

Exploring the Effect of Teaching Methods on Students' Learning of School Informatics

Said Hadjerrouit
University of Agder, Kristiansand, Norway

Said.Hadjerrouit@uia.no

Abstract

School informatics is a relatively new subject in comparison with mathematics. As a result, there is much less research in informatics education. Likewise, there is lack of theoretical grounding of didactics of informatics in general and school informatics in particular. The lack of research in the field of school informatics makes it difficult to identify which methods are effective to teach the subject. In addition, despite the emphasis on a set of well-defined topics, the New National Curriculum in Norway makes few suggestions as to which didactical skills teachers need to acquire in order to teach informatics-related subjects. In fact, school informatics is still in its infancy. It lacks an extensive research base of materials that can be used to achieve educational goals and learning outcomes. The main research objective of this work is to investigate the effectiveness of teaching methods and their impact on learning school informatics.

Keywords: Interaction, learning pyramid, learning theories, school informatics, teaching methods

Introduction

Despite the emphasis on a set of well-defined informatics topics and clear competency aims and goals, the New National Curriculum in Norway from 2006 does not provide clear information on the methods that are needed to teach these topics to achieve the goals. Basically, informatics as a school subject is confronted with a number of closely related problems, which makes the teaching of informatics more difficult than traditional subjects, such as the complexity of the software being used, lack of training and qualifications in informatics, existing teaching resources and textbooks, nature of informatics subject, etc. School informatics is a relatively new subject, and it is rapidly evolving, but it lacks an extensive research base of materials like the one published in mathematics education (Cápay, Magdin, & Tomanová, 2012; Dagdilelis et al., 2004; Dagdilelis, & Xinogalos, 2012; Hammond, 2004; Hadjerrouit, 2008, 2009a, 2009b; Kumar, 2006; Minaidi & Hlapanis, 2005; Schubert & Swill, 2004; Schulte, 2002; Webb, 2002). While some progress has been made in teaching digital literacy, informatics education “lags sorely behind” (JIE & ACM

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

WGIE, p. 15). In fact, “informatics education has retreated in most European curricula since pioneering efforts in the 1970s and 1980s” (JIE & ACM WGIE, p. 15). As a result, there is little research on which methods are suitable for teaching informatics in secondary schools, and what students think about their learning effect. Clearly, there is a need for research to gain more knowledge on the subject and the way it should be

taught to achieve learning benefits. The main goal of this work is to assess students' perceptions of teaching methods in relation to their effect on learning informatics in secondary schools. It uses a theoretical framework that classifies teaching methods in 4 categories: (a) Whether the methods promote behavioristic, constructivist, or socio-cultural learning; (b) Whether the methods are formal or informal; (c) the type of interaction the methods imply, and finally; and (d) their rank or position in the Learning pyramid.

Although this work used a relatively small sample with a limited source material to investigate this issue, it is hoped that this study makes a contribution to the knowledge base of school informatics. This study is situated within teacher education at the university level in collaboration with two classes at the upper secondary schools level, where teaching practice of six weeks was performed. The main goal of this work is to answer the following question: What teaching methods do students find most helpful in learning school informatics? To address this question, a survey questionnaire with open-ended questions is designed to ask school students in two different classes with respectively 15 and 25 participants about their perceptions on the learning effect of teaching methods.

The paper is structured as follows. First, the current state of informatics education in secondary schools is described. Secondly, the theoretical perspective on teaching methods is outlined. This is followed by the research design of the work. Then, the results, discussion, and implications are reported. Finally, some remarks conclude the article.

Current State of Informatics Education in Secondary Schools

Informatics as a school subject in the New National Curriculum in Norway from 2006 gives more emphasis on practice than theory. It is divided into Information Technology I and II. Information Technology I includes four topics: *Digital equipment, programming, multimedia applications, and Web development I*. Information Technology II includes three topics: *Information systems design, databases, and Web development II*. Teachers and students need to be familiar with these topics in order to acquire digital competency, which involves the acquisition of a number of informatics skills. The first skill is concerned with using software tools, such as Excel, Dreamweaver, Flash, Photoshop, Access, and similar software for problem solving, simulation, modeling, exploration, visualization, and publication. The second skill is about using the Internet for finding information, analyzing, processing, and presenting data with appropriate technological aids, and to be critical of sources and results. Then, students need to learn about Web design, development of interactive multi-media applications, connecting Web applications with databases, publishing and maintaining Web sites, as well as user-centered design and associated evaluation criteria, such as ease-of-use and ease-of-learning, subjective satisfaction, etc. Furthermore, the curriculum requires the acquisition of software development skills and approaches, both object-oriented and conventional methodologies, for analyzing problem situations, modeling, implementing, and testing appropriate solutions. Finally, programming with Action Script and similar languages is also a necessary skill. Acquiring programming skills means analyzing the problem to be solved, breaking down the problem-solving process into its components, and designing an algorithmic solution before coding, programming the solution, and testing it on the basis of some quality criteria.

