Hindawi Publishing Corporation
Advances in Human-Computer Interaction
Volume 2008, Article ID 597629, 9 pages
doi:10.1155/2008/597629

Research Article

Child-Centered Evaluation: Broadening the Child/Designer Dyad

Sofia Pardo, Steve Howard, and Frank Vetere

Interaction Design Group, Department of Information Systems, The University of Melbourne, Level 4 111 Barry Street, Carlton, Melbourne 3010, Australia

Correspondence should be addressed to Sofia Pardo, miriamp@pgrad.unimelb.edu.au

Received 1 October 2007; Accepted 6 July 2008

Recommended by Adrian Cheok

Some settings challenge a literal interpretation of user-centered design orthodoxy; that design is best done *for* a user, by designing *with* that user. We explore the value that a copresent proxy and interpreter can bring to certain hard-to-reach or difficult-to-interpret situations; in this case the evaluation of educational software intended to be used by children. We discuss the effect that introducing a teacher had on the results of the evaluation and conclude that adding an expert-based component to evaluations increased its diagnostic power.

1. Introduction

Certain hard-to-reach places make adopting a literal interpretation of user-centred design difficult. Designing with users can be challenging if those users have limited communication skills, restricted cognitive abilities [1], or if there are large power differentials present between designer and user. Here, we are concerned with bridging such distances, and our case is the evaluation of educational software intended for use by children.

We present two evaluations, firstly of current practice and secondly of a novel method called Kids and Teacher Integrated Evaluation (KaTIE), and we focus on the role and the effects of the teacher as an adjunct to the traditional designer/child dyad. KaTIE combines elements of both expert and user-based evaluation, and strongly embeds the teacher in the process, aiming to account for both the usability and pedagogical aspects of educational software in tandem.

The next section discusses child-centred design, and surfaces the challenges implied. Then, we describe and evaluate a widely adopted current approach to child-centred evaluation. Findings from this evaluation influenced the development of KaTIE, which we next describe in detail, and report the results of its evaluation. We conclude with a discussion of hybrid evaluation methods, and the effect that the teacher's presence had on the evaluation outputs.

2. Background

2.1. Child-centered design

A central tenet of child-centered design is the primacy of the child-designer conversation, wherein the voice of the child is heard first hand by the designer, without the mediating influence of teachers or parents [2, 3]. This view is however not without its difficulties. Some challenges implied by the child-designer dyad relate to the child's developing but still immature communication skills [4]. Other issues include potential power differentials between the designer and the child that may hamper the gathering of the child's genuine views and opinions [2, 5]. As a user group, children have particular characteristics that need to be taken into consideration as participants in technology evaluation. These considerations have driven the creation of methods specifically tailored for the inclusion of children [6].

Children have been involved as testers in the evaluation of technological products that serve different purposes, such as play, learning, entertainment, or functionality (e.g., enabling products such as word processors, searching engines) [7]. The input collected from children in the evaluation of these technologies has been considered as mostly regarding usability and the level of engagement or fun involved when interacting with the product [8]. Accounting for usability and fun is often sufficient for products whose goal is to

entertain or facilitate particular tasks, but are of incomplete value for those products that aim to engage children in learning.

2.2. Challenges in the context of educational software

The evaluation of educational software places additional challenges to the child-designer dyad since a major aim of such evaluations is to gather feedback regarding children's understanding of the concepts/ideas conveyed in the software, as well as determining how the learning goals anticipated by the designer are being met. Learning is not an overt time-independent behavior that can be easily observed and easily measured [9]; gathering evidence of learning often requires the evaluator to make judgments of what is being observed or expressed, and to what learning unfolds over time. In this sense, learning is in sharp contrast with current approaches to the evaluation of usability or fun, where error rates and facial expressions [10] might be taken as indicators of these attributes, respectively.

In the context of educational software, the verbal exchange between designers and children is often the main communication channel by which designer can gauge whether children are gaining new understanding or refining their existing prior knowledge. In this sense, being able to engage with the children in a dialog that uncovers children's understanding is a major concern. In addition to the children's developing verbal skills, the new vocabulary associated with a particular knowledge domain can be problematic. Moreover, children are often unaware of the learning goals of the software and quite often these are not their own [7]. This lack of knowledge becomes a hurdle when trying to determine how well the software supports the children in attaining those goals, as they cannot provide feedback regarding the goals they have not attained yet [8]. Thus communication, domain, and pedagogical literacy each challenge the place of the child in child-centered evaluation.

