Can the teaching of HCI contribute for the learning of Computer Science? The case of Semiotic Engineering methods

Silvia Amélia Bim
Universidade Estadual do Centro-Oeste (UNICENTRO)
Depto. Ciência da Computação
Guarapuava – PR – Brasil
sabim@unicentro.br

Carla Faria Leitão
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Departamento de Informática
Rio de Janeiro – RJ - Brasil
cfaria@inf.puc-rio.br

Clarisse Sieckenius de Souza
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Departamento de Informática
Rio de Janeiro – RJ - Brasil
clarisse@inf.puc-rio.br

ABSTRACT
This paper presents the results of an in-depth qualitative study carried out to investigate the teaching and learning of Semiotic Engineering methods, namely the Semiotic Inspection and the Communicability Evaluation methods. We identified three kinds of abilities that are necessary for a better learning and application of these methods: *systematic interpretation*, *abstraction* and *wide perspective*. We have also identified the relation among these three abilities. In the triangulation step of the research we found that these abilities are also recurrently invoked as necessary to the learning of other Computer Science subjects such as programming, induction, and object-oriented design and development. In conclusion, we suggest that strategies used to teach Semiotic Engineering methods can explicitly explore the connections with other contents in the Computer Science curriculum and thus begin to reap mutual benefits.

Author Keywords
HCI Education, Computer Science Education, Semiotic Engineering, Semiotic Inspection Method, Communicability Evaluation Method, Systematic Interpretation, Abstraction, Wide perspective

ACM Classification Keywords
K.3.2. Computer Science education. H.5.2 User interfaces – Evaluation/methodology, H.5.2 User interfaces – Theory and methods

General Terms
Human Factors.

INTRODUCTION

This paper discusses the main results of a qualitative research carried out to investigate the teaching and learning process of two HCI evaluation methods [6] – Semiotic Inspection Method [10, 13] and Communicability Evaluation Method [10, 11, 14, 24]. Both methods were proposed by Semiotic Engineering [11], a semiotic HCI theory.

The Semiotic Engineering is a communication-centered HCI theory [10] aiming to support the communication between designers and users at interaction time. In this perspective the HCI phenomenon is a special case of computer-mediated human (designer-user) communication. Since the focus of the theory is communication it works with a qualitative approach and communicability is its HCI quality. To evaluate this criterion two methods were proposed – Semiotic Inspection Method and Communicability Evaluation Method.

The Semiotic Inspection Method (SIM) [10, 13] focuses the evaluation in the designer's message *emission* by analyzing the signs that composes this message. There are three types of signs: metalinguistic, static and dynamic. Metalinguistic signs are those that explicitly inform, illustrate or explain the meaning of static and dynamic signs. Examples of this type of sign are error messages, warnings, clarifications dialogs, tips and the system help itself. Metalinguistic signs can be static or dynamic. In Figure 1 the tip *"Pan right"* is an example of a simple metalinguistic sign.

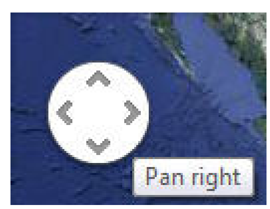

Figure 1: Examples of signs

On the other hand static signs are interface signs whose interpretation is limited to the elements that are present on the interface at a single moment in time, independently of temporal and causal relations. Toolbar buttons and menu options are examples of static signs. Figure 1 is also an example of a group of static signs from Google Maps interface: the navigation controls signs and the tip sign (which is also a metalinguistic sign).

And finally dynamic signs are those that emerge with the interaction. For example, when the user puts the mouse on the right control sign ">" the system shows the *"Pan right"* tip sign. The causal association between the mouse positioning on the right control sign and the appearing of the *"Pan right"* tip sign is a dynamic sign.

In the first three steps of the method the evaluator has to analyze each type of the signs in order to fulfill what we call the metacommunication template (a general schema of designer-user message):

"Here is my (the designer´s) understanding of who (the user) are, what I´ve learned you want or need to do, in which preferred ways, and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this version."

In the fourth step the evaluator has to compare the three instances generated in the previous steps. And finally, in the fifth and final step the evaluator has to reconstruct a unified message and give a final assessment of the system's communicability.

On the other hand the Communicability Evaluation Method (CEM) [10, 11, 14, 24] focus the evaluation in the *reception* of the designer's message. Consequently, in this method it is necessary to observe the user interacting with the system under analysis. The method is composed by three steps: Tagging, Interpretation and Semiotic Profile. In the first step the interaction video captured during the user interaction is analyzed by the evaluator in order to find communication breakdowns. There are thirteen breakdowns expression (tags) to categorize the communicability problems. Next, in the Interpretation step, the evaluator interprets the set of communicability breakdowns looking for recurrences, kinds of failures represented by the breakdowns, relations among the breakdowns and with other HCI kinds of problems, as usability ones, for example. In the last step (Semiotic Profile) the evaluator reconstructs the designer's message based on the message's reception by the user.

