Adapting Heuristics for Notification Systems

Brandon Berry
Center for HCI & Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg VA 24061-0106
brberry@vt.edu

ABSTRACT

Notification systems are growing in use from home to the workplace. They are used to monitor stock prices, navigate in a vehicle, and to encourage collaboration at a job, among countless other applications. These systems must be developed and tested, but existing usability evaluation methods are not always useful. They fail to consider systems used in divided-attention tasks and the impact on primary tasks.

Thus there is a need for usability evaluation methods that are adapted specifically for notification systems. Being able to find more usability problems with the adapted methods than the existing methods will aid in not only evaluations, but iterative design as well. Since there has already been much work on developing usability evaluation methods, those can be used as a basis for creating methods to work with notification systems.

Since heuristic evaluation is a popular method with low cost, it is a good choice for adaptation. So a new set of heuristics for notification systems is created and tested against a standard set of heuristics. Applying these adapted heuristics against a standard set resulted in the adapted method performing slightly better than the standard method. They showed promise in identifying valid usability problems along with applying directly to notification systems.

Categories and Subject Descriptors

D.2.2 [**Design Tools and Techniques**]: User interfaces

Keywords

usability evaluation, heuristics, notification systems

1. USABILITY EVALUATIONS

The main focus of this project is studying and adapting usability evaluation methods for notification systems. Some terminology is thus needed to understand this project. Usability is [6] "the extent to which a computer system enables users, in a given context of use, to achieve specified goals effectively and efficiently while promoting feelings of satisfaction." It is measuring the ability of an interface to allow the user to achieve their goals as easily as desired.

A usability evaluation method is defined [5] as "any method or technique used to perform formative usability evaluation (i.e., usability evaluation or testing to improve usability) of an interaction design at any stage of its development." Note that a formative evaluation is an evaluation performed during the design process and is meant to inform the designers of what they should do with their system to improve usability. This is opposed to a summative evaluation, performed after a system is considered complete to inform the designers about what they have accomplished.

The output of a usability evaluation method is a list of usability problems. That is common to all methods. But also, some usability evaluation methods [5] "have additional functionality, such as the ability to help write usability problem reports, to classify usability problems by type, to map problems to causative features in the design, or to offer redesign suggestions." An evaluator is any person who uses a usability evaluation method to test an interface. Participants are typically recruited for this role.

1.1 Heuristics

The usability method studied in this project is heuristic evaluation. This is defined [15] as "a usability inspection method whereby a set of evaluators produces lists of usability problems in a user interface by going through it and noting deviations from accepted usability principles." So the evaluators are given a list of guidelines an interface should follow, and the evaluators then decide how close the interface is to meeting those guidelines. In the process of doing that, the evaluator can list the specific interface elements that led to their decisions. The result is a list of usability problems that if fixed will lead to an interface that meets all of the guidelines (heuristics).

Since heuristic evaluation is described [14] as the "most general of the usability inspection methods and is also the easiest to learn and apply" it was chosen for this project. A main feature [1] is that it helps prevent the user from wasting their time trying to find trivial problems and instead focuses them on the most important usability issues.

2. NOTIFICATION SYSTEMS

To study notification systems, we first must define [9] and understand them. Notification systems are used in any divided-attention, multitasking situation. Their purpose is to "provide reaction to and comprehension of valued infor-

ACM-SE Conference 2003 March 7-8, 2003, Savannah, GA.

mation in an efficient and effective manner without introducing unwanted interruption to a primary task." The key idea to remember is that they are not "used in extended periods of concentration in an orderly, predictable task-action flow." This is what separates notification systems from other interfaces. Most interfaces intend for the user to concentrate on using them and they do not worry about distractions or interruptions.

Some examples of notification systems include new email notification and car navigation systems. For email, the user's primary task is not (hopefully) to wait on new email. Instead, they are performing other duties and wish to be notified when a new message arrives.

This is similar to car navigation. A driver's main goal should be to drive safely and legally. If the driver only cares about reaching their destination, then other motorists are in danger. Thus sound notifications are commonly used to inform the driver of when to turn and onto what streets. Auditory delivery of the notifications allows the driver to retain focus on driving while still informing them of navigational decisions.