3. Related work

The challenges associated with the dyad child-designer in the context of educational software have been addressed in different ways. However, the solutions given have mostly resorted to the employment of different evaluation methods and techniques that in conjunction with direct observation and dialog with the children may provide a more complete account of the educational effectiveness of the software. Pre- and postwritten tests have been used to determine any learning outcomes [11, 12]. The conduct of heuristic evaluations [13, 14] is also a common practice, as experts are able to provide informed views on the educational soundness of the instructional design implemented. Alternatively, the self-administration of questionnaires has been used to determine the pedagogical usability of educational software [15].

These methods are of peripheral nature to child-centered design practices, as they provide complementary views to those of children and designers. In this sense, results yielded by pre- and post-test, expert reviews or self-administered questionnaires need to be amalgamated in such a way that a single body of feedback is produced. This amalgamation process can be challenging, it has been found that children's views are not necessarily in synchrony with good educational practice [8]. What children may consider amusing and entertaining at the interface may hinder their engagement with a particular learning task.

The use of pre- and postwritten tests does not provide feedback regarding the process by which a correct or incorrect answer is given. Obtaining the expected answer on a written test is of little value for design purposes if account of how the children got to that answer is not provided. The predictions established through heuristic evaluations carried out by experts cannot anticipate the learning process that will take place when children interact with the software [9]. Therefore, determining the educational effectiveness of the software using this evaluation method has serious limitations. Lastly, self-administered questionnaires also share the limitations of pre- and postwritten tests as they mostly rely on children's perceptions of what they think they learned and little attention is given to the behaviors that took place during the interaction [16].

These limitations lead to the consideration of alternative evaluation practices that would preserve the benefits of directly involving children, include the views of experts, the designer's insights, and design agenda. As a consequence, these alternative practices are drawn from the child-based and expert-based evaluation methods [17] in order to overcome the limitations that the child-designer dyad poses in the context of educational software.

4. Understanding current practice

In many examples of current practice, the child-designer conversation is broadened by the inclusion of a third stakeholder, an "expert." The expert may bring literacy in technical issues, such as usability evaluator [16] pedagogical insights, such as a teacher or educational expert [18], domain knowledge such as a scientist if the software is designed to teach science concepts [9] or communication, or translation skills to the team [19].

Approaches that combine elements of child-based and expert-based evaluation methods can host different evaluation practices as different arrangements involving children, designers, and experts can take place. In this paper, we focus on arrangements wherein experts, more specifically teachers, are involved in order to determine the learning path followed by the children as they use the software and the attainment of the anticipated learning goals. This focus however is not indifferent to the overlapping nature of usability and learning when children interact with educational software. We adhere to the compartmentalized view of others [11] who consider that these two dimensions although interrelated can be focally evaluated.

Teacher involvement in evaluation, but also in other parts of the design process, has been seen as undesirable as the existing power relationship between teachers and students could lead to a situation wherein the children

might feel tested or compelled to perform well [2]. This concern has kept teachers on the periphery of child-based evaluation. However, where computing is concerned these power differentials can be inverted as children, increasingly "digital natives," are often more technically literate than their "digital immigrant" teachers [20]. In this sense, the involvement of teachers does not seem to strengthen the existing power differentials between adults and children. Despite this concern, the following two evaluation methods involve teachers in child-based evaluations in two different ways. The subsequent findings present the influence teachers' participation had on the outcomes produced by the evaluations.

4.1. In-school evaluation (ISE)

ISE is an evaluation method created by one of the biggest educational software producer in Australia. This method has been used extensively across different educational contexts, and has involved over 300 schools in Australia and New Zealand. ISE emerged from the need to reach large numbers of children from different socioeconomic backgrounds in order to create software that is suitable for different types of learners in different contexts.

In ISE, both teachers and designers work in a parallel fashion with a pair of students each, often in the same room but apart from each other to minimize disruption. Teacher and designer use the same data collection protocol and this is provided by the designer as a form to be completed in real-time during the evaluation. The form guides the evaluation process with prescribed questions to ask the children and particular things to observe and record. This form contains the questions the designer has considered relevant in determining the educational effectiveness of the software. The evaluation form is often given to the teacher a week in advance along with the nearly completed software prototype so that familiarization can occur ahead of time.

The rationale behind ISE is partly economic, and partly about process standardization. Having the designer and the teacher play the same role halves the time needed for data collection, and helps facilitate comparisons across different child/school settings by standardizing the process.

4.2. Evaluating ISE

The ISE method was considered a suitable case study given its uniqueness in combining the Child-Based and Expert-Based evaluation methods. Similar hybrid evaluation practices had not been found to our knowledge in Child-Centred Design literature; much less an account of the type of outcomes produced by such practices.