Informal statements from teachers who were teaching these methods pointed out that the qualitative nature as well as the complexity of theoretical basis of the methods were a challenge for the teaching and learning process. This scenario pointed out the necessity to investigate scientifically what was really happening.

Our first expectation before the investigation was discovered problems with the methods steps mainly related to clarity (for instance, the necessity of better definitions of parts of the methods) and to epistemological issues (for example, the lack of familiarity with non-predictive and qualitative scientific knowledge which is not common among computer science students). However, later, the results revealed that the reason for the teachers' and also students' and professionals' difficulty in dealing with the methods were in fact related to three abilities: *systematic interpretation*, *abstraction* and *wide perspective*.

Further in the research, while triangulating the results, we discovered that the same abilities were recurrently invoked as necessary to the learning of other HCI subjects – Usability Engineering [17] and Interface Design [5], for example, and to other Computer Science disciplines, like nondeterminism [1], object oriented approach [20], programming [30] among others. Researchers revealed that the students from these disciplines also have problems in their development of the three abilities mentioned before.

Abstraction is a well-known ability required for many, if not all, Computer Science disciplines and as so it has been widely discussed. Our research gives a step further presenting its relation with the *systematic interpretation* and *wide perspective* abilities. Moreover we found that these three abilities are necessary, in different levels and in different context of many Computer Science disciplines. It indicates that if one ability is exercised in a specific discipline, then the other ones can also reap the benefits.

So, we started the research looking for problems in the methods structure and epistemology and while analyzing the results we found problems on an upper level, namely, problems concerning the abilities to apply the methods. The qualitative research allowed us to investigate in depth from a small point and then discovered a broader set of results, similar to a bottom-up perspective.

The paper is structured in four sections. After the introduction, we present the methodology followed in the research. Then, we report our findings, illustrating the participants' statements that support them and discussing our findings in the light of related work, reporting the triangulation step of the research. In the last section we make some considerations about how strategies used to teach Semiotic Engineering methods can explicitly explore the connections with other contents in the Computer Science curriculum.

METHODOLOGY

An in-depth qualitative research [9] was conducted to investigate the teaching and learning process of SIM and CEM. The research was divided in three parts: (i) interviews with the authors, professionals, teachers and students of both methods; (ii) investigation of the progress of three undergraduate HCI disciplines; and (iii) triangulation with other HCI and Computer Science works.

In the first part of the research fourteen interviews were conducted, seven about SIM and seven about CEM, with every participant profile. The aim of this part was to identify the meanings that each participant assign to the teaching and learning process of the methods under investigation (advantages, difficulties, etc.) In the second part, the progress of three undergraduate HCI disciplines in three distinct Brazilian universities was analyzed. The data were collected from two individual interviews, conducted with each teacher, in the beginning and in the end of the semester. In addition the data from the diary classes, registered by the teachers in a virtual forum, were also analyzed. The aim of this second part was to identify didactic practices and the teachers' perspective about what was happening in the class context.

The results of each part were first analyzed separately and then integrated. These results were then triangulated with results from studies conducted in other Computer Science disciplines. The aim of the triangulation process is to validate the research results comparing and contrasting them with the results of other researches, looking of plausibility instead of replicability [8].

Part 1 – Data Collection

To recruit the participants for the first part of the research we used the purposive sampling with maximum variation [25]. The purposive sampling guaranties that the most representative participants – people who have rich data about the subject under investigation – are involved with the research. The maximum variation gives the opportunity to listen to different kinds of experience with the research subject. In spite of this variation, in our research the involvement with the Semiotic Engineering study was the common characteristic between the participants.

Fourteen in-depth interviews, with open-ended questions, were conducted in the first part of the research. Table 1 shows the participants' profiles, who were interviewed about CEM, concerning their HCI experience.

Participants	Time experience (average)	HCI experience
A_CEM	14 years	Teaching in the graduate and in the undergraduate level, master and doctoral leading, research projects coordination
P_CEM	1 year and 7 months	CEM application in scientific projects and exercises during the learning process
T_CEM	5 years	Teaching in the undergraduate level, exercises during the learning process, and professional and scientific evaluation projects
S_CEM	9 months	Exercises during the learning process

Table 1: Part 1 CEM participants' profile

In the first column the groups of participants are identified – A means the methods' authors, P means professional, T means teacher and S means students. For each group two participants were interviewed. The second column shows the average time of HCI experience. And the third column has the information about the kind of HCI experience.

Similarly, Table 2 shows the participants' profiles, who were interviewed about SIM, concerning their HCI experience.