A computer does not even have to be visible for a notification system to work. In an office lobby, a waterfall display can be set up. The rate of the water falling is then linked to the amount of network traffic that is occurring to the company's website. Or there could be a fake flower on a person's desk. Whenever the flower blooms, that means a meeting is about to start.

As can be seen, notification systems can also be decorative while still providing useful information. With the waterfall, the water stopping would be an immediate sign there is some problem with the network.

The examples indicate there is a wide range of notification systems just as there is a wide range of common desktop programs. Now think about having to evaluate those common programs. Would it be accurate to evaluate document creation software in the same way as a web browser? Similarities in evaluations would be present, but there would have to be some differences to account for the different tasks being performed. Therefore, all notification systems should not be evaluated in the same manner.

2.1 Classification

To evaluate notification systems differently, they must be split into different types. Now there is the problem of grouping notification systems into categories. For that purpose, McCrickard and others developed a classification scheme [3]. They began by deciding on three critical parameters that can be used to describe a notification system.

First, there is interruption. This is reallocation of attention from a primary task to a notification. The intent is to measure how much a user's attention becomes diverted away from their primary task and onto whatever notification has occurred.

Next, there is reaction. Reaction is the immediate response resulting from a stimuli classification. Measured here is what actions a user takes when presented with a notification. The user could ignore the notification or take many steps to interact with it.

Last, there is comprehension. This measures how information is made sense of, related to existing knowledge, and stored in long term memory. In other words, what the user learned from the notification.

One way to think of these three parameters is as steps towards handling a notification. A user must be somehow interrupted to notice the notification. Then the user has some reaction to the notification, which could even be choosing to ignore the notification. Notice that just because a user becomes aware of a notification does not mean there is any meaningful reaction involved. After the reaction, the user is now able to integrate the knowledge gained from the notification into their thoughts.

The creation of these parameters did not occur in a vacuum. In one study [7] of designing and testing a notification system, the authors considered level of interruption, requirement for response, and content overhead. Those closely relate to interruption, reaction, and comprehension. The authors rated each parameter using one to three checkmarks. But the McCrickard classification scheme attempts to use a more exact rating.

A scale from 0 to 1 is used to assign a value to each parameter, with 0 being low and 1 being high. Values in-between 0 and 1 are also used, but assigning them is still a subjective process. For example, the difference between 0.6 and 0.7 is not clearly defined yet. As of now, using values of 0, 0.25, 0.5, 0.75, and 1 are sufficient for classifying a system.

Using values from 0 to 1 with axes for I (interruption), R (reaction), and C (comprehension), a three-dimensional unit cube can be built. The cube now contains all possible notification systems. Each corner of the cube is the basis of one type of notification system. The authors [5] describe eight types of notification systems with their associated interruption-reaction-comprehension (IRC) values and provide an example scenario for each.

The eight categories are: ambient media, indicator, secondary display, noise, diversion, alarm, information exhibit, and critical activity monitor. All notification systems should fit into at least one of those categories. Using values in-between 0 and 1 for the IRC rating results in systems that are hybrids of multiple categories.

The categorization of notification systems allows for a standard method of describing such systems. Research on one category can then be extended to multiple researchers, as they have a common baseline to build upon. Design guidelines can be developed for a category so that designers can maximize usability for their product, as opposed to using more general and less specific guidelines.

Similarly, usability evaluation methods can be adapted to each category to maximize the usefulness of a method. At a notification systems seminar, seven out of eight participants stated that different categories should be evaluated differently [10]. Part of the goal of this project is to discover if that is the correct approach. The hope is that by doing so, suggestions can be developed as to how to evaluate a specific category of notification systems.

3. MOTIVATION

Desktop programs are the main concern of many usability evaluation methods. This is understandable since the focus of usability testing began on interfaces used on personal computers. Indeed, that is the overwhelming method in which people interact with computer-based systems. But many notification systems do not run on the desktop, such as the car navigation system. Ubiquitous computing interfaces, in which the computer systems become integrated into everyday life, also do not run on the desktop. So as notification systems grow in use, the current usability evaluation methods may not fit. The methods used to test command line interface, for example, would not then directly apply to testing a graphical user interface, if only because a graphical user interface introduces some features that evaluators, before the existence of the new graphical interface, would not have thought of having to test. Therefore the evaluation methods applied to primary task interfaces must evolve to be able to fully evaluate divided-attention, multitasking interfaces.