4.2.1. Data collection

The data collection consisted of collection of artefacts, that is, the evaluation forms containing the hand-written notes of the evaluators (teacher and designer), video footage of the evaluation process and interviews with all participants (teacher, designer, and children). As this paper focuses on the feedback collected, the findings here presented are concerned exclusively with the analysis carried out on the hand-written notes.

The case study involved the participation of one designer from the software company, who was well experienced in the ISE method, two primary school teachers (from two different schools) who had used ISE a few times before, and a total of 16 children between 7–9 years old (8 children per school). The evaluations took place at two primary schools over a period of 2 hours each, thus each teacher performed two evaluations and the designer performed 4 evaluations in total (2 per school).

Four educational software prototypes were evaluated with the ISE method. Each prototype was evaluated twice, by the teacher and the designer in a simultaneous fashion. Two prototypes aimed to teach children about the relationship between sample space and likelihood of outcomes, the third one taught about classification of substances according to their behavior in water, and the fourth taught about energy chains.

4.2.2. Data analysis

The analysis of the feedback collected during the ISE evaluations was typed in into tables for coding. The coding scheme employed to analyze the feedback emerged from the data and it was refined through several iterations. As the ISE evaluation form contained prescribed questions, the feedback was broken down into instances that followed the same structure of the questions. In this way an instance could correspond to a prescribed question and its associated hand-written note, or to any notes that were not necessary linked to a question (e.g., notes written on the margin of the page).

The coding scheme consisted of four-high-level codes, which were broken down into subcodes for a total of 13 codes. The feedback was divided into reporting, descriptive, diagnostic, and advisory types. A single feedback instance could be coded as belonging to one of these types or a combination of them; hence they were further coded as single or combined. Feedback coded as reporting regarded to those wherein the evaluator was reporting the verbal response of the children to the questions asked. Within the reporting feedback, a distinction was made between those instances where the verbal answer/question referred to the content conveyed by the software, to the interface or to improvements children thought could be made to the software.

Descriptive feedback included observations made by the evaluators of what was taking place as the children interacted with the software. This feedback was broken down into those instances where the description referred to the interface/navigation, to technical problems, to the evaluator behavior, and to the learner behavior. The latter was further classified into those instances conveying learners' emotional responses, interaction between children and approach to learning task. Diagnostic feedback included those feedback instances wherein the evaluator made a judgment. This type of feedback was either diagnosing the interface/navigation or the learner's understanding or behaviors. Finally, advisory feedback included instances where the evaluator made

suggestions to change or improve something in the software. These suggestions were classified into those addressing changes to the interface/navigation and those regarding the learning tasks or instructional design in order to better support children's learning.

The following is an example of a combined feedback instance collected through ISE method:

> Did the students enjoy the animations? Please record any comments. (Prescribed instruction) "*Is it finished*" [**Reporting**]
>
> *Would be good to see propeller working* [**Advisory**] *as these students don't know how a boat travel thru water* [**Diagnostic**]. *Would improve understanding* [**Diagnostic**].

As the evaluations were focused on determining the educational effectiveness of the prototypes, three expectations were anticipated regarding the amount and type of feedback collected. First, it was expected that the feedback coded as referring exclusively to the interface/navigation would be less in number. Second, it was expected that those feedback instances providing insights into the learner and the evaluator's behavior would have a significant percentage. Thirdly, as learning is not an overt behavior, it was expected that an increased number of diagnostic feedback would be found, either on its own or in conjunction with descriptive, reporting, and advisory feedback.

4.2.3. Findings

The results of the coding process are presented at two levels. First, differences and similarities between the teachers and the designer feedback are identified across all evaluations, and second overall tendencies on the type of feedback collected are also drawn. There were no major differences between the feedback collected by the teachers and the designer across the four-software prototypes evaluated. The amount of feedback collected by both was approximately the same, with teachers writing slightly more instances in three out of four evaluations. The amount of reporting, descriptive, and diagnostic feedback was similar, while advisory feedback was mostly reported by the designer.

The feedback collected by the teachers and the designer remained focused on the learning aspects of the interaction across all evaluations, as the number of single feedback instances regarding the interface/navigation of the software was less. Most reporting and descriptive feedback regarded learners' verbal and nonverbal behavior, respectively, hence providing a rich account of the children interaction with the learning tasks.