Participants	Time experience (average)	HCI experience
A_SIM	13 years	Teaching in the graduate and in the undergraduate level, research projects coordination
P_SIM	4 years	SIM application in scientific projects and exercises during the learning process
T_SIM	6 years	Teaching in the graduate and in the undergraduate level, master and doctoral leading, research and professional projects coordination
S_SIM	8 months	Exercises during the learning process

Table 2: Part 1 SIM participants' profile

Part 2 – Data Collection

For the second part of the research we used the purposive sampling constructed in the homogenous way: three teachers of ongoing undergraduate HCI disciplines in Brazilian universities.

The results from the first part of the research were shared with the three HCI teachers who participated in the second part. During the interviews, from the reading of the report of the first part results, the teachers were asked about (i) their impression about the results, (ii) their experience in the HCI teaching and learning process, and (iii) their expectations for the discipline they were initiating.

During the semester the teachers also used a virtual forum to register some information about the classes: the date, the content and their impressions (the students' participation in the class, difficulties and facilities about the content).

At the end of the semester the teachers were interviewed again about their impressions concerning the discipline.

The three teachers selected to participate in the second part of the research have already contributed in the first part, not necessarily in the teachers' profile. The teachers have in average fourteen years of HCI experience – teaching in the graduate and undergraduate level, supervising master students and research projects coordination.

All the disciplines analyzed were taught in undergraduate courses. The disciplines were taught in sixty hours, divided in two weekly meetings of two hours each, in different moments of the courses. The content about Semiotic Engineering, SIM and CEM were distributed as Table 3 shows.

Content	Discipline1	Discipline2	Discipline3
Sem.Eng.	1 lesson	2 lessons	2 lessons
CEM	4 lessons	------	3 lessons
SIM	2 lessons	4 lessons	2 lessons

Table 3: Disciplines, contents and number of classes

Parts 1 and 2 - Data Analysis

The data collected were analyzed in two steps using discourse analysis [19, 25]. Firstly, an intra-participants analysis was made in the interviews transcription. This process consists in identifying categories of analysis present in each of the participants' statements for each topic in the interviews guide. Next, the second part of the analysis consisted in an inter-participant analysis, where the categories identified in the intra-participants analysis are compared looking for recurrences in each interviewed profile and then among the different profiles, as suggested by [19, 25].

The data were analyzed in a long process of successive and iterative steps of meaning assignment and categorization. In each iteration the categories identified in previous steps were revisited in order to reach an abstraction level even higher. The result of this process of analysis generated a categorized set of meanings that the participants have given to the subject under investigation and served as a guide to the researcher interpretation about the problem being investigated.

Part 3 - Triangulation

From the final categories we then did the research validation through the research results triangulation [9] with the analysis of other scientific works results about the HCI and other Computer Science disciplines teaching and learning process.

The triangulation process was conducted looking for similarities in the analysis categories. We would like to investigate which were the disciplines that shared the same results identified in our research. So, the analysis of the related work only happened after the identification of our results, namely, after the categories of analysis have been identified.

In the next section we present the research results with the focus on the three abilities needed to better learn, understand and apply the methods – *systematic interpretation*, *abstraction* and *wide perspective*.

RESULTS

Parts 1 and 2 – Data Analysis

The data analysis generated three classes of categories about the difficulties in the teaching and learning process of SIM and CEM: practical difficulties, difficulties in the development of three essential abilities for the learning process of SIM and CEM, and teaching initiatives to minimize the difficulties in the teaching and learning process of the methods under investigation [6]. In this paper we focus only on the three essential abilities needed to better learn, understand and apply the methods.

The three abilities identified are *systematic interpretation*, *abstraction* and *wide perspective*. The abilities were not necessarily explicitly named as so in the participants' statements (as can be seen further in this text). They came to light after long and careful iterative cycles of the researcher's interpretation, as is expected in a qualitative research. From the identification of the threes abilities we then looked for definitions suitable for the context of the research, which are presented in the following paragraphs.

Following the Peirce's semiotic perspective [21], *interpretation* is defined as the process of meaning assignment to signs [11]. In the context of scientific methods, it is essential to add another attribute to this definition: systematicity. Since this research is a methodological discussion about CEM and SIM, we define interpretation as a systematic process of meaning assignment to signs. It means that the interpretation must be oriented by specific categories. In CEM, for example, the interpretation of the communicability breakdowns must be oriented by the thirteen communicability expressions.

The definition of *abstraction* adopt in this research comes from the Kramer's [16] analysis about the Webster's [27] definitions for this concept. From this perspective the process of *abstraction* consists in removing details of a certain meaning to create generalizations, i.e., identifying the relationships among the meanings. In SIM this process occurs when, after identifying a group of signs, the metalinguistic ones, for example, the evaluator abstracts the individual characteristics of each sign identified to construct a generalization that instantiate the metacommunication template.

