Market-based Adaptive Discussion Forums *

Natalia López, Manuel Núñez, Pablo Rabanal,
Ismael Rodríguez, and Fernando Rubio

Dept. Sistemas Informáticos y Programación
Facultad de Informática
Universidad Complutense de Madrid
C/. Profesor José García Santesmases s/n
28040 – Madrid. Spain
e-mail:{natalia,mn,isrodrig,fernando}@sip.ucm.es

Abstract. One of the most successful and simplest Internet-based systems to promote the spread of knowledge are discussion forums. Unfortunately, not much research has been done to improve them by introducing adaptive capabilities. In this paper we discuss a market-oriented mechanism to promote the knowledge exchange activity in discussion forums. Our system provides a strategy to dynamically adapt the structure of the forums by using statistics about users behavior. By doing so, we minimize the response time in which questions are satisfactorily answered. In addition to that, the effort needed to generate useful information is also reduced.

Key words: E-learning, discussion forums.

1 Introduction

The field of e-learning is reaching a certain degree of maturity. Different aspects are the responsible for this improvement in e-learning. On the one hand, the new technologies and programming tools allow to develop more complex applications. On the other hand, the wide implantation of the Internet has produced an increasing demand for this kind of systems. A notable advance in the creation of intelligent tutoring systems was produced when artificial intelligence techniques appeared in the development of this kind of systems. Good examples of these new generation tutoring systems are [16, 7, 21, 22, 14, 13, 1]. These interactive learning environments allow students to change from passive observers to active constructors of knowledge [2], by favoring *active* learning [9, 8]. Besides, by using adaptive hypermedia techniques (e.g. [5, 11, 23, 6, 4]) when developing intelligent tutors, we have systems that automatically adapt themselves according to the responses of students (e.g. [21]).

* Research partially supported by the Spanish MCYT project TIN2006-15578-C02-01, the Junta de Castilla-La Mancha project PAC06-0008-6995, and the Marie Curie project MRTN-CT-2003-505121/TAROT.

One of the simplest and most classical, but also extremely successful, electronic mechanisms used for e-learning are *discussion forums* and *news groups*. By using these applications, the knowledge about a specific topic can be easily shared. In fact, thanks to the collaborative action of many users, doubts and questions can be solved by their mates, as the more experienced users help beginners by transmitting them their knowledge.

Unfortunately, the cooperation mechanisms appearing in classical discussion forums are still somehow primitive. In fact, they do not usually promote the efficient generation of information. First, the incentives to share information by answering questions of other users can be weak. In most cases, the altruism or the desire of prestige is the main motivation to help beginners. Unfortunately, such motives could be not shared by many users. Thus, the collaboration degree can be quite asymmetric. Second, experts can lose their motivation to help others as they can get saturated by a huge amount of questions. Even though most of the problems could be solved by not so skilled users, more experienced ones receive all the questions.

The key to understand the problem underlying discussion forums is to consider the *generation* of information as a *scarce resource*. More precisely, if we consider the cost of the generation of information in terms of the time needed to create it, then it becomes clear that such activity must be considered a scarce resource. Obviously, the costs to copy, store or transmit information are negligible compared to the cost of generating it. So, we will only consider the aspects related to the generation of information.

Since the efficiency in the generation and use of information in a discussion forum can be defined in terms of optimizing the access to a scarce resource, there exists a science that can help us dealing with it. Quoting from [18], "*Economy is the science that deals with distributions of scarce resources which can be valued by having different utilities by different users.*" In our concrete problem, the effort for generating information is clearly a scarce resource, and the utility that each user of the system gives to each piece of information is also clearly different. As an example, let us consider two different questions performed by two different users. The first of them is more complex than the second one. In fact, only experts can answer such a question, while the second one can be answered by a user with only some command in the topic. In that situation, it is clearly inefficient to ask an expert to answer the easiest question and to assign the hardest question to a user with only some command on the corresponding topic. The most usual method in economic systems to promote the efficient utilization of resources consists in using *prices* for them. These prices define the effort level that an interested agent should apply to obtain a certain resource. If a resource is scarce (supply lower than demand) then the price raises, so only the most interested agents can obtain the desired resource. Moreover, a high price in a resource encourages producers of the resource to increase its production. Thus, the difference in prices among the different resources represents the real necessity of each product as well as promotes the production of the items most estimated by the users.