The similarities of the feedback collected by the teachers and the designer are perhaps related to the evaluation form and set up of the ISE method. The focus on learning aspects of the interaction can be associated to the overall flavor the evaluation form employed has, as most of the prescribed questions and issues to observe aim to explore children's understanding of the concepts introduced by the software. The highly structured evaluation form may have been beneficial in keeping both the teacher and the designer focused on the purpose of the evaluation, to determine the educational effectiveness of the software, and stop them from wandering into other aspects of the interaction, such as usability.

The lack of significant differences between the type of feedback collected by the teachers and designers' feedback can be a result of the prescriptive evaluation form, as this tended to homogenize the feedback collected. This standardization may have resulted in the reduction of the potential added value that involving teachers may have brought. Teachers' involvement in ISE method did not take advantage of their expertise as educator, as the setup resembled a form-filling exercise wherein the questions prescribed the flow and interaction between teacher and children. In this sense, the ISE method heavily relied on the evaluation form as opposed on the insights the teacher might have brought to the evaluation as an expert educator.

At an overall level, the type of feedback collected through the ISE method showed that a significant amount of the feedback instances was of descriptive and reporting nature with less number of diagnostic types. This tendency can also be the result of the structure of the evaluation form, as this one mostly encouraged the collection of these feedbacks. The diagnostic section on the form was located at the end and it consisted of a liker scale. The reduced amount of diagnostic feedback was not perceived as problematic by the designer (as commented in interview) as the main purpose was to be able to recreate what happened on the field (reporting and descriptive feedback) as opposed to emitting judgments. This is also in accordance with the nonexpert role played by the teachers during in the ISE method.

It can be assumed that the diagnosis regarding the educational effectiveness of the software with the ISE method is carried out outside the field based on what was reported and described in the evaluation form. This diagnostic practice takes as face value what is written down and requires the close examination of the hand-written notes to determine whether the prototype achieved what it was meant to. A detached examination of the feedback captured in the ISE evaluation form can be misleading as in some cases children's responses recorded on the form may require further interpretation as to: in which context and after how much and what sort of prompting were the children able to articulate a particular answer.

Our findings show that ISE method does not provide insight into educational effectiveness; rather this is determined by a third party based on what was described and reported on the evaluation forms. Moreover, teacher's involvement, although central to the evaluation, is not as a pedagogical expert, but of a form-filling aid. The collection of diagnostic feedback is fundamental in determining the educational effectiveness of the software, as it is in the field with all the necessary contextual clues that an informed and empirically based judgment can be attained. Although there is considerable value on reporting children's verbal responses/comments and describing their behaviors, these fall short in conveying a complete picture of the children's learning experience.

There are opportunities to provide an integrated commentary by merging child and expert-based evaluation methods in such a way that the teacher is allowed to play to his or her strengths. In doing so the teacher may be encouraged to provide diagnostic feedback regarding educational effectiveness. Encouraging the teacher and designer to share their insights and come to a collective view could facilitate this. Care would need to be taken so as not to overly constrain the teacher's commentary with design-centric questions. An open evaluation form could potentially encourage teachers to draw from their expertise and guide the dialog with the child.

5. Changing the place of the teacher

Although ISE is a hybrid child- and expert-based evaluation methods that includes a teacher, the centrality of the child-designer conversation is not diminished. Rather it is complemented by a child-teacher conversation, but one in which the teacher is playing the role of a designer. What might a teacher bring to the process if his or her pedagogical voice were strengthened? How might this be achieved?

KaTIE extends current practice, by facilitating a child-designer-teacher conversation, and aims to strengthen the account of the educational effectiveness of the software resulting from the evaluation. By including teachers as teachers, the evaluation's set up is changed radically. KaTIE is a collaborative [21] and lightweight [22] or discount method, wherein the inclusion of teachers, designers, and children is grounded in their respective areas of expertise in order to gather rapid insights into the educational effectiveness of software. KaTIE agrees with the view that the evaluation of the pedagogical design implemented in educational software remains an adults' matter [8] but also firmly believes that adults' judgment needs to be grounded not just on theoretical views but on direct observation and dialog with children.

The collaboration between the teacher and the designer in KaTIE takes place over a period of three consecutive hours distributed as follows.

(1) Preparation stage (1 hour). This stage aims to create rapport between teacher and designer since KaTIE is not based on a continuing relationship between them. There is a need to develop a shared understanding of the evaluation purpose as well as a sense of importance of their respective roles and perceptions. This stage also aims to familiarize the teacher with the software and the evaluation form that will be employed.

(2) Data collection stage (1 hour). The second stage involves observing and engaging in dialog with the children as they use the software, with a particular focus on children's understanding of the content conveyed by the software and the overall learning experience the software supports. In this stage the teacher leads the dialog with the children while the designer mostly observes in the background providing support if required or asking additional questions to the children. Both, teacher and designer also play the role of note takers.