And finally, the *wide perspective* concept was constructed from the definition proposed by Aronson *et al.* [2] in the health context. Basically the *wide perspective* ability consists in the process of identification and establishment of the function among the relationships. For example, in the last step of SIM this ability is necessary to identify the function[1] of the relationships between the three metacommunication instantiations generated and compared in previous steps in order to check how good or bad is the communicability of the artifact being evaluated.

Once the concepts were defined, we comprehended from an articulated analysis of the participants statements that there is a precedence relationship among the three abilities. Based on the global analysis of the statements we identified that the *systematic interpretation* is the basis to *abstraction* which is on the other hand the basis to the *wide perspective*. The *interpretation* process establishes the meanings, the *abstraction* process establishes the relationships among the meanings by generalizations and finally in the *wide perspective* process the function among the generalizations is established. The Figure 2 illustrates the relationship among the three abilities.

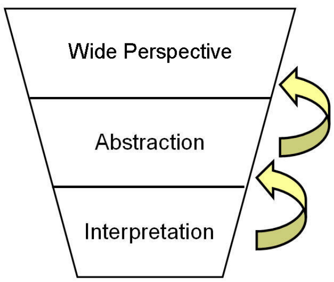

Figure 2: Relationship among the three abilities

In the next lines we will illustrate the relationship among the three abilities through some participants' statements about SIM and CEM steps. The whole path of interpretation followed by the researcher can be accessed in [6[2]].

Since *systematic interpretation* is the process of assign a meaning to signs, in SIM this ability is clearly necessary to identify and classify (assign a meaning to) the interface signs as metalinguistic, static or dynamic. The following statements illustrates the difficulties that different participants' profile have in identifying the signs, showing that there is a problem with the *systematic interpretation* ability.

"The distinction between static and dynamic signs generates many doubts..." – T1_SIM

"The difference between the statics and dynamics signs is not clear enough." – P1_SIM

"They [the students] frequently express more difficulty in comprehending the dynamic sign. The reason is that it involves behavior. I believe that they associate the sign with its representation (text or iconic) and express difficulty in comprehending that it may be behavior." – T2.3[3]

The participants comprehend the signs definition. The problem they face is in associating each sign in the interface to its right definition, in other words, in assigning the right meaning (metalinguistic, static or dynamic) to a sign. They have difficulties in the classification process. They are not able to remove a sign from the concrete context of the interface and sort it.

The difficulty with the *systematic interpretation* in CEM is shared both by students and professionals as the following statements illustrate.

"Because in my opinion the task is very complicated. I think that after doing, when you have experience in tagging it become clear to you. But in the first time it is really a very complicated task." – S1_CEM

"The biggest difficulty is to know the breakdowns expressions definitions. It´s difficulty to know the differences between them and also to know why you are doing that." – P2_CEM

These two statements show the difficulty that the evaluator has, both the beginner and the expert, in assigning a communicability problem in the user interaction with the system and the difficulty in choosing the correct breakdown expression to categorize the problem.

On the other hand the *abstraction* ability (the process of removing details of a certain meaning to create generalizations, i.e., identifying the relationships among the meanings) is necessary in different moments of SIM. The two next statements talks about the difficulty in abstracting the problems identified in classes of problems and the difficulty in abstracting the relationship among the messages generated by different types of signs to comprehend the metacommunication.

"I remembered that I found many problems… but I wasn´t able to abstract them into classes of problems." – A2_SIM

"It´s necessary to have knowledge to be able to "see" through the different messages. It´s necessary to be able to analyze the messages and comprehend the metacommunication message." – T2_SIM

In the case of CEM the *abstraction* ability is necessary in the tagging consolidation and in the Interpretation step as the following statements illustrate.

"The Tagging in general is difficult because it depends on interpretation. Because you have to decide between one or another breakdown expression to categorize a problem. And after it happens in the consolidation. Because in the Tagging everything is disconnected. And then you have to

---

1 To check how good or bad is the communicability of the artifact being evaluated, for example.

2 The text of this reference is in Portuguese.

3 Participant 3 (teacher) of the research's part 2.

connected everything and the Tagging itself doesn´t matter." – P1_CEM

"It is possible that something occurs once but in fact it is the iceberg point that will help you to interpret the whole thing." – A1_CEM

Besides indicating the necessity of the *abstraction* ability these statements also show the precedence relationship between the *interpretation* ability and the *abstraction* one. If the evaluator has difficulties in identifying the communicability problems (assign the meaning of "problem" to a specific event in the interaction video) and make the association with the breakdowns expressions (assign the meaning of a specific tag to the problem identified), then in the next step, it means Interpretation, the results will be harmed.