As it can be expected, the most common unit to measure the effort level to be paid is *money*. However, in order to promote an efficient access to resources it is not necessary to use *real* money: It can be virtual money provided that its quantity is scarce. In fact, several market oriented solutions have already been used to regulate the access to information in databases or the processing time in distributed systems (see e.g. [20, 19, 17, 10, 3]). We claim that a market-oriented approach could also be applied by an intelligent autonomous agent to control the generation of information in a discussion forum. In this way, each user should have to *somehow* pay other users in order to get her doubts solved. However, a direct application of the market oriented concepts could not be completely satisfactory, as e-learning environments need to consider additional factors not appearing in other computational systems. In this paper we continue our previous work [15] in order to deal with these important differences and propose a suitable market-oriented methodology to optimize the use of the educational effort in discussion forums. The system has been implemented and our experience indicates that discussion forums may benefit from the ideas presented in this paper.

The structure of the rest of the paper follows. In Section 2 we introduce the main ideas of our market-oriented methodology. Then, in Section 3 we show how discussion forums can be dynamically adapted to improve their efficiency. Afterwards, in Section 4 we present the implementation of the system. Finally, in Section 5 we present our conclusions.

2 Improving the Basic Structure of Discussion Forums

The current section describes how a market-oriented methodology can be applied to optimize the performance in a discussion forum. In particular, we present the main ideas to provide incentives to the users.

2.1 Paying for Good Answers (without Money)

First of all, let us remark again that the utility obtained by receiving the answer to a question (strongly) varies from user to user. Thus, the number of different resources may be, eventually, equal to the number of users, so it can be huge. In this case, the system will not be able to decide, in general, which user is the best choice to answer a concrete question. To overcome this difficulty, the system should dynamically *classify* users based on the *quality* of their effort generating answers. By using classes of users at different levels, the system can determine the suitability of a given user to answer a question of another given user. Let us remark that the quality of the effort of the generation of information does not only depend on the knowledge of the users, but also on their willingness to transmit their knowledge.

It is important to point out that, since we are dealing with knowledge, it is impossible for a user to know with full certainty whether her question requires the help of an expert, or only that of a not so skilled user. Thus, not much

improvement of the efficiency could be obtained by allowing the questioner to choose the type of user that should answer her doubt. On the contrary, the system must provide a mechanism of distribution of questions to decide which type of users should answer each type of questions.

Another important point is the fact that the quality of the resources, that is, the quality of the received answers, is not known a priori. In other words, a user asking a question does not know how good the answer is until she obtains it. In fact, she can get disappointed with an incomplete/erroneous/bad answer. Analogously, a user answering a question cannot know the quality of her answer until the other user studies it. Thus, it is not possible to set the price of an enquiry a priori. Actually, the receiver of the resource should be the one who fixes the price she will pay. Obviously, if money is the scarce resource that enables access to information then the buyer of the resource will try to pay as few as possible, arguing that she was not satisfied with the answer. This problem can be overcome if the paid *money* does not perceptibly modify the future capacity to *buy* new knowledge. In order to do so, the units paid by users will be potentially *inexhaustible*, as we will comment below.

The main reason for including prices in our approach is to give incentives to the providers of knowledge. Thus, when a user gets satisfied with an answer then the provider of the answer should receive an incentive. As the only resource available in the system is the educational effort, such incentive should be given in terms of it. Since the users will not be allowed to choose the type of users to answer their doubts, the system will have to appropriately award incentives to users, both by providing and by restricting the access to educational effort. If a user provides answers that are positively perceived then:

- The system will try to help the user to get her future doubts solved. As she has proved that she has valuable knowledge, her future questions will be shown to more expert users. Thus, she will subsequently obtain satisfactory answers with a higher probability.
- As she has proved that she has valuable knowledge, the system will also try not to show her easy questions in the future. By doing so, she will be able to save her effort for those questions that really require her skills.

In order to obtain such behavior, users will be split into classes according to their education efforts in the past. Each time a user posts a new question, it will be initially shown only to the users belonging to her own class. If after a period of time (depending on the class) the question is still unanswered then it will be forwarded to the next level. If the question is not answered there, it will be forwarded again and so on until reaching the highest level. So, easy questions will be usually answered before they reach the expert level. This is so because, as we remarked before, the user providing an answer to a question may improve her *reputation* by getting into a *better* group.