However, despite the emphasis on a set of well-defined topics and skills, the New National Curriculum makes few suggestions as to which pedagogical knowledge and strategies are needed to teach these topics. In fact, school informatics education is still in its infancy. It lacks an extensive research base of materials like the one published for mathematics education, a discipline that is well established at the university level in Norway. According to the research literature (Dagdilelis

et al., 2004; Hadjerrouit, 2008; Hammond, 2004; Hubwieser, 2004; Minaidi & Hlapanis, 2005; Webb, 2002), informatics as a school subject is confronted with a number of closely related problems that need innovative solutions. First, schools adapt slowly to technological and pedagogical changes despite the fact that many attempts at implementing informatics have been initiated since 1980 in many countries. In addition, teachers are often reluctant to abandon their own pedagogy. As a result, teaching methods based on conventional teacher-directed epistemologies are still dominant despite the potentialities of innovative pedagogies based on constructivism and situated learning (Avidov-Ungar & Iluz, 2014; Ben-Ari, 2001, 2004; Brodahl, Fagernes, & Hadjerrouit, 2007; Hadjerrouit, 2008, 2009a; Fee, & Holland-Minkley, 2010; Kolikant, 2010). Furthermore, teachers lack specific training and qualifications in teaching informatics (Schulte et al, 2012). A major problem for teachers is the complexity of software being used to teach informatics. While knowledge about the principles and functionality of software is needed to teach informatics, it is difficult for teachers to know all the specific features of the software tools that they use in classroom settings, because software is continually being developed and improved. Programming is still considered as a challenging task (Bennedsen, & Caspersen, 2012). Very often students lack deep level learning skills and problem-solving skills. Learning is hindered by inadequate support, unfamiliar pedagogical approaches, and many other factors (Apiola, & Tedre, 2012). Another important problem for teachers is identifying suitable context of informatics use, since the subject is closely linked to almost all other subjects such as mathematics and science (Armoni, 2011). These contexts may be taken from the school context, the wider community, for example private or public sector, or from another subject area. Moreover, teaching resources and textbooks in informatics are still based on traditional pedagogies. Textbook publishers have difficulties to address innovative teaching approaches that are suited to informatics, mostly because they are driven by marketing forces rather than pedagogical considerations. Finally, a number of informatics topics are more difficult to teach than other subjects, because they require a much greater range of professional and pedagogical skills and knowledge than those required to teach a unit of work within a specific subject area such as mathematics. Other difficulties are: insufficient use or superficial adoption of learning theories, misapplication of acquired knowledge, teacher's adherence to textbooks, lack of innovative methods (Dagdilelis & Xinogalos, 2012). Given the current state of informatics education, there is a need for an integrated informatics teaching strategy that recognizes the interrelations of pedagogy, informatics skills, students' needs, and school environment to successfully exploit the potential capabilities of informatics in secondary education.

Teaching Methods: Theoretical Perspective

What constitutes good or poor teaching methods is not easy to define, because these are influenced by teachers' normative assessments, the kind of learning the methods are supposed to foster, students' characteristics, the tools being used, physical and social environment, and many other factors from society, school institution, and pedagogical and cultural factors as well (Imsen, 2009). Basically, teaching methods should aim at promoting learning among the students and be adapted to competency goals of the curriculum, foster higher order skills such as "examine", "analyze", "discuss" and "explore", and provide guidelines on how the methods should be used (Repstad, 2006). According to Ogden (2004), a prerequisite for effective teaching is that "teachers master a broad range of teaching principles and know how they can be applied in different situations to promote learning" (free translation). From a practical point of view, no teaching method is better than another, even though methods differ considerably from a theoretical point of view. In authentic educational settings, methods must be adapted to the students' and teacher's characteristics, interests and needs, as well as to the subject matter being taught and educational context where it is taught (Ogden, 2004). Given this background, the most important theoretical background material are the following teaching methods that are numbered from 1 to 10. The rationale behind the choice of the teaching methods is that these are the most commonly used methods in

school informatics in Norway, depending on the learning environment and other factors, such as teachers' experience, students' knowledge level, etc. This study does not aim at using innovative methods as in a previous research study (Hadjerrouit, 2009). The survey questionnaire with 10 statements is closely connected to the teaching methods listed below:

1. Teaching with blackboard/overhead projector
2. Demonstration of software tools
3. Solving software exercises with teacher assistance
4. Project work over several days or weeks
5. Group work under the guidance of the teacher
6. Individual problem solving
7. Homework assignments and ask friends, siblings, or parents for help
8. Reading textbooks and user manuals
9. Using study material made online by the teacher
10. Present solutions to problems in front of the class

The nature of these methods is not defined by their goals, strengths, or limitations, but classified according to 4 broad categories that are addressed in more details in the following sections:

- Teaching methods and learning theories
- Formal and non-formal teaching methods
- Type of interactions involved in the teaching methods
- Rank of teaching methods in the Learning pyramid

Teaching Methods and Learning Theories

Students should consider what kind of methods they felt they learn most from. According to McGonigal (2005), teaching is considered as a transformation process from learning theories to teaching strategies. Learning theories can be categorized as methods that promote behavioristic, constructivist or socio-cultural learning.