Finally, the *wide perspective* ability (process of identification and establishment of the function among the relationships) is necessary to reconstruct the designer's metacommunication message as mentioned by one of the SIM's authors in the following statement.

"The students have many difficulties (and so do I, sometimes) in reconstructing the metacommunication message. What happens is that they do the first three steps and then present a list of problems with redesign suggestions." – A2_SIM

This author's opinion is shared by the other SIM's author interviewed who talks about the students' difficulty with the *wide perspective* ability, as mentioned in the next statement.

"In the other end, they have a HUGE difficulty in having a wide perspective about the application, in seeing how the things are connected with each other and other things. (…) They have a very fragmented vision about the things (…)" – A1_SIM

The participants interviewed about CEM also expressed the necessity of the *wide perspective* ability. Next we present the statements of two participants, one teacher and one student, talking about the necessity of the *wide perspective* in two moments of the method, in the Tagging and in the Interpretation.

"The difficulty in this step [Tagging] and also in the Interpretation, because this gives a direction for the Interpretation, is because I have to comprehend the process, it is not a punctual thing." – T1_CEM

"I think that [Interpretation] is the most complicated step. In my opinion it was really the most complicated step. You have to define the importance of each breakdown expression. Suddenly a tag occurs many times but it is irrelevant. And on the other hand the one that occurs once has a huge impact in the interaction." – S1_CEM

To define the importance of a communicability problem, expressed by a communicability tag, the evaluator has to see the whole set of tags and identify the relationship among them and with the entire interaction. Once again, we can see the relationship among the three abilities.

Part 3 – Triangulation

Once the categories of analysis were identified, we started the triangulation step of the research, looking for works concerning the investigation of the teaching and learning process of HCI and other Computer Science disciplines. The aim of this part of the research was to validate the results identified in the previous part searching for plausibility. In this section we comment about the works which share, in some way, the findings regarding the three abilities: *systematic interpretation, abstraction and wide perspective.*

The study conducted by Yuen [30] identified that students have difficulties in explaining and applying the concepts of function and recursion. They are able to memorize the concepts but they have difficulties in explaining it, better saying, they are not able to assign a meaning to the concepts (*interpretation* ability) or to construct the relationship among the concepts and the code of a program (*abstraction* ability).

A similar problem was identified by Polycarpou *et.al.* [22] in the induction learning process. The deficiency in the students' *interpretation* and *abstraction* abilities makes the comprehension of the proof by induction a challenge. When the students "succeed" in general the problem is similar to the exercises problems discussed previously in the classroom. The students have difficulties using the proof by induction in new and different problems.

The *abstraction* ability is also essential in the Usability Engineering lessons. According to Leventhal *et.al.* [17] teachers face a challenge when teaching the specification phase. Students consider that everything is so abstract and have many doubts when defining user tasks from the problem scenarios.

While investigating the possible causes for high attrition rate for Computer Science students Beaubouef and Mason [3] identified that the ability to solve problems is often very weak in Computer Science students. To solve a problem in the Computer Science context it is necessary to do iterative and successive cycles of *interpretation*, *abstraction* in order to construct a *wide perspective* of the problem and then implement a computational solution.

According to Carroll [7] the *wide perspective* question is a problem of the HCI area as a whole. There are so many theories, methods and approaches in the area and their articulation frequently does not exist either in theory or in practice or in the classroom. The consequence is a huge fragmentation which contributes to the students and teachers difficulty in constructing an articulated view about the area and consequently in developing their *wide perspective* ability.

Berkun [5] also discusses the necessity of the *wide perspective* ability in the interface design perspective. In his study he identified that the *wide perspective* ability is necessary to HCI professionals to develop their capacity to deal with a problem in the strategic level because to solve the problem only in the tactic level is not sufficient.

More recently, studies concerning the computational thinking necessity reinforce the need of *abstraction* ability. De Souza *et.al.* [12] identified that a group of 9th-grade students from a public school could do better in an AgentSheets activity if they had the appropriate programming abstractions to avoid duplication of code. It can be seen as an indication that the programmers' *abstraction* ability can be a differential to improve the quality of the programs' code.

Although the works mentioned are not statistically representative they are valuable seeds to our reflection about the necessity of the three abilities mentioned in this paper – *systematic interpretation, abstraction* and *wide perspective* – for different disciplines in the Computer Science curriculum. Once the problem has been identified by several researchers the questions that remain are "Why the problem persists?" and "How to solve it?".

In the next section we make some considerations about these questions and briefly describe a case showing how HCI teaching can contribute to the learning of Computer Science.

FINAL CONSIDERATIONS

Although the *interpretation*, *abstraction* and *wide perspective* abilities are clearly necessary to many Computer Science subjects, if not all of them, until now the students, and sometimes even the teachers, face problems with these abilities. So, how we can solve this problem? Computer Science teachers expect that their students begin the course with these abilities well developed. However, some studies show that even in industrialized societies the college students' ability to reason with abstractions is strikingly limited [18].