Note that, by using this mechanism, we encourage that the questions are answered by those users that are the most appropriate to do it: Users with medium command answer the questions of beginners while experts answer both the questions of the medium users and the ones from beginners which could not

be solved by medium users after the corresponding delay. That is, questions are answered by the less skilled users who can correctly answer them. Therefore, the scarce resources of the system (that is, the effort to create valuable information) are exploited in an efficient way, keeping the most valuable resources for the specific questions that really need them.

Let us remark that the underlying idea of this method consists in restricting the access to educational effort while not restricting the access to the information itself. In fact, in our implementation anybody have access to the full repository of questions and answers (moreover, standard SQL queries are available for searching the appropriate information).

2.2 Structuring Users in Classes

Many discussion forums (e.g. [12]) allow users to be structured in different classes. However, this distribution usually depends on the interest of the users on different topics. In our discussion forum, the distribution of users among classes does not depend on their interest on specific topics, but on their previous effort generating information for other users. Each group is a set of sorted users (according to the amount of gained points). So, the set of *guru* users is made of the u_1 best users of the ranking, the set of *expert* users is made of the next u_2 users, and so on. As it usually happens in knowledge communities, the amounts of users in each class should follow a pyramidal structure. Thus, the condition $u_1 < u_2 < \ldots < u_n$ will be considered.

Let us remark that by structuring classes through the ranking system a user can, after some time, either improve or fall in the hierarchy. For instance, in the case that a user reduces her activity, she will be overtaken by other more active users. Therefore, even if the points each user owns to acknowledge other users are inexhaustible, these points can actually affect her. Let us remark that, however, the effect of acknowledging others has a minimal repercussion if we consider them *individually*. In a system with thousands of users, giving points to some other specific user will not have, in general, a perceptible effect on the ranking of the specific user who gives them. Nevertheless, the result of the simultaneous delivering of points of all the users of the system yields a competitive environment as a whole.

Let us note that, by using this method, the incentive for a user to reduce the points she gives to others is very low, because what can actually constrain her access to the resources is mainly the activity of *others*. This is specially clear if we compare this method with a method where the units each user delivers are *exhaustible* and each user is forced to pay fixed amounts of units before accessing the information provided by another user. Let us also note that structuring classes by fixed intervals of points (for example, users in the range 1000–1500 are located in the *medium* class) leads to the same competitive behavior: even if the points cannot decrease and degradation is impossible, quality of the accessed information effort can fall. The reason is that what is really important about being in a given class is not the name of the class but the set of mates one has in the class.

It is important to remark that the distribution of users among classes is dynamically done by taking into account their behavior. However, when creating a forum for the first time there is not information about the previous behavior of the users. In that case, the administrator of the forum can choose between assigning the same initial level to all the users, or creating some initial classes based on *her* knowledge about the users. Note that, in many situations, the administrator can have information about the skills of some of the users. For example, when creating a new forum on a specific research topic, the set of guru users can be easily obtained by considering the publications of the users about this concrete research topic.

3 Adaptive Capabilities of the System

In this section we briefly comment how to adjust certain parameters in order to minimize the mean time required to satisfactorily solve the questions. The interested reader can find more details in [15]. The main parameters to be dynamically adjusted are the *waiting times*, that is, the times questions need to stay unsolved at each level before forwarding them to an upper level. We will denote by t_{ij} the time that questions of users of level i remain in level j before forwarding them to level $j+1$. Let us remark that these values should be neither *high* nor *low*. On the one hand, if they are very high then the response time will unnecessarily increase: If a question is not solved in a reasonable time then it will probably not be solved in any time. On the other hand, if the time is very low then the probability to receive an answer at each level is very small. Hence, upper levels would receive more questions that needed. In fact, the frequency of questions can saturate the upper levels, reducing the overall efficiency of the system. In this case, the response times would be increased as well. In the rest of this section we comment how to compute appropriate *intermediate* values for the waiting times.