The behaviorist learning theory emphasizes external influence and the behavior it causes. Stimulus and response are key concepts that define behaviorist learning. Behaviorism assumes that the human mind is a passive recipient of knowledge, and since it is not possible to observe or measure what is happening inside a human being, researchers focus on external behavior that can be observed (Driscoll, 2000). Teaching methods that promote behaviorist learning are those that focus on teacher-directed instruction, systematic training, and repetition. Behavioristic methods are tailored to external stimulation of students. The following methods are typically associated with methods that foster behaviorist learning:

1. Classroom teaching with blackboard/overhead projector.
2. Demonstrations of software tools

The advantage of these methods is that they are practicable and easy to use in classroom and ensure that the students "get through" the curriculum. The downside is that there is little personalization, and that students develop little understanding but only mimic or repeat the knowledge they are supposed to acquire, and that all must follow the same progression regardless of conditions, interests or needs.

Constructivist learning rejects the behavioristic view that knowledge is an objective truth, which exists out there, which can be acquired by correcting the knowledge of new sensory data. Rather, the constructivist learning theory views knowledge as a constructed entity made by each learner through a learning process. Constructivism frames learning less as the product of passive transmission than a process of active construction whereby the learners construct and reconstruct their own knowledge based upon prior knowledge (Duffy, Lowyck, & Jonassen, 1993; Piaget, 1971;

Steffe & Gale, 1995; Taber, 2011). Constructivist learning requires learners to demonstrate their skills by constructing their own knowledge when solving real-world problems. The constructivist model calls for learner-centered instruction, because learners are assumed to learn better when they are forced to explore and discover things themselves. Constructivism as an educational philosophy has been introduced in computer science education (Ben-Ari, 2001; Hadjerrouit, 1998, 1999). Teaching methods in informatics that encourage students to construct their own knowledge are the following:

3. Solving software exercises with teacher assistance
6. Individual problem solving
8. Reading textbooks and user manuals
9. Using study material made online by the teacher

The advantage of these methods is that there is a high degree of activity where students must make their own discoveries through problem solving. This fosters creativity and reflection. It is also easier to add up to differentiated tasks. A disadvantage may be that students cannot get an understanding of the problem if they do not identify the correct answer, or that they find the answer that they think is correct but is not in a specific situation.

The sociocultural perspective, or similar designations such as situated learning, ties together the individual and the social context. It can be seen as a correction of constructivism. Whereas in the constructivist paradigm learning is assumed to occur as an individual learner interacts with study material, learning in the socio-cultural perspective is understood as a social process in which individuals interact with the social environment (Vygotsky, 1978). Learning occurs as learners construct their knowledge through discussion, dialogue, communication, collaboration, information sharing, and interaction with others. Furthermore, Vygotsky argued that the guidance given by more capable others, allows the learner to engage in levels of activity that could not be managed alone. Like constructivism, situated learning has been introduced in computer science education (Ben-Ari, 2004). Typical methods that foster sociocultural learning are as follows:

4. Project work over several days or weeks
5. Group work under the guidance of teacher
7. Homework assignments and ask friends, siblings, or parents for help
10. Present solutions to problems in front of the class

The advantage of such methods is that they foster dialogue, discussion, cooperation, and student participation. Knowledge is acquired through collaboration, and it is engaging if the topic is interesting. Disadvantages of group and project work may be timeliness and efficiency. It can be difficult to evaluate individual members, and it does not necessarily promote independent learning and being able to stand on its own.

Formal and Informal Methods

According to McDougall and Boyle (2004), teaching methods can be divided in formal and informal methods. Formal methods are defined as those that take place within the normal classroom interaction between students and teacher, on the one hand, and among students, on the other hand. Informal methods emphasize students' own desire and ability to figure out things on their own through interaction with other sources either in the classroom or outside. The following table shows the methods grouped according to whether they are formal or informal (Table 1). Methods 4 and 9, which are located in the middle colon, mean that they can relate to both categories.