As discussed by Chewar and McCrickard [3], traditional usability evaluation methods do not take into account the impact of the notification system on primary tasks. Studies have been done, such as on Scope [17], that only evaluated the program without any other task. Having a tester concentrate on using the system, and trying to find usability problems that way, fails to consider the usability impact of other tasks. Therefore the evaluation methods must be adapted to work with divided-attention tasks.

To that end, the authors developed a "realistic usage environment" that had the participants perform primary tasks while simultaneously having Scope run on the desktop. A questionnaire was then used to see how well the user's perception of the system matched to the design model claims from the design team. A claim is defined [9] as "an expression of tradeoffs about an artifact in reference to its usage scenario." A claim can be pro or con. For example, one claim is "When new mail arrives, the Scope interface displays a blinking dot which: (+) enables quick reaction from the user to new info, but (–) may cause unwanted interruption to the primary task." Participants were asked to rank their agreement that the interface supports those claims. This way, system designers can discover if their intended interface is actually what people are perceiving. It also has the positive of specifically identifying usability problems with certain features.

After running the experiments, the authors concluded that the evaluation method showed promise in identifying usability problems. Plus, in relation to evaluating different categories of notification systems differently, as discussed before, the questionnaire used claims that applied to the category the features fit into and were not application specific.

Citing the problem that existing usability evaluation methods only focus on primary tasks, one study [2], by Greenberg and others, developed heuristics to evaluate team tasks with a groupware system. They also adapted Nielsen's heuristics and found the performance of their adapted heuristics on a groupware interface compared favorably to the performance of Nielsen's heuristics on primary task-based interfaces. So the approach of adapting heuristics to fit certain system domains has been successful before.

4. NIELSEN'S HEURISTICS

Originally published in 1990 [11], Nielsen's heuristics have evolved to a set with [13] "maximum explanatory power." He developed these heuristics by studying hundreds of usability problems and trying to categorize them. Most usability problems should fit into one of the heuristics. These heuristics have not been altered since 1994, so he believes they are the most useful heuristics he has developed. There are 10 heuristics and they are listed in Table 1.

Table 1: Nielsen's Heuristics

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Nielsen discusses that category-specific heuristics can be developed as supplemental to the existing heuristics. This approach has been tried with ambient media [8] and has resulted in some success.

5. ADAPTED HEURISTICS

Mankoff and others [8] took the approach of starting with Nielsen's heuristics, deleting non-applicable heuristics, rewording some heuristics to apply to ambient media, and adding some heuristics of their own.

This project took a different route. Here, the major usability problems that can be associated with notification systems were thought of collectively. Then those problems were categorized into 8 major heuristics. Note that one heuristic encompasses three guidelines about customization of IRC values for a system. Just as most usability problems of a standard interactive systems should fit into one of Nielsen's heuristics, most usability problems with notification systems should fit into one of these new heuristics. Here are the new heuristics with an accompanying explanation:

5.1 Notifications should be timely

A notification must arrive in time to be useful. If data is being monitored, the notification should report the most current data as soon as the user desires it. A sports score or stock data that arrives hours too late is worthless. A warning that a critical system is down for a machine that arrives even seconds late can be life-threatening. Also, notifications can arrive too often, as perhaps a user only wants a data update every hour.

5.2 Notifications should be reliable

All of the information in the notification should be identical to that of the underlying system. Incorrect info can lead to undesirable reactions. False alarms become very annoying, ranging from fire alarms to critical status info for a system, like a nuclear reactor. This leads to mistrust of all future notifications. Also, any notifications that should occur but never do are incorrect notifications.

5.3 Notification displays should be consistent (within priority levels)

The method of displaying a notification should be consistent for a level of notification. For example, all high priority notifications should be consistent so the user knows the priority when they notice the notification. The only reason to change the display method is if the intent is to alter the level of interruption, reaction, or comprehension.