The factors we will use to optimize waiting times t_{ij} will be controlled by users statistics. These data can be automatically obtained by the system without affecting the normal performance of the application. Let us suppose that there are n levels of users in the system. Then, for $1 \leq i \leq n$ we need to compute the frequency f_i of generation of questions of users belonging to level i. Besides, we use a set of random variables to codify the time needed for questions to be correctly answered in a given level. That is, discrete random variables ξ_{ijk} describe the time required for a question coming from users of level i to be solved by users of level j, assuming that the frequency of questions arriving at level j (both from level j and other lower levels) is k. Let us remark that the frequency of incoming questions influences the response time. This is so because a high value implies that users of class j will waste a lot of time just reading the upcoming questions. So, their efficiency answering questions will decay.

By considering the previous data, we can compute the probability p_{ijkt} that the random variable ξ_{ijk} takes a value less than or equal to t, that is, p_{ijkt} will represent the probability that t units of time have passed. So, the following

equality holds: $p_{ijkt} = Prob(\xi_{ijk} \leq t) = \sum_{x=0}^{t} f(x)$, where f is the probability density function associated with ξ_{ijk}. In the following we will denote by p_{ijk} the probability of answering in level j a question initially placed at level i, assuming an input frequency k. That is, p_{ijk} stands for $p_{ijkt_{ij}}$.

By considering the previous random variables, the real frequency of questions arriving at each level can be computed. Basically, the level i will receive those questions generated at that level as well as those questions coming from lower levels that have not been successfully answered in the previous levels. Thus, if we denote by g_i the frequency of questions arriving at level i then we have $g_i = \sum_{1 \leq j \leq i} f_j \cdot \prod_{j \leq k < i} (1 - p_{jkg_k})$. Note that g_i depends on other values g_k, so it is defined in a recursive fashion. This recursion is well founded because when g_i depends on g_k we have $k < i$.

Finally, in order to compute the mean response time for users of level i, we add the mean times needed to obtain the answers at each level from i to $n+1$, weighting this time with the probability that the question is satisfactorily answered at this level. Due to lack of space we do not show the formula computing it, but it can be found in [15]. Then, the problem of deciding the best *waiting times* is reduced to obtaining the best values to optimize the formula.

Let us finally remark that it is also interesting to reduce the variance of response times among different levels. That is, it is interesting to try to obtain similar mean response times in all the levels. One could argue that if the mean response times are the same for all classes then it is not relevant the class a user is located in. So, there is no motivation to answer the questions of others. However, this is false. If the variance is low, it means that the *mean* response times are similar for all classes. In each class, the considered response time is the mean time for all the members of the class. So, if a user inside a class learns and improves her skills, there will be a time when her questions will be harder to be answered. As a whole, the mean response times of the class could remain the same, but her *own* response times could grow. So, the only way for this user to keep her response times will be to pass to the next class. Therefore, the incentives for users to answer questions remain even if the variance is low. That is, a reduction of the variance of response times does not eliminate the incentives for users to improve their classes.

4 Implementation Details

In this section we describe our implementation of the system. In particular, we will comment on the technologies we have used during the development of the system. Moreover, we will also describe the different views available in the system. First, let us consider the technologies used to construct the implementation. In order to improve the portability and reusability of the system, open source systems have been used as much as possible. More specifically, we have used the following:

PHP: Most web pages of the system have been implemented by using this interpreted language.

Fig. 1. User point of view of the forum

Apache: Our server application uses it to serve the web pages.

MySQL: All the information available in the discussion forum is managed by using it.

QK SMTP Server: We have used it as smtp server to send e-mails directly from the local host to the mailboxes of the receivers.

JavaScript: This programming language - interpreted by the browser - was used as complement of PHP to implement certain peculiarities of the discussion forum.

After introducing the basic technology used in the implementation, now we will concentrate on the views provided to the user. Thus, now we briefly present the main screens of the application interface. Moreover, we will outline the behavior of the hierarchical forum. First, let us focus on the view shown to normal users. Before granting users with the capability of posting messages, a welcome screen requests for a registered user name to enter the forum. As usual, a login and a password are required. If the user does not posses a registered user then she may check in at the system by accessing the *Get Registered* functionality. As it is usual, we also offer the possibility to *Remember Password*. Let us remark that a help system is always available for the user. In particular, it explains how to create a new account, how to solve problems with a previously created account, etc.