(3) Reporting stage (1 hour). This last stage is of crucial importance in KaTIE as it consolidates the outcomes of the evaluation. This stage requires the teacher and the designer to engage in a conversation wherein their observations and written notes are shared with the purpose of creating a rich picture of the children's interaction with the software. This conversation is assisted with an open ended template. This stage also brings to the table the learning goals and objectives that have been anticipated by the designers and looks at them under the light of the teacher and designer observations. The reporting stage allows for the consideration of solutions that can address identified issues with the pedagogical design as well as more open ended brainstorming.

These three stages and the key tasks associated with them were summarized into three reference cards for the designers' guidance. These cards were given to the software designers along with two short documents containing the tenets on which the KaTIE method was based. Although KaTIE has its origins in ISE, it has distinctive characteristics. The number of children involved in KaTIE is less than the ISE method as a consequence of the three-hour duration of the evaluation. The teacher and designer focus their attention on the same pair of students, rather than a different pair each, in order to gather and merge their complementary views on the educational effectiveness of the software.

In addition, the evaluation form used in KaTIE does not contain prescribed questions to ask the children or specific things to observe as they interact with the software. The evaluation form provides general instructions to the facilitator (the teacher) to explore children's understanding of the content conveyed by the software. The openness of the evaluation form aims to encourage teachers to draw from their expertise and knowledge without being constrained by predetermined design-centric questions. This type of form was considered adequate as unanticipated issues the designer might have not thought of could be also identified by the teacher, hence adding extra value to the feedback otherwise collected by a prescribed form.

In terms of the evaluation outcomes, KaTIE's aim is to strengthen the commentary on the educational effectiveness of the software, without diminishing the usability and engagement components of the interaction. The ISE evaluation evidenced the predominant reporting and descriptive nature of the commentary produced by the teachers and the designers. KaTIE aims to increase diagnostic commentary, in the hope that these better account for educational effectiveness.

5.1. Comparing KaTIE and ISE

KaTIE's empirical evaluation consisted of four-field studies that aimed to contrast the commentary provided by the designer and the teacher in ISE with the combined commentary produced in KaTIE. All designers were given a brief

introduction to the ISE method and the KaTIE method. Three out of the four designers who participated were new to both ISE and KaTIE, thus there were no bias or preferences towards either. One of the designers received instructions only on KaTIE as she mastered the ISE method.

5.1.1. Data collection

The data collection for the evaluation of KaTIE resembled the one employed in the evaluation of ISE: video footage, interviews, and collection of artefacts. As mentioned before the findings here presented regarding these artefacts exclusively.

Each field study involved the participation of one software designer, one primary school teacher, and 6 children between 7–10 years old. Four software companies and four primary schools were involved working in pairs; each pair evaluated different educational software, which was created by the associated software company. The four educational software evaluated had different learning purposes, one aimed to teach English as second languages (ESL) children about insects and their habitats; the second introduced hospital terminology and some relaxations techniques; the third software was an argument-mapping tool for which some online exercises involving the concepts of reasons and objections were tested, and the fourth consisted of a reading program that involved some reading comprehension activities.

Each primary school was visited twice, one visit for the implementation of ISE and the other for the implementation of KaTIE. The order in which ISE and KaTIE were implemented was alternated across the four-field studies to minimize order effects. The number of commentary sets collected per field study was three, two collected through the ISE method (one belongs to the teacher and the other to the designer) and the third through the KaTIE method (teacher and designer independent and combined notes).

5.1.2. Data analysis

The analysis carried out on the commentaries collected followed three steps. The first step was coding according to the previously described coding scheme. The second step involved an audience review and the third step involved an expert review panel. These reviews had a three-fold purpose; first, they aimed to provide some measure of the level of usefulness and comprehensiveness of the feedback collected through ISE and KaTIE methods. Second, they served as a way of testing the coding scheme, as the reviewers were not given any coding scheme to guide their reviews. And third to contextualize the findings in terms of what was desirable feedback for designers in general.

The coding of the feedback produced by ISE method followed the same procedure as in the first evaluation presented. The feedback produced by KaTIE on the other hand required a slightly different coding approach as there were no prescribed questions to guide the identification of feedback instances. As the feedback collected through KaTIE tended to be narrative, clues such as dashes and bullets that implied a different idea or change in topic.

After the feedback was typed in and coded, the audience review was undertaken. The audience was defined as a member of the design team that was/had designed the software under evaluation. Therefore, a second software designer for every software company that participated played the role of audience. The audience was given a typed in version of the feedback to facilitate comparisons, but they were also given the original written notes for reference if required. The feedback was deidentified, so the audience did not know which feedback belonged to which evaluator (teacher, designer, or both).