Computational thinking projects are offered for many schools level all over the world [12, 28, 29] with the aim to offer a fresh approach to build many abilities as abstract thinking, problem solving and deductive reasoning, for example. The Computer Science Unplugged project [4] is another initiative to develop the abilities and the knowledge necessary to computer scientists through activities without computers. The activities are recommended mainly to kids in elementary school. But what we can do for our students that have already chosen a Computer Science course and shows difficulty in the *interpretation*, *abstraction* and *wide perspective* abilities? One possibility is presented by Kafura and Tatar [15] who share an experience of teaching computational thinking for Computer Science students. And what can be done in the HCI context?

The observation of an experience in a Brazilian university during the first semester of 2012 suggests that the HCI discipline may contribute to the development of the three abilities identified in this project which is also necessary in other disciplines of the Computer Science curriculum.

During a HCI discipline students can develop abilities related to communication with users (in interviews) and to ethical procedures (involved in research with human beings), which are not usually developed in traditional computer science disciplines. During this learning process, it is necessary to practice the abstraction ability to *translate* the users' need to interaction and interface models, for example. In the case of Semiotic Engineering methods under investigation (CEM and SIM) we also identified that *systematic interpretation*, *abstraction* and *wide perspective* are intensively exercised. The triangulation step of the research revealed that these abilities are required in other Computer Science subjects such as programming, induction, and object-oriented design and development. Consequently, when learning and applying CEM and SIM the students are exercising their *systematic interpretation*, *abstraction* and *wide perspective* abilities which may help them in the learning process of other Computer Science disciplines.

As an illustration to our discussion, let us show a brief example of part of an introductory HCI discipline taught in the first semester of a Computer Science course in a Brazilian university. We will focus on the first written test applied in this discipline.

The test was answered by 22 students. It was composed of three questions each one divided in queries. The first question was about general concepts about HCI. Queries Q1a and Q1b were related with the interdisciplinary characteristic of the discipline. The students were asked to answer an HCI aspect that only Computer Science can deal and an HCI aspect that Computer Science cannot deal with. In both case the students need to justify their answer. Looking Table 4 we can see that for the first query (Q1a) only half of the students succeed in giving a completely right answer. In the second query (Q1b) the score is even worse. We consider that to answer a correct answer for these questions the students need to put in practice their *interpretation*, *abstraction* and *wide perspective* ability. They have to assign a meaning to an aspect that is an HCI one, then they have to identify the relationship of this aspect with the Computer Science area and finally they have to construct a wide perspective of the interdisciplinary context of HCI to comprehend why other disciplines cannot deal with the aspect chosen.

Query Q1c was about the ethics aspect in HCI studies and we considered that only the *interpretation* ability is necessary. Looking the score in Table 4 however, we can see that the students have difficulties with this query too. Query Q1d asked about the conceptual difference between "interface" and "interaction" concepts. Query Q1e was related to Q1d and asked the students to illustrate the previous answer using an example given in the classroom and in a print screen picture given in the test. Once again the students showed difficulties in the *interpretation* ability when asked to describe the concepts. None of the students could give a completely right answer for question Q1d. The teacher's discipline considered that the students made considerable mistakes about the concepts. Finally, the last query about HCI general concepts was related to the Nielsen's heuristic. The students succeed in identifying the design errors (*interpretation* ability) but they have difficulties in comprehending the relation between these errors and the heuristics (*interpretation* and *abstraction* ability).

Question	100%	90% - 25%	0%
Q1a	11	2	9
Q1b	5	6	11
Q1c	9	7	6
Q1d	0	12	10
Q1e	2	6	13
Q1f	4	7	11

Table 4: Queries of question 1 and the number of students for each percentage of correctness in their answers

Question 2 was related to HCI evaluation methods. The queries were about the differences between inspection methods and user test (Q2a), about the technique that is not suitable when the evaluation model approach is chosen (Q2b) and the differences between usability test and field studies (Q2c). The main problem with this question is that the students had difficulties in justifying their answers showing doubts about the concepts involved (*interpretation, abstraction* and *wide perspective* abilities). Table 5 presents how students did in each query of question 2.

Question	100%	90% - 25%	0%
Q2a	7	5	10
Q2b	3	5	14
Q2c	6	4	12

Table 5: Queries of question 2 and the number of students for each percentage of correctness in their answers

The last group of queries was about the Cognitive and Semiotic Engineering models. Queries Q3a and Q3b were related to Cognitive Engineering and the students were asked about the Execution and Evaluation Gulfs in an example of interaction presented in the test. And finally queries Q3c and Q3d were related to Semiotic Engineering concepts as metalinguistic, static and dynamic signs, the communicability quality and the designer's metacommunication in the same example of queries Q3a and Q3d.