Once within the forum (see Figure 1), the user can change the personal data associated with her account, or she can actually uses the forum. In this case, she can access the themes exposed on the forum by pressing on any of the titles exposed in the tree. She can also search for specific topics (see Figure 2), create new questions, answer previously created questions, etc. It is also possible to sort the messages according to desired criteria (date, level, author, etc.). Besides,

Fig. 2. Searching messages in the forum

the user can search messages by using either fast queries (by title, author or contents) or advanced queries. In addition to the previous options, the user can also request to receive an e-mail every time an answer to a specific topic is posted in the forum. By doing so, we provide an alternative way to check the status of a given question.

Now we focus on the application view for the administrator of the system. Although the system can adapt itself automatically, we also allow the administrator to monitor the evolution of the system. In particular, she receives alerts each time a relevant change happens in the system. The available options include reading the help system, removing users and discussion threads (to protect the system against illegal or undesired uses), adjusting system variables, and obtaining statistics about the use of the system. Let us briefly describe the most relevant features:

Queries The most typical queries needed by the administrator are predefined (questions accomplished in the later month; registered users; percentage of answers not punctuated yet; requesting consultations on given punctuations; etc.) Moreover, the system also allows to introduce any SQL query against the database. By doing so, we provide a great flexibility to the administrator, as she can access any information available in the system.

Graphics The system also allows the generation of special graphics to improve the readability of certain statistics. In particular, predefined special graphics include the following: Number of users per level (see Figure 3, left); number of messages per level; average days to punctuate a question per level; comparison between the questions and the answered questions (see Figure 3, right); average number of days to answer questions; and number of days to answer each question.

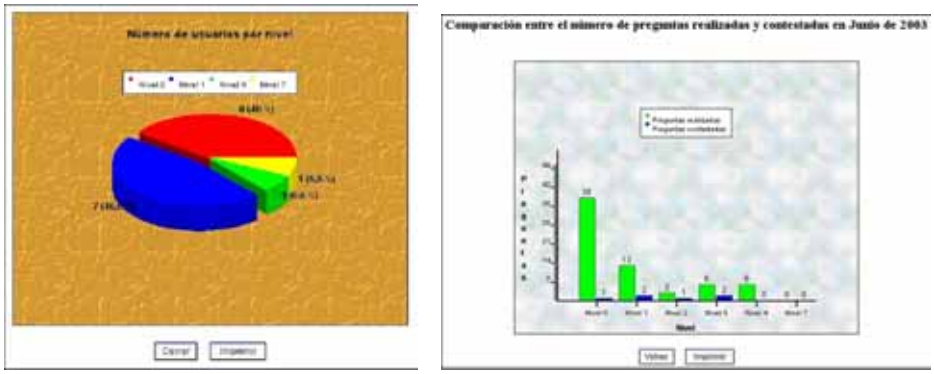

Fig. 3. Graphical statistics provided by the system

Help In addition to the standard help pages of the system, the system also provides help on the structure of the database's tables. Let us remark that this information is very important to allow the administrator to use SQL queries against the system database: In case the administrator has no information about the internal structure of the database, she cannot take profit of all the advanced queries available by using SQL queries.

5 Conclusions

A market-oriented implementation has been presented to improve the performance of discussion forums. The main idea underlying our tool is that questions should be answered by the less qualified users who are still able to solve the problem. By doing so, the effort of more experienced users is saved, so that they can use it for answering those questions that really require their advance knowledge. In order to achieve this objective, users are dynamically distributed in groups according to the capabilities that they have shown by answering the questions of other users. If a user has a new question then she initially posts it in her group. If after a certain amount of time nobody answers the question then the question is moved to a higher-level group. Let us remark that the system dynamically changes the group of each user by taking into account the perceived quality of their answers.

We would like to point out that we have already provided an implementation of a discussion forum based on the ideas presented in the paper. In fact, in the previous section we have commented some of the characteristics of the system we have implemented. In particular, we have described both the technologies we have used and the main functionalities we provide.

Acknowledgments

The authors would like to thank Javier Fraile and Tomás Muñoz for their contribution in the implementation of the system. The authors would also like to thank the anonymous referees for valuable comments on previous versions of the paper.