FORMAL METHODS	BOTH FORMAL AND INFORMAL METHODS	INFORMAL METHODS
1. Teaching with black-board/overhead projector	4. Project work over several days or weeks	6. Individual problem solving
2. Demonstration of software tools	9. Using study material made online by the teacher	7. Homework assignments and ask friends, siblings or parents for help
3. Solving software exercises with teacher assistance		8. Reading textbooks and user manuals
5. Group work under the guidance of the teacher		10. Present solutions to problems in front of the class

Type of Interactions

When considering the effectiveness of methods, it may be important to look at the methods and the various interactions they bring about, that is whether the interaction is between student and teacher, between students, or between students and tools or other persons (McDougall & Boyle, 2004). The methods can be attributed to interactions are as follows (Table 2):

INTERACTION	TEACHING METHODS
Teacher – student	1. Teaching with blackboard/overhead projector 2. Demonstration of software tools
Student – student	4. Project work over several days or weeks 5. Group work under the guidance of the teacher 10. Present solutions to problems in front of the class
Student – textbooks	8. Reading textbooks and user manuals
Student – friends/ parents/ siblings	7. Homework assignments and ask friends, siblings or parents for help
Student – Internet	9. Using study material made online by the teacher
Student – software tools	3. Solving software exercises with teacher assistance 6. Individual problem solving

This survey questionnaire used in this work helps to see the interactions that take place in conjunction with the various methods. This must be seen also in connection with the categorization in formal and informal methods.

Learning Pyramid

Finally, teaching methods can be placed within the Learning pyramid, which shows an overview of different teaching methods, and ranks them according to their learning effect, for example reading and listening to a lecture as the least effective method to learn from (Figure 1). Sousa (2001) and Danielson (2002) provide information on the Learning pyramid, which shows that students retain knowledge best by “teaching others”. The Learning pyramid has become popular when using teaching methods, despite the fact that the original data used in its conceptualization has been lost (Ibid). The pyramid shows the percentage of what people learn best from, but it is difficult to assess how much or how little students learn from lectures or other methods in authentic classroom environments. Research and larger investigations are still needed to determine the learning effect of different methods and the extent to which the percentages are right. Hence,

teachers need to be critical when using the pyramid as a model for learning (Ibid). It is not recommended to use the pyramid as a blueprint of how this all fits together and what methods are most effective. Despite this criticism, or rather with it in mind, it is still useful to place the 10 teaching methods listed above into the Learning pyramid. The linking of the 10 methods in the pyramid provides a perspective to understand what makes good and poor teaching. The pyramid needs to be slightly modified to insert other methods and new scales - method 3 and 6 -, since these do not fit into the existing ones. The Learning pyramid is over 30 years old, and as result, it cannot capture newer teaching methods.

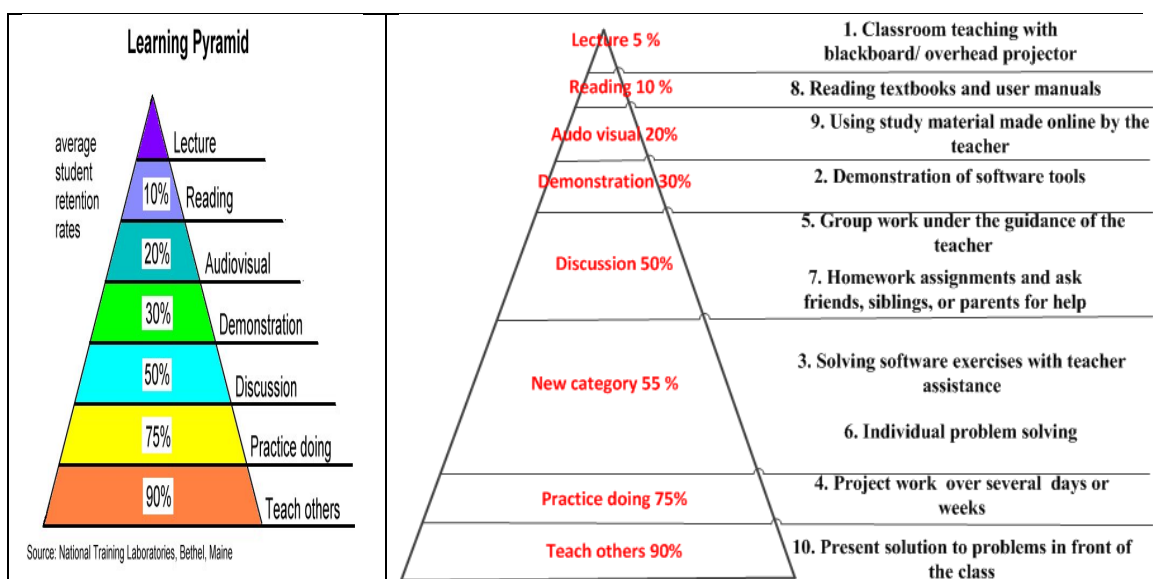


Figure 1: Learning pyramid

Research Design

Data Collection and Analysis Methods

This study was situated within teacher education at the university level in collaboration with two secondary schools, where teaching practice of six weeks was performed. The unit of study was the learning of informatics as a school subject using a number of teaching methods. The main goal of the study is firstly to analyze students' perceptions of the teaching methods being used during the teaching practice, and secondly, to identify successful teaching methods that may require attention in future experiments. The study does not aim at investigating student performances in relation to students' perceptions of teaching methods. Accordingly, particular attention was devoted to the following sources of information:

