Evaluation of Tool System Interoperability. In *Proceedings of the Second Conceptual Structures Tool Interoperability Workshop (CS-TIW 2007)* (Sheffield, UK, July 2007). Research Press International, Bristol, UK, 1-19.

[8] Gaines, B., Lee, L. C., & Shaw, M. (1997). Modeling the Human Factors of Scholarly Communities Supported through the Internet and the World Wide Web. *Journal of the American Society for Information Science, 48*(11), 987-1003.

[9] Garratt, B. (2000). *The Learning Organization: Developing Democracy at Work*. London: HarperCollins.

[10] Gurstein, M. (2003). Effective Use: a Community Informatics Strategy Beyond the Digital Divide. *First Monday, 8*(12).

[11] Hendrickson, E. (1999). Evaluating Tools. *Software Testing & Quality Engineering Magazine*(Jan/Feb), 39-42.

[12] Hiltz, S. R. (1998). Collaborative Learning in Asynchronous Learning Networks: Building Learning Communities. *In Proceedings of WEB98* (Orlando, Florida, November 1998).

[13] Hosie, P., & Schibeci, R. (2001). Evaluating Courseware: A Need for More Context Bound Evaluations? *Australian Educational Computing, 16*(2), 18-26.

[14] Kling, R., McKim, G., Fortuna, J., & King, A. (2000). Scientific Collaboratories as Socio-Technical Interaction Networks: A Theoretical Approach. In Proceedings of *The 2000 Americas Conference on Information Systems* (Long Beach, CA, August 10-13, 2000).

[15] Kurtz, C. F., & Snowden, D. J. (2003). The New Dynamics of Strategy: Sense-Making in a Complex and Complicated World. *IBM Systems Journal, 42*(3), 462-483.

[16] Parker, M. M., Benson, R., & Trainor, H. (1988). *Information Economics: Linking Business Performance and Information Technology*. Englewood Cliffs, NJ: Prentice Hall.

[17] Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data Quality Assessment. *Communications of the ACM, 45*(4), 211-218.

[18] Preece, J. (2000). *Online Communities : Designing Usability, Supporting Sociability*. Chichester, New York: John Wiley.

[19] Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., & Suthers, D. (1999). Developing Educational Software Components. *IEEE Computer, 32*(9), 2-10.

[20] Sawyer, S. (2001). A Market-Based Perspective on Information Systems Development. *Communications of the ACM, 44*(11), 97-102.

[21] Talbott, S. (1995). *The Future Does not Compute : Transcending the Machines in our Midst*. Sebastopol, CA: O'Reilly & Associates.

[22] von Krogh, G., Nonaka, I., & Aben, M. (2001). Making the Most of Your Company's Knowledge: A Strategic Framework. *Long Range Planning, 34*, 421-439.

[23] Werry, C. (2001). The Work of Education in the Age of E-College. *First Monday*, 6(5).