References

1. E. Aïmeur, F. S. M. Onana, and A. Saleman. Sprits: Secure pedagogical resources in intelligent tutoring systems. In *Intelligent Tutoring Systems, ITS'06*, volume 4053 of *Lecture Notes in Computer Science*, pages 237–247. Springer, 2006.
2. T. Angelo. The campus as learning community: Seven promising shifts and seven powerful levers. *AAHE Bulletin*, 49(9):3–6, 1997.
3. O. Ardaiz, P. Artigas, T. Eymann, F. Freitag, L. Navarro, and M. Reinicke. The catallaxy approach for decentralized economic-based allocation in grid resource and service markets. *Applied Intelligence*, 25(2):131–145, 2006.
4. P. D. Bra, Ad.Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! the adaptive hypermedia architecture. In *HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 81–84. ACM Press, 2003.
5. P. Brusilovsky. Efficient techniques for adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
6. P. Brusilovsky and M. T. Maybury. From adaptive hypermedia to the adaptive web. *Communications ACM*, 45(5):30–33, 2002.
7. P. Brusilovsky, E. Schwarz, and G. Weber. ELM-ART: An intelligent tutoring system on the World Wide Web. In *3rd Int. Conf. on Intelligent Tutoring Systems, LNCS 1086*, pages 261–269. Springer, 1996.
8. R. H. C. Carver and W. Lane. Enhancing student learning through hypermedia courseware and incorporation of student learning styles. *IEEE Transactions on Education*, 42(1):33–38, 1999.
9. A. Davidovic and E. Trichina. Open learning environment and instruction system (OLEIS). *SIGCSE Bulletin*, 30(3):69–73, 1998.
10. T. Eymann, M. Reinickke, O. Ardaiz, P. Artigas, F. Freitag, and L. Navarro. Self-organizing resource allocation for autonomic network. In *Database and Expert Systems Applications, 2003*, pages 656–660, 2003.
11. S. Ferrandino, A. Negro, and V. Scarano. CHEOPS: Adaptive hypermedia on World Wide Web. In *IDMS'97, LNCS 1309*, pages 210–219. Springer, 1997.
12. J. Greer, G. McCalla, J. Vassileva, R. Deters, S. Bull, and L. Kettel. Lessons learned in deploying a multi-agent learning support system: The I-Help experience. *Artificial Intelligence in Education*, pages 410–421, 2001.
13. N. López, M. Núñez, I. Rodríguez, and F. Rubio. Including malicious agents into a collaborative learning environment. In *6th Int. Conf. on Intelligent Tutoring Systems, LNCS 2363*, pages 51–60. Springer, 2002.
14. N. López, M. Núñez, I. Rodríguez, and F. Rubio. WHAT: Web-based Haskell adaptive tutor. In *AIMSA'02, LNAI 2443*, pages 71–80. Springer, 2002.
15. N. López, M. Núñez, I. Rodríguez, and F. Rubio. Encouraging knowledge exchange in discussion forums by market-oriented mechanisms. In *ACM Symposium on Applied computing (SAC'04)*, pages 952–956. ACM Press, 2004.

16. G. McCalla. The search for adaptability, flexibility, and individualization: Approaches to curriculum in intelligent tutoring systems. In M. Jones and P. Winne, editors, *Foundations and Frontiers of Adaptive Learning Environments*, pages 91–122. Springer, 1992.
17. M. Miller and K. Drexler. Markets and computation: Agoric open systems. Available at: `http://www.agorics.com/agoricpapers.html`, 2000.
18. L. Robbins. *An Essay on the Nature and Significance of Economic Science*. Macmillan, 1984, originally published in 1932.
19. M. Stonebraker, P. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu. MARIPOSA: a wide-area distributed database system. *The VLDB Journal*, 5:48–63, 1996.
20. C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and S. Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18:103–117, 1992.
21. G. Weber. Adaptive learning systems in the World Wide Web. In *7th Int. Conf. on User Modelling, UM'99*, pages 371–378. Springer, 1999.
22. B. Woolf, J. Beck, C. Elliot, and M. Stern. Growth and maturity of intelligent tutoring systems: A status report. In K. Forbus and P. Feltovich, editors, *Smart Machines in Education*, chapter 4. AAAI Press/The MIT Press, 2001.
23. H. Wu, P. De Bra, A. Aerts, and G. Houben. Adaptation control in adaptive hypermedia systems. In *AH 2000, LNCS 1892*, pages 250–259. Springer, 2000.