- Presentation and discussion of trainee teachers' work during their practice period of 6 weeks in upper secondary schools
- Analysis of trainee teachers' written reports they produced by analyzing students' perceptions of teaching methods they used during their teaching practice period of 6 weeks.
- Teacher's observations of trainee teachers' activities during their teaching practice in upper secondary schools.

Based on these information sources, the main data collection and analysis methods used for evaluating students' perceptions of teaching methods were a survey questionnaire and open-ended questions. The work also draws on trainee teachers' information exchange with the students and

their responses to open-ended questions. The survey was presented to the students at the end of the teaching practice period. It consisted of 10 questions in which students had to decide to what extent they agreed or disagreed with the statements and open-ended questions that students should answer in their own words. To measure the responses a 5-point Likert scale was used, where *Learned a lot* = 5, *Learned quite a lot* = 4, *Learned something* = 3, *Learned not much* = 2, *Learned nothing* = 1. To evaluate the results of the questionnaire, the mean was calculated, and a synthesis of the results was elaborated. The questionnaire was also organized so that students could describe in their own words what they liked, disliked or wanted changed with the teaching methods.

Participants

The work uses a convenience sampling (Bryman, 2012), one in which the researcher has directly access to the participants, all of whom are trainee teachers taking a full-time post-graduate certificate in education, with a specialization in informatics didactics, on the one hand, and school students from two different classes, on the other hand. Supervised practice is incorporated in the study with 12 weeks in all, divided into two 6-week periods. The first period is comprehensive practice, and the second period is in-depth practice. This study is related to the trainee teachers' in-depth practice in upper secondary schools. It involves trainee teachers' efforts to teach two different classes using several teaching methods. The teachers have basically the same competence in planning, carrying out, documenting, and assessment of training and learning in informatics. The first class consists of 25 school students. Most students know each other well, but one student has reading and writing difficulties. In this class, some students were more proficient than others in terms of knowledge and skills. The second class consisted of 15 students. The class seemed interested and motivated. Even though there were no particular challenges to classroom management, there was a certain knowledge level difference between the best and weakest students. Three students were proficient in creating Web sites and delivered solutions of very good quality, and went far beyond the tasks required. However, most of them were rather bound to follow guidelines and examples from textbooks to get their Web sites developed. Even though the students were from different classes, the teaching conditions were basically similar in terms of curriculum, computer infrastructure, students' knowledge level background, teachers' pedagogical experiences, and school environment as well.

Teaching Material

The teaching subject in both classes was basically database design using Dreamweaver. The trainee teacher from class 1 taught chapters from a database textbook with the following content: Working with MySQL, connecting Dreamweaver and MySQL, retrieving Data, and entering and updating data. In class 2, even though the students had a textbook for teaching databases, the teacher did not use it much, because it was not particularly well suited to the teaching material that was conveyed. Instead, the teacher made the most of the study material himself using different textbooks, some screenshots from the self-produced material, and a task sheet as well.

Results

First, the results achieved by means of the survey questionnaire are described. Then, a synthesis is presented. The tables show the number of students' responses to each category of the Likert scale from 5 to 1, including the mean (MEAN) and the rank (RANK) that is assigned to each method according to the mean. Students' responses to two open-ended are also presented and analyzed.

Class 1

Table 3 shows the results achieved for class 1 ($N=25$). The table shows the number of students' responses within the categories of the Likert scale for each method. The mean (MEAN) is also calculated, and a rank (RANK) is attributed to each method.

TEACHING METHODS	5	4	3	2	1	MEAN	RANK
1. Teaching with blackboard/overhead projector	4	11	8	2	0	3.68	6
2. Demonstration of software tools	4	12	7	2	0	3.72	4
3. Solving software exercises with teacher assistance	3	16	5	1	0	3.84	2
4. Project work over several days or weeks	10	10	5	0	0	4.28	1
5. Group work under the guidance of the teacher	5	10	7	2	1	3.64	7
6. Individual problem solving	7	7	8	3	0	3.72	4
7. Homework assignments and ask friends, siblings, or parents for help	2	6	10	7	0	3.12	10
8. Reading textbooks and user manuals	6	11	4	4	0	3.76	3
9. Using study material made online by the teacher	0	16	8	1	0	3.60	8
10. Present solutions to problems in front of the class	3	9	6	4	3	3.20	9

According to the results, *project work* is the teaching method that the students think they learned best from (MEAN=4.28, RANK=1). Hence, it appears that this method is well suited to the informatics subject, since it can help meet the demands of customized training and contribute to variation in education.