The expert review consisted of a panel of three experts, who had background in education and learning technologies. All experts had a good sense of designing technology for learning and teaching, just one of them had direct experience designing and evaluating software for primary school children. The expert panel gathered together four times, every time to review the three sets of feedback belonging to one of the softwares evaluated. Experts were also given a typed in version of the feedback and one of the researchers was present during all review sessions. As these experts had no knowledge of the software evaluated they were introduced to it by the researcher to ensure the feedback was contextualized within the particularities of each computer program. As with the audience review, experts were not given any coding or framework for reviewing besides some general instructions.

These instructions consisted of the identification of any differences across the three sets of feedback and the selection, when possible, of the feedback that was perceived as most useful and more comprehensive regarding children's learning processes as they interacted with the software.

5.1.3. Findings

The findings of the feedback collected through KaTIE method are presented in three sections, coding, audience review, and expert review.

Coding

The coding results of the feedback collected through the ISE method were consistent with those yielded in the previous evaluation described in Section 4.2. There were fewer instances of diagnostic feedback as compared to the reporting and descriptive types. Equally, advisory feedback remained very low. There were not also major differences between the teachers and the designer's feedback. This confirms the homogenizing effect, previously identified, that a prescribed evaluation form has on the type of feedback collected. The feedback also referred predominantly to the learner verbal and nonverbal behavior as opposed to the interface/navigation of the software.

The coding of the feedback gathered through KaTIE was collected in three sections associated with the notes taken independently by the teacher and the designer during the data collection stage and the concluding notes taken by the designer during the reporting stage. Although all these notes were considered part of the feedback, the dissection allowed

a better account of where the different types of feedback were mostly being produced.

As it was expected, the notes taken by the designer and the teacher while observing the children interacting with the software tended to be of reporting and descriptive nature. One exception to this tendency was the notes taken by a teacher in one of the filed studies where the amount of diagnostic feedback was greater than the descriptive and reporting. This finding reflects the nature of the task teacher and designer were engaged in: observing and talking with the children; thus it can be reasonably expected that at this stage of the evaluation, less judgment would be recorded.

On the other hand, an increased number of diagnostic feedback was produced during the reporting stage wherein teacher and designer got to discuss their observations. The number of diagnostic feedback coded in the combined notes of the teacher and the designer resulted in an overall increase in the number of diagnostic feedback collected in KaTIE as opposed to ISE. This diagnostic feedback remained mostly focused on the learning aspects of the interaction despite the omission of prescribed questions.

An increased number of advisory feedback was also collected through KaTIE; however the number of those instances regarding children's interaction with the learning tasks or the overall instructional design was equal to those regarding the interface/navigation of the software. This finding suggests that teachers and designers also engaged in considering potential solutions to some of the issues found during the evaluation.

Audience review

All reviews collected from the audience identified similar types of feedback. This validated the coding scheme employed previously. The following interview extract evidences the similarity between the coding scheme and the audience views:

> "*... you really need to offer some view, some judgment as to what the implications of this observation is for improving the software or for improving the interaction between the kids and the software.... As a developer you often have to work with this and developers do not necessarily have an education background even though they are developing educational activities and so they need that interpretation there.*" Audience reviewer 4 [Diagnostic and descriptive feedback].

When the audience was asked to determine the usefulness and comprehensiveness of the feedback regarding the educational effectiveness of the software, the reviewers considered that the feedback collected through KaTIE was "too discursive" and that the format was difficult to read as the information followed a narrative style. This was considered cumbersome, as designers are often looking for the next steps they need to follow with the prototype. They also referred to the feedback in KaTIE as hard to interpret for a third party, as it lacked the structure the ISE feedback had in abundance.

As a consequence, the audience tended to select in most cases the feedback provided by ISE as their choice. However, some of the reviewers considered that the ISE and KaTIE feedback were complementary as the latter could provide a richer context for the answers and descriptions recorded in ISE. This finding shows that the feedback collected through KaTIE was overwhelming for most audience reviewers, hence it requires further filtering before a third party, who has not been in the field, could read it and make use of it.

In the context of formative evaluations that look into the educational effectiveness of the software, all types of feedback were considered desirable. Nonetheless, the diagnostic feedback as a "secondary source" type of feedback raised issues of reliability on the judgments made by the evaluators. The presence of advisory and diagnostic feedback was on the other hand seen as less useful if they were not complemented with other types of feedback that would provide context to particular suggestions or judgments. This shows that combined types of feedback are more useful to designers as compared to the single types. A reviewer affirmed that "things can always be done better," so if an advisory comment was not linked to a problematic issue identified with the software, this commentary would be considered less useful.