Interestingly the students did better in the last two queries (Q3c and Q3d) which are related to Semiotic Engineering concepts, as can be seen in Table 6. This table shows that the number of students who did not were able to give a correct (or partially correct) answer – 0% of correctness – to these last two queries decreased significantly compared to the performance in the other queries. Our hypothesis is that the communication perspective is better understood by the students, even in the first semester of a course.

Question	100%	90% - 25%	0%
Q3a	7	5	11
Q3b	5	1	16
Q3c	9	9	2
Q3d	7	12	3

Table 6: Queries of question 3 and the number of students for each percentage of correctness in their answers

It is important to mention that although the average score was 2.9 and the median was 4.5 the scores distribution (Table 7) shows that the level of the difficulty in the test were adequate considering the fact that in the set of students who got between 0.0 and 2.9 only one students assists the lessons regularly.

Score	Number of students
> = 7.0	4
Between 5.0 and 6.9	7
Between 3.0 and 4.9	3
Between 0.0 and 2.9	8

Table 7: Score distribution

The students' facility to comprehend the Semiotic Engineering concepts, as showed in this brief example of the first written evaluation of an introductory HCI discipline may be seen as a good opportunity to develop the *systematic interpretation*, *abstraction* and *wide perspective* abilities through this semiotic HCI theory concepts and methods.

Furthermore, this case is an example that an introductory HCI discipline may help teachers to identify the students' *interpretation*, *abstraction* and *wide perspective* abilities and others as writing for example. Writing, in fact, is a very valuable ability not only during the academic studies but also in the professional context [8]. However, until now the initiatives in the academic scenario did not give the expected result. HCI disciplines can be a good opportunity to explore this ability in the Computer Science context.

The students' better performance in the last two questions of the test indicates that they are more familiar to communication concepts. However it does not means that their writing ability is well developed. A HCI discipline can give the students the opportunity to comprehend the importance of this ability during their course. While analyzing an HCI situation the students are in contact with both technology and human aspects. So the students have the opportunity to write about technology. And in a computational contextualized writing activity the students have more chances to comprehend that their writing ability is very important for their success during the course and in their career. However, the writing activities must be planned following recognized strategies [8] to make them more effective.

Silveira and Prates [26] presents the results of a group study realized in 2006 which have generated suggestions to the HCI curriculum for undergraduate and graduate courses. More recently Prates and Filgueira [23] share the result of a research about the Brazilian reality concerning HCI. However none of these works mention when the HCI disciplines are taught during the Computer Science courses. Our experience and knowledge about the teaching of HCI in Brazil says that the period when HCI disciplines are taught varies from university to university.

Although, as the HCI discipline example briefly presented in this paper shows, the sooner the discipline is offered early the students' difficulties are identified and early the teachers can help the development of the necessary abilities. Another contribution that the teaching of HCI in the first semester of a course can bring is the construction of a wide perspective about the Computer Science area. HCI has the capability to show to students that the area involves technology and people and many issues related to each of these two important *subjects*. And consequently, the students can understand early that the Computer Science professional has to develop different and complementary abilities to have more chances of success. In addition this wide perspective about the Computer Science may contribute to the reduction of the well-known problem of the courses evasion in the area [3].

ACKNOWLEDGMENTS

Silvia Amelia Bim wants to thank CNPq for her scholarship and UNICENTRO for her license to do her PhD. The authors also like to express their gratitude to the participants that voluntarily contributed with this research.