Then, most students think they are getting good outcomes from *solving software exercises with teacher assistance* (MEAN= 3.84, RANK=2). It seems that this method is well suited to the objectives of the curriculum, which encourages students to explore on their own, take responsibility for their own learning process, repeat and do exercises on their own pace.

Likewise, students think that they learned from *reading textbooks and user manuals* (MEAN=3.76, RANK=3). Among individual working methods, this is what the students think they learn best.

Similarly, most students believe that they have reasonably good outcomes from *individual problem solving* (MEAN=3.72, RANK=4). Only 3 students believed they do not learn much.

The results also show that only 2 students believed that they learned little from *demonstration of software tools*. The overwhelming majority, 16 of 25 students, that is 64% of all students, think they learned a lot or quite a lot from this type of teaching method (MEAN=3.72, RANK=4). This is clearly a good result, even though this method does not stimulate reflection and discussion. It is about to mimic and repeat teacher's actions when demonstrating the functionality of software tools. Nevertheless, it seems that students liked this method, perhaps because it is suited to informatics education as a practical subject.

Furthermore, most students were quite positive about *teaching with blackboard/overhead projector*, and only 2 had a negative perception (MEAN=3.68, RANK=6). Even though the mean is 3.68, which is a good result, on the one hand, the method is classified as number 6, which is a negative result compared to other methods, on the other hand. Hence, it seems that other teaching methods seem to give better results than teaching using the blackboard/overhead projector. As a consequence, it appears that this type of teaching is less suitable than other methods to promote learning. Nevertheless, it can be concluded from the score achieved (MEAN= 3.68, RANK=6) that this type of teaching can still be used in the teaching of informatics, even if the students believed that they learn better from other methods.

Group work under the guidance of the teacher is a teaching method that is often used in school, and which, like project work, also can help to achieve curriculum goals such as the ability to collaborate and develop social skills. However, students' responses are in contrast to the view that group work is a teaching method that is often popular among students (MEAN=3.64, RANK=7). Even though group work is a teaching method that students can advantage of, the rank indicates that the students prefer other methods than group work to achieve learning outcomes, which is a quite surprising result.

Students also seem to learn from the *study material made online by the teacher* (MEAN=3.6, RANK=8). In contrast, students believed that they actually learned more from textbooks than from the material that the teacher made online.

In contrast, students think they have bad or less effective learning outcomes from *presenting solutions to problems in front of the class* (MEAN=3.2, RANK=9). Aside from homework, this is the way of working students believe they learned less from. This is somewhat surprising, because the method is considered as one of the best ways of teaching according to the Learning pyramid.

Finally, students believed that *homework, ask friends, siblings, or parents for help* results in poor learning outcomes (MEAN=3.12, RANK=10). That is the question in the survey the students responded the most negatively.

Class 2

Table 4 shows the results achieved for class 2 (N=15)

Table 4: Number of students' responses on the basis of a Likert scale from 5 to 1							
TEACHING METHODS	5	4	3	2	1	MEAN	RANK
1. Teaching with blackboard/overhead projector	2	8	5	0	0	3.82	3/4
2. Demonstration of software tools	4	8	2	1	0	4,00	2
3. Solving software exercises with teacher assistance	4	8	3	0	0	4.09	1
4. Project work over several days or weeks	1	9	3	1	1	3.82	3/4
5. Group work under the guidance of the teacher	0	6	7	2	0	3.27	7
6. Individual problem solving	2	7	3	2	1	3.45	6
7. Homework assignments and ask friends, siblings, or parents for help	0	2	2	4	7	1.91	10
8. Reading textbooks and user manuals	2	8	2	2	1	3.55	5
9. Using study material made online by the teacher	0	5	4	4	2	2.82	9
10. Present solutions to problems in front of the class	0	6	6	1	2	3.09	8

According to the results in this class, the best teaching method is "*Solving software exercises with teacher assistance*", which obtained a high average score (MEAN=4.09, RANK=1), followed by "*Demonstration of software tools*", which got a MEAN of 4.00 (RANK=2). Students believed that they learned quite a lot from those methods. According to the students, these methods are well suited to informatics because there is much use of software programs in teaching.

Then, the methods that obtained a score between 3.50 and 4.00 are "*Teaching with board/overhead projector*" (MEAN=3.82, RANK=3), "*Project work for several days or weeks*" (MEAN=3.82, RANK=3), and "*Reading textbooks and user manuals*" (MEAN=3.55, RANK=5).