5.4 Information should be clearly understandable by the user

Language and display method should target the users. When a notification occurs, a user should be able to understand at least the basic information the notification is trying to convey. This allows the user to decide quickly how to respond, where one response may be that the notification deserves further study.

5.5 Allow for shortcuts to more information

When appropriate, a notification should provide a gateway to a related information system. For example, a news headline can link to a full news story. A new email notice can link to the actual mail. When this is not possible, such as a notification system involving a waterfall that indicates amount of rain, the user should be able to know enough about the system so that they can find detailed information related to the notification. In this case, there would be a method of determining the exact weather outside (besides a window).

5.6 Indicate status of notification system

The system should be aware of the rate it is able to provide information. If new info is only available every hour, the user should be aware of that. If a system is unable to provide notifications, it should inform the user of that status. This way the user does not expect notifications that will never occur. For example, if an email server can not be reached, a new email notification system should notify the user of that problem.

5.7 Provide context of notifications

If there is a cycle, or progression, of information, such as sports scores, there should be a way of indicating context. This allows the user to plan their allocation of attention, get a sense of the cycle length, and expect status changes. That means the user should be able to see what scores are upcoming, and thus are able to plan when to pay attention to the notifications so that they see the info they desire.

5.8 Allow adjustment of notification parameters to fit user goals

5.8.1 Interruption

The way a notification is presented can be altered. For example, the user can control the amount of viewing area a notification takes up. Or the levels of sound or physical movement can be changed to become more or less interruptive.

5.8.2 Reaction

The actions required by a notification can be altered so that they can range from virtually ignoring the notification to having to take multiple steps to respond to the notification.

5.8.3 Comprehension

The amount of information conveyed can be adjusted to provide varying levels of detail. The system can also adjust the types of info it displays to meet user goals. For example, a user can request to only be notified about one sports team, instead of all sports scores.

6. INTERFACES

Brainstorm[18] is the notification system evaluated in this project. It is a groupware product that allows users to post ideas and replies to those ideas. Three separate client interfaces were developed, each meant to have different design-model IRC ratings.

Using different types of notification systems for evaluation accounts for trying to develop usability methods to not only apply to notification systems in general, but to also maximize their effectiveness with a certain type of notification system.

- Interface A has an IRC rating of 0.5/0.5/0.5 and this places it in the center of the IRC cube. So it does not belong to any one category of notification systems. The designers considered this a baseline system.
- Interface B has an IRC rating of 0.25/0.25/0.5, meaning it belongs in the lower half of the cube and is an equal mixture of noise, indicator, ambient media, and secondary display. This system is meant to have low interruption and reaction, yet still allow for meaningful comprehension.
- Interface C has an IRC rating of 0.75/0.75/0.75, placing it as a critical activity monitor. Therefore this is meant to be very interruptive and prompt high reactions while providing a lot of comprehension.

7. EVALUATION

Participants for the evaluation were recruited at Virginia Tech. Each participant was shown each interface and given a demonstration by the designers. The participants were also allowed to ask questions about the interface to clarify their understanding. They were then asked to fill out an electronic form that listed one set of heuristics and the explanation for each heuristic.

The form asks the evaluator to rate, on a 5-point scale, their agreement that the interface follows the heuristic. Their choices were from {strongly disagree, disagree, neutral, agree, strongly agree} which maps to the integer range [-2,2]. They were then asked to comment on why they felt that way by citing elements of the interface. From these comments, a list of usability problems is built for each interface. Participants were also allowed to note that a heuristic did not apply to the interface.

The order in which the three interfaces were shown was varied to account for a learning curve with the evaluations. One person sees all interfaces, but only one set of heuristics. Therefore no participant is aware there is a different set also

Table 2: Usability problems found by heuristics

Interface	Nielsen	Adapted	Difference
A	14	13	-1
B	11	18	7
C	13	16	3

Table 3: Heuristic agreement scores

Interface	Nielsen	Adapted	Difference
A	-2	-6.5	-4.5
B	1.33	5.5	4.17
C	-1	3.25	4.25

being used. This is required, as explained by Nielsen [12], or else an evaluator would have the additional knowledge of another set of heuristics while attempting to apply the set in front of them.