REFERENCES

1. Armoni, M.; Lewenstein, N.; Ben-Ari, M. Teaching students to think nondeterministically. SIGCSE Bull. ACM, New York, v. 40, n. 1, p. 4-8, mar. 2008.
2. Aronson, R.E.; Norton, B.L.; Kegler, M.C. Achieving a "Broad View of Health": Findings From the California Healthy Cities and Communities Evaluation. Health Education & Behavior. SAGE, v. 34, n. 3, p. 441-452. 2007.
3. Beaubouef, T.; Mason, J. Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. SIGCSE Bull. ACM, New York, v. 37, n. 2, p. 103-106. 2005.
4. Bell, T.; Alexander, J.; Freeman, I.; Grimley, M. Computer Science Unplugged: school students doing real computing without computers. New Zealand Journal of Applied Computing and Information Technology, 13 (1). pp. 20-29. 2009.
5. Berkun, S. Perspectives on User Interface Design Training. Interactions. ACM, New York, v. 9, n. 2, p. 123-125. 2002.
6. Bim, S.A. Obstáculos ao ensino dos métodos de avaliação da Engenharia Semiótica. Tese de Doutorado - Departamento de Informática, PUC-Rio. Rio de Janeiro, 2009.
7. Carroll, J.M. HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science. Amsterdam: Morgan Kaufmann Publishers. 2003.
8. Cilliers, C.B. Student perception of academic writing skills activities in a traditional programming course. Computers & Education, v. 58, issue 4, p. 1028-1041. may 2012.
9. Creswell, J.W. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. 3rd Edition. Thousand Oaks. SAGE Publications. 2009.
10. De Souza, C. S.; Leitão, C.F. Semiotic Engineering Methods for Scientific Research in HCI. Morgan & Claypool Publishers. 2009.
11. De Souza, C.S. The Semiotic Engineering of Human-Computer Interaction. Cambridge, MA: The MIT Press. 2005.
12. De Souza, C.S.; Garcia, A.C.B.; Slaviero, C.; Pinto, H.; Repenning, A. Semiotic traces of computational thinking acquisition. In Proceedings of IS-EUD'11, Springer-Verlag, Berlin, Heidelberg, p. 155-170. 2011.
13. De Souza, C.S.; Leitão, C.F.; Prates, R.O.; da Silva, E.J. The Semiotic Inspection Method. In: Anais do VII Simpósio sobre Fatores Humanos em Sistemas Computacionais. SBC: Natal, RN. p. 148-157. 2006.
14. De Souza, C.S.; Prates, R.O.; Barbosa, S.D.J. A Method for Evaluating Software Communicability. In C.J.P. de Lucena (ed.) Monografias em Ciência da Computação. PUC-Rio Inf MCC11/99. Departamento de Informática, PUCRio. 1999.
15. Kafura, D.; Tatar, D. Initial experience with a computational thinking course for computer science students. In Proceedings of SIGCSE '11. ACM, New York, NY, USA, p. 251-256. 2011.

16. Kramer, J. Is abstraction the key to computing? Communications of the ACM. ACM, New York, v. 50, n. 4, p. 36-42, abr. 2007.
17. Leventhal, L.M.; Barnes, J.; Chao, J. Term Project User Interface Specifications in a Usability Engineering Course: Challenges and Suggestions. In Proceedings of the SIGCSE’04. ACM Press, p. 41-45. 2004.
18. Moursund, D. Computational Thinking and Math Maturity: Improving Math Education in K-8 Schools. Avaiable at http://www.uoregon.edu/~moursund/Books/ElMath/K8-Math.pdf. 2007
19. Nicolaci-da-Costa, A.M.; Leitão, C.F.; Romão-Dias, D. Como conhecer usuários através do Método de Explicitação do Discurso Subjacente (MEDS). In: Anais do VI Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais. SBC: Curitiba-PR, p. 47-56. 2004.
20. Or-Bach, R.; Lavy, I. Cognitive Activities of Abstraction in Object Orientation: An Empirial Study. SIGCSE Bull. ACM, New York, v. 36, n. 2, p. 82-86. 2004.
21. Peirce, C. S. The essential Peirce (Vols. I and II). Editado por Nathan Houser e Christian Kloesel. Bloomington, IN: Indiana University Press. 1992, 1998.
22. Polycarpou, I.; Pasztor, A.; Adjouadi, M. A conceptual approach to teaching induction for computer science. In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education. SIGCSE '08. ACM, New York, p. 9-13. 2008.
23. Prates, R. O.; Filgueiras, L. V. L. Usability in Brazil. Global Usability – Human-Computer Interaction Series. Douglas, I., Liu, Z (Eds), Springer. 2011.
24. Prates, R.O.; De Souza, C.S.; Barbosa, S.D.J. Methods and tools: A method for evaluating the communicability of user interfaces. Interactions. ACM, New York, v. 7, n. 1, p. 31-38. 2000.
25. Seidman. I. Interviewing as Qualitative Research: a guide for researchers in Education and the Social Sciences. New York: Teachers College Press. 1998.
26. Silveira, M. S.; Prates, R. O. Uma Proposta da Comunidade para o Ensino de IHC no Brasil. Anais do XV WEI, XXVII Congresso da SBC, SBC, p. 76-84. 2007.
27. Webster´s. Webster’s Third New International Dictionary. 1966
28. Wentworth, P. Can Computational Thinking Reduce Marginalization in the Future Internet? Proceedings of the 2010 ITU-T Kaleidoscope Academic Conference. India, p. 1-5. 2010.
29. Wolz, U.; Stone, M.; Pearson, K.; Pulimood, S.M.; Switzer, M. Computational Thinking and Expository Writing in the Middle School. *Trans. Comput. Educ.* 11, 2, Article 9, July 2011.
30. Yuen, T.T. Novice’s Knowledge Construction of Difficult Concepts in CS1. SIGCSE Bull. ACM, New York, v. 39, n. 4, p. 49-53. 2007.