Using the blackboard and overhead projector naturally relate to demonstration of software tools, so it makes sense that this also obtained a high score. Many students believed that project work has a good learning effect, but the students' comments show that some are not so happy with presenting their work, which often follows from project work. Ultimately, the learning outcome is a function of how well students feel confident in front of others. The learning effect can still be high.

Reading manuals and textbooks obtained also a quite good result (MEAN=3.55, RANK=5). A possible explanation is that much of the teaching was based on the textbook (or written study material), so that students were accustomed to relate to and be based on it.

Furthermore, four methods obtained a score between 2.82 and 3.45, "*Individual problem solving*" (MEAN=3.45, RANK=6), "*Group work under the guidance of the teacher*" (MEAN=3.27, RANK=7), "*Present solutions to problems in front of the class*" (MEAN=3.09, RANK=8), and "*Reading study material made online by the teacher*" (MEAN=2.82, RANK=9).

Students considered that they learned something from these methods. Most of them were quite dependent on help from the teacher in their attempt to solve problems. The fact that solving problems on their own got a good result (MEAN=3.45) indicates that they liked doing activities on the computer. The relatively lower score for presentations in front of the class was not unexpected (MEAN=3.09, RANK=8), since there were some students who do not like this method. Concerning the method "*Reading study material made online by the teacher*", students probably perceived it as a primarily theoretical rather than practical activity (MEAN=2.82, RANK=9).

Finally, "*Homework and ask friends, siblings, or parents for help*" got a low score (MEAN=1.91, RANK=10), which means that the students did not learn much from this method. A possible explanation is that the informatics subject requires a high degree of expertise so that only few students may ask their siblings, friends or parents for help. Another explanation is that homework assignments are negatively perceived, and as a result, most students will give a low score to a negatively charged term.

Responses to Open-Ended Questions

The students were asked to respond to two open-ended questions:

1. Describe in your own words what you like and dislike about the teaching methods.
2. Describe in your own words what should be improved with the teaching of informatics.

Class 1

The most representative answers to the first question were: "*Better to work on your own*"; "*Okay to be working with websites*"; "*Liked variation*"; "*Liked project work, learned a lot, do not like theory and reading*"; "*Don't like to read, the teacher should explain*"; "*Liked practical work and exercises, dislike theory*"; "*Like project, then we must do something and find out for ourselves*"; "*Don't like old book, like independent work*"; "*Like variation and blackboard, software demonstrations and autonomous work*"; "*Liked Power Point Presentations, very instruc-*

tive"; *"Enjoyed freedom, work on own pace"*; *"liked project and independent work"*; *"a little too much theory, but good"*; *"Much free software"*; *"Liked review of study material, take notes, work with tasks"*.

The most representative answers to second question were: *"More practice"*; *"Better textbook with better examples"*, *"Disconnected from theory, not so much homework"*; *"Teacher should help more"*; *"More projects"*; *"Teachers must have better skills"*; *"More challenging tasks"*; *"More transparency and structure"*; *"More excitement and independent experimentation"*; *"More practical work, more teacher demonstrations, and new IT-book"*; *"New IT-book, the opportunity for independent work while the rest follow the teacher, programming"*; *"New book, and more free projects"*; *"Less power point, more homework and discussion of homework"*; *"Forced to do more, more homework, easy to relax"*; *"Several teachers / smaller classes"*.

Class 2

The most representative answers to the first question were: *"I like that we get projects to create, but I dislike that we suddenly jump in the project without a little preparation first"*; *"I like best when we are doing projects and when we do things together. I like mostly projects related to real-life software production"*; *"I like best when we have projects or tasks (...)"*; *"I do not like group work because it ends up presenting it. I like that we do different tasks, (...). There I learn a lot from"*; *"I liked to work with tasks from the book. (...)"*; *"Disliked that we could be stuck for quite a while and lose concentration. Like to learn new things as I see I'm going to get great use for if I were to end in the computer industry"*; *"Hands-on instruction where student can imitate what the teacher shows. It can sometimes be problematic to work alone with textbook, but it depends on the study material"*.

The most representative answers to the second question were: *"Go through the study material thoroughly before trying"*; *"Fine. The only thing that possibly could have helped had been several teachers to help"*; *"Let those who will work with tasks, and let those who will follow when something is happening on the blackboard"*; *"Slightly more diverse teaching than to follow the teacher on the blackboard or group project"*; *"Seems we should have projects that we are divided into pairs or groups. Preferably choose groups themselves"*.

Summarizing, the students from both classes seem to like project work, independent and autonomous activities, and practical tasks as well, and to a certain degree teacher-directed instruction. They disliked theoretical knowledge without practical applications, and old-fashioned informatics textbooks. Hence, the students' opinions on those issues seem to be in line with the responses achieved by means of the survey questionnaire.