8. ANALYSIS

Ignoring the specific heuristics being used, comments on the forms the participants filled out were used to create the list of usability problems found during the evaluation. Then for both set of heuristics, a list of usability problems found by each set was created. By counting the number of usability problems a set found, we get a number to associate with that set. This process was repeated for all three interfaces.

The technique of simply counting usability problems was used by Doubleday and others in their study [4]. It is important to note that their results showed that while experts founds more problems than the recruited users, the users found problems the experts did not. This indicates user-testing is crucial for testing usability evaluation methods, which is exactly what has been done for this study on adapted heuristics.

With the 5-point agreement scale discussed earlier, a mean value of agreement can be calculated for each interface on both set of heuristics. Using this, a comparison of whether an interface followed one set of heuristics closer than the other can be determined.

Examining the number of times the users thought that heuristic was not applicable gives a sense of the usefulness of the heuristics. A large number of comments that claim a heuristic does not apply indicates the set should undergo some revision. Otherwise evaluation time is being wasted and results will suffer.

9. RESULTS

9.1 Number of usability problems found

By counting the total number of usability problems each heuristics set found for each interface, the difference between going from Nielsen's set to the adapted set can be calculated. These results are summarized in Table 2.

The adapted heuristics found 9 more usability problems than Nielsen's heuristics. But with 90% confidence, the mean difference between the number of usability problems found is in the interval (-2.33, 8.33). Since 0 is included in that interval, there is no statistically significant difference between the two set of heuristics.

9.2 Not applicable heuristics

For the evaluators using Nielsen's heuristics, all but one thought that some heuristics did not apply to their evaluation of the interface. This is in stark contrast to the evaluators using the adapted heuristics, as there was not a single evaluator who thought that even one heuristic did not apply.

9.3 Agreement scale

Evaluators rated their agreement that an interface followed a heuristic on a 5-point scale. Summing the answers for each evaluation results in a number indicating how well the interface followed the usability principles of the heuristics overall. For example, a "perfect" score here would be 20, which would mean the evaluator strongly agreed the interface followed every heuristic.

Using the sums for each evaluator, the mean score for each interface with each heuristic set can be calculated. Then the difference between the scores from both heuristic sets can be used to see if the interface followed one heuristic sets more than the other. Table 3 summarizes the mean scores.

10. CONCLUSIONS AND FUTURE WORK

10.1 Adapted heuristics

Since there is no significant difference in performance between both sets of heuristics, this means that the adapted heuristics performed no worse than Nielsen's heuristics. And since the adapted heuristics found 9 more usability problems and the confidence interval is centered at 3, it appears the adapted heuristics performed slightly better. Because the interfaces evaluated were relatively simple and the number of evaluators smaller than desired, a statistically significant difference would not be expected.

It is interesting that no evaluators thought any of the adapted heuristics did not apply. The goal of this project was to create heuristics that applied directly to notification systems. It appears that has succeeded. This especially is meaningful because there were frequent instances of evaluators claiming that one of Nielsen's heuristics did not apply.

The agreement scores showed that interfaces B and C followed the adapted heuristics more than Nielsen's heuristics. Interface A had the opposite occur. Recall that interface A is the one with an IRC rating of 0.5/0.5/0.5, placing it in the center of the IRC cube.

Perhaps interfaces that do not fit well into any classification category do not perform well as notification systems. That is something to consider in future evaluations.

For the other two systems, one a mixture of noise, indicator, ambient media, and secondary display and the other a critical activity monitor, the evaluators tended to agree more than disagree that those interfaces followed the notification system heuristics. So the interfaces that are clearly notification systems followed the adapted heuristics despite no knowledge by the designers about these new heuristics.

10.2 Future Considerations

Time constraints limited the testing phase, so future evaluations will involve more participants and a wider range of interfaces. Along with that, the interfaces used must be studied in detail as it will help for the researchers to be very familiar with the usability problems before analyzing the problems found by the testing participants.