Expert review

The experts also identified very similar types of feedback across the four-group reviews undertaken. One drawback identified by the audience reviewers regarded the conciseness of some of the feedback found in ISE. Some of children's responses required the reader to "guess" how the children came about particular answers, and most importantly if the answers jotted down represented the first, second, or third attempt on the evaluator's behalf to uncover the children's understanding.

As the audience reviewers, the experts considered the presence of diagnostic feedback necessary when trying to determine the educational effectiveness of the software, as reporting and describing what the students are saying or doing is insufficient to convey the idea of the degree of engagement with the learning task. Nevertheless, the way in which the KATIE method supported the collection of diagnostic feedback was perceived, again, as not necessarily useful.

As it was mentioned earlier, most of the diagnostic feedback collected through KaTIE was produced during the reporting stage, hence these feedback tended to be detached of the reporting and descriptive feedback collected in the data collection stage. This was perceived by experts as problematic since for judgments to be informative a description of the context in which these are made is necessary. Some of the audience reviewers also expected these judgments to be grounded on evidence of children's behaviors or responses.

6. Discussion

Although both ISE and KaTIE can be situated in a hybrid space between child- and Expert-based evaluations, this is not so clearly reflected in the type of feedback collected by

them. The feedback produced using ISE is predominantly child-based, that is, reporting and describing children's behaviors; while diagnostic comments are less accounted for reflecting the minor role of expert-based feedback. We argue that the feedback produced with KaTIE is more balanced in this sense, increasing expert-based feedback without diminishing child-based feedback. Due to this balance, KaTIE conveys a richer picture of the children's learning experience by providing hybrid and merged feedback. As it was suggested by the reviewers, in the case of diagnostic feedback, there is a need to provide contextual information, such as descriptive or reporting type of feedback, for a more complete account of the educational effectiveness of the software. In this context, combined feedback as opposed to single types may ensue in more useful and comprehensive results.

The inclusion of teachers as experts in KaTIE has added value to the feedback collected. However, this value was hard to see given the format in which the feedback was collected, its narrative style needs to be refined. The prescribed evaluation form used in ISE guided the reader and dissected the information in more manageable chunks, while the open-ended evaluation form used in KaTIE perhaps serves better a data collection purpose rather than a reporting purpose.

In the context of current practice, KaTIE has shown that it is possible to account for the educational effectiveness of educational software within the boundaries of child-based evaluations and that resorting to independent expert reviews is of limited value. Experts can be included in the child-designer dyad and have a favorable influence on the evaluation outcomes. The use of measuring instruments, such as pre- and postwritten tests, to account for learning outcomes [11, 12] is restricted in the context of formative evaluations as they cannot provide the type of diagnostic feedback obtained with KaTIE. Through observation and dialog with the children, the formulation of diagnostic feedback takes into account contextual information that is not available on a written test.

The use of self-reported questionnaires [16] is not in opposition to KaTIE or ISE methods, as children's perceptions on their learning experiences are also fundamental. However, these should be treated as *perceived* educational effectiveness, which may not conform to the perception of experts and designers. In determining the educational effectiveness of educational software, the involvement of experts brings theoretical and experiential views to the evaluation, the participation of designers accounts for the design rationale behind the software, and the contribution of children accounts for their views on what they find easy, difficult, or fun to do when using the software.

7. Conclusions and future directions

We have demonstrated the added value that teachers can bring to the evaluation of educational software designed for children. By entrenching the teacher meaningfully in the evaluation process *as an expert educator*, we have shown that the diagnostic power of the evaluation and its sensitivity to pedagogical issues can be improved. In contrast to the views of earlier work, we have shown that both designers and children welcome the teachers into the process, and if supported teachers have much to offer to child-centered evaluation.

However, the evaluation process exerts a powerful influence over its practice, and great care is needed in its design if the teachers' voice is to be heard. Teachers' multifarious acts support the child, translate for the designer, and provide the pedagogical critique missing from a typical usability evaluation. Future work should examine the multitude of methodological options for the conduct of child/designer/teacher-centric evaluations, and indeed design more generally.

KaTIE and our findings, in regard to the inclusion of other members of a user's community in the process, may also apply to other situations, where psychological or social factors compromise the user's role in a participative process, or the setting of use is out of reach of the design team.