Synthesis of results

A synthesis of the results based on the survey questionnaire consists of calculating the mean of the mean achieved for both classes. A synthesis of the results is possible because the teaching conditions of both classes are basically the same in terms of classroom culture, infrastructure, teachers' and students' knowledge background, etc. The synthesis gives the following results (Table 5). As the table shows the best (or most effective) teaching method was *"Project work"*, followed by *"Solving software exercises with teacher assistance"* (no. 2), *"Demonstration of software tools"* (no. 3), and *"Teaching with blackboard/overhead projector"* (no. 4). Then, the method that was ranged as number 5 is *"Reading textbooks and user manuals"*, then *"Individual problem solving"* (no. 6), and *"Group work under the guidance of the teacher"* (no. 7). Finally, the teaching methods that were considered as the less effective in terms of learning outcomes were: *"Using study material made online by the teacher"* (no. 8), *"Present solutions to problems in front of the class"* (no. 9), *"Homework assignments and ask friends, siblings, or parents for help"* (no. 10).

Table 5: Synthesis of results obtained for class 1 and 2

	Class 1: MEAN	Class 1: RANK	Class 2: MEAN	Class 2: RANK	Class 1 & 2: MEAN	Class 1 & 2: RANK
1. Teaching with black-board/overhead projector	3.82	3/4	3.68	6	3.75	4
2. Demonstration of software tools	4,00	2	3.72	4	3.86	3
3. Solving software exercises with teacher assistance	4.09	1	3.84	2	3.965	2
4. Project work over several days or weeks	3.82	3	4.28	1	4.05	1
5. Group work under the guidance of the teacher	3.27	7	3.64	7	3.455	7
6. Individual problem solving	3.45	6	3.72	4/5	3.585	6
7. Homework assignments, and ask friends, siblings, or parents for help	1.91	10	3.12	10	2.515	10
8. Reading textbooks and user manuals	3.55	5	3.76	3	3.655	5
9. Using study material made online by the teacher	2.82	9	3.6	8	3.21	8
10. Present solutions to problems in front of the class	3.09	8	3.2	9	3.145	9

Discussion

Implications for teaching informatics can be drawn on the basis of students' evaluations and their own assessments in relation to the theoretical framework.

Project Work

Project work over several days or weeks seems to be a highly relevant method to students. Thirty students agreed that they like this method, in line with the Learning pyramid, which puts this method into the category of practice with 75% learning effect. Hence, this method can be considered as the most important method to teach the informatics subject when it comes to learning benefits. Project work promotes social-cultural learning, interaction, and collaboration, as well as formal and informal ways of learning. Interaction is primarily between small groups of students collaborating and sharing knowledge to build software applications. Very often, project work is following by presenting and discussing the final product with other students in front of the class. However, there will always be some students who are not so confident in the presentation part of the project as the results clearly reveal (see results achieved for method 10). Nevertheless, project work is important to be prepared for a workplace where team work and group dynamics are often crucial parts of the work content. Trainee teachers also reported that good motivation is a very important starting point for project work as a teaching method otherwise it will not work well.

Solving Software Exercises with Teacher Assistance

Besides project work, it appears that students seem to learn a lot from this method. The number of students who liked this method (31) is even higher the one achieved for project work. This method supports constructivist learning, and it is a formal method that takes place in classroom. The teacher may play an important role in communicating information and providing help in solving exercises using software tools, for example creating Web pages and other software applications. The students believed that the method will be an important part of their future professional career as software engineers. In addition, solving software exercises with teacher assistance is in line with the spirit of the subject and the National Curriculum that views informatics as a practical subject.

Demonstration of Software Tools

Students considered demonstration of software tools, which instructs students explicitly in how to use a software tool, as very important. Twenty eight students were satisfied with this method. This is a behaviorist and formal method in which the interaction is mainly between teacher and students. Demonstrations are not considered as highly effective in terms of learning benefits according to the Learning pyramid (30% learning effect), but when demonstrations are linked to learning software functionalities, they appear to be highly relevant to the students, as training teachers clearly indicated in their reports. Training teachers think that students at the secondary school level are probably pretty addicted to be shown how things should be done through demonstrations via projector that students can imitate. However, demonstrations of software tools are highly dependent on competent teachers who are able to provide students with prompt and adequate knowledge on how to use the software. Otherwise students will probably remain passive and lose their motivation.

Teaching with Blackboard/Overhead Projector

Direct teaching via the blackboard, overhead projector or Power Point, which provides students with explicit information on software tools, theoretical and practical knowledge on programming or similar topics, seems to be relatively important, in stark contrast to the Learning pyramid, which places this method as the lowest in terms of learning effect. Twenty five