For comparing usability evaluation methods, using measures of validity and thoroughness along with weighted usability problems as discussed by other researchers [5, 16] may be useful. A more rigorous approach to comparing the effectiveness of the evaluation methods will result in comparisons applicable across multiple studies.

After further testing, each heuristic from the adapted set must be studied to learn which heuristics are finding the most usability problems and which are finding very few problems. This information can then be used to design a new set of heuristics for finding the most usability problems with the fewest number of heuristics. Finding the most efficient heuristics will result in the most useful set. This may involve a combination of heuristics from Nielsen and the adapted set, an approach taken in a similar study [8].

From the results, the adapted heuristics show great promise for effectively evaluating notification systems. Their performance was definitely no worse than Nielsen's heuristics, and his are considered a standard. With some more testing and adjustment, notification system designers may use these heuristics as their standard.

11. ACKNOWLEDGEMENTS

Thanks to Dr. McCrickard for creating the Notification Systems Seminar and to Christa Chewar for helping to run it. Also thanks to all of the seminar participants for providing insightful discussion and especially to Shahtab Wahid, Steve Battjer, and Aditi Jain for building the systems this study evaluated.

12. REFERENCES

[1] C. D. Allen, D. Ballman, V. Begg, H. H. Miller-Jacobs, J. Nielsen, J. Spool, and M. Muller. User involvement in the design process: why, when and how? In *Proceedings of the conference on Human factors in computing systems*, pages 251–254. Addison-Wesley Longman Publishing Co., Inc., 1993.

[2] K. Baker, S. Greenberg, and C. Gutwin. Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 96–105. ACM Press, 2002.

[3] C. M. Chewar and D. S. McCrickard. Workshop position paper: Adapting uems for notification systems. In *UbiComp 2002*. Available at: http://csgrad.cs.vt.edu/ cchewar/paper7.pdf.

[4] A. Doubleday, M. Ryan, M. Springett, and A. Sutcliffe. A comparison of usability techniques for evaluating design. In *Proceedings of the conference on Designing interactive systems : processes, practices, methods, and techniques*, pages 101–110. ACM Press, 1997.

[5] H. R. Hartson, T. S. Andre, and R. C. Williges. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 13:373–410, 2001.

[6] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516, 2001.

[7] S. Lock, J. Allanson, and P. Phillips. User-driven design of a tangible awareness landscape. In *Conference proceedings on Designing interactive systems : processes, practices, methods, and techniques*, pages 434–440. ACM Press, 2000.

[8] J. Mankoff, A. Dey, G. Hsieh, J. Kientz, S. Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *Procedings of the ACM Conference on Human Factors in Computing Systems (CHI '03)*, 2003.

[9] D. S. McCrickard and C. M. Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM 46(3)*, March 2003.

[10] S. Members. Pe 16: Evaluation of notification systems, Nov 2002. http://research.cs.vt.edu/ns/seminar /week13/PE16/evaluation.html.

[11] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33:338–348, 1990.

[12] J. Nielsen. Enhancing the explanatory power of usability heuristics. In *Conference proceedings on Human factors in computing systems : "celebrating interdependence"*, pages 152–158. ACM Press, 1994.

[13] J. Nielsen. Ten usability heuristics, Dec 2002. http://www.useit.com/papers/ heuristic/heuristic_list.html.

[14] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Conference proceedings on Empowering people : Human factors in computing system: special issue of the SIGCHI Bulletin*, pages 249–256. ACM Press, 1990.

[15] J. Nielsen and V. L. Phillips. Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared. In *INTERCHI*, pages 214–221, April 1993.

[16] A. Sears. Heuristic walkthroughs: Finding the problems without the noise. *International Journal of Human-Computer Interaction*, 9:213–223, 1997.

[17] M. van Dantzich, D. Robbins, E. Horvitz, and M. Czerwinski. Scope: Providing awareness of multiple notifications at a glance. *In Procedings of the Symposium on Advanced Visual Interfaces (AVI '02)*, 2002.

[18] S. Wahid, S. Battjer, and A. Jain. Developing ideas using personal & large screen displays. In *Procedings of the ACM-Southeast Conference*, 2003.