Acknowledgments

We would like to acknowledge the valuable participation of the software designers, teachers who willingly embarked on the process of implementing a new evaluation method. Also, we would like to thank all the children that took part in the evaluations.

References

[1] P. Francis, L. Firth, and S. Balbo, "Approaching the sensitive user with care: an online Delphi study," in *Proceedings of the International Workshop on Appropriate Methods for Design in Complex and Sensitive Settings. Australasian Conference on Human Computer Interaction (OZCHI '05)*, Canberra, Australia, November 2005.

[2] A. Druin, *The Design of Children's Technology*, Morgan Kaufmann, London, UK, 1999.

[3] A. Druin, "The role of children in the design of new technology," *Behaviour & Information Technology*, vol. 21, no. 1, pp. 1–25, 2002.

[4] P. Markopoulos and M. Bekker, "Interaction design and children," *Interacting with Computers*, vol. 15, no. 2, pp. 141–149, 2003.

[5] C. Jones, L. McIver, L. Gibson, P. Gregor, et al., "Experiences obtained from designing with children," in *Proceedings of the 1st International Conference for Interaction Design and Children (IDC '03)*, Preston, UK, July 2003.

[6] L. Hanna, K. Risden, and K. J. Alexander, "Guidelines for usability testing with children," *Interactions*, vol. 4, no. 5, pp. 9–14, 1997.

[7] J. Read, "Unpublished IDC master class notes," in *Proceedings of the 5th International Conference for Interaction Design and Children (IDC '06)*, Tampere, Finland, June 2006.

[8] M. Scaife, Y. Rogers, F. Aldrich, and M. Davies, "Designing for or designing with? Informant design for interactive learning environments," in *Proceedings of Computer Human Interaction Conference (CHI '97)*, pp. 343–350, Atlanta, Ga, USA, March 1997.

[9] S. W. Draper, M. I. Brown, E. Edgerton, et al., "Observing and measuring the performance of educational technology," TILT Project report 1, University of Galsgow, Glasgow, UK, 1994.

[10] S. MacFarlane, G. Sim, and M. Horton, "Assessing usability and fun in educational software," in *Proceedings of 4th International Conference for Interaction Design and Children (IDC '05)*, pp. 103–109, Boulder, Colo, USA, June 2005.

[11] G. Sim, S. MacFarlane, and J. Read, "All work and no play: measuring fun, usability, and learning in software for children," *Computers & Education*, vol. 46, no. 3, pp. 235–248, 2006.

[12] M. Virvou and V. Tsiriga, "Involving effectively teachers and students in the life cycle of an intelligent tutoring system," *Educational Technology & Society*, vol. 3, no. 3, pp. 511–521, 2000.

[13] P. Albion, "Heuristic evaluation of educational multimedia: from theory to practice," in *Proceeding of the 16th Annual Conference of the Australasian Society for Computers in Learning for Tertiary Education (ASCILITE '99)*, Brisbane, Australia, December 1999.

[14] J. W. Robertson, "Usability and children's software: a user-centred design methodology," *Journal of Computing in Childhood Education*, vol. 5, no. 3-4, pp. 257–271, 1994.

[15] P. Nokelainen, "An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students," *Educational Technology & Society*, vol. 9, no. 2, pp. 178–197, 2006.

[16] K. Beattie, "How to avoid inadequate evaluation of software for learning," in *Interactive Multimedia in University Educational: Designing for Change in Teaching and Learning (A-59)*, K. Beattie, C. McNaught, and S. Wills, Eds., pp. 245–258, Elsevier Science, Amsterdam, The Netherlands, 1994.

[17] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design, Beyond Human-Computer Interaction*, John Wiley & Sons, New York, NY, USA, 2002.

[18] R. Heller, "Evaluating software: a review of the options," *Computers & Education*, vol. 17, no. 4, pp. 285–291, 1991.

[19] L. van Leeuwen, "How children contribute to the design of technology for their own use," in *Developing New Technologies for Young Children*, J. Siraj-Blatchford, Ed., pp. 139–159, Trentham Books, London, UK, 2004.

[20] M. Prensky, "Digital natives, digital immigrants," *On The Horizon*, vol. 9, no. 5, pp. 1–6, 2001.

[21] V. John-Steiner, R. J. Weber, and M. Minnis, "The challenge of studying collaboration," *American Educational Research Journal*, vol. 35, no. 4, pp. 773–783, 1998.

[22] A. Monk, "Lightweight techniques to encourage innovative user interface design," in *User Interface Design, Bridging the Gap from User Requirements to Design*, L. E. Wood, Ed., pp. 109–129, CRC Press, Boca Raton, Fla, USA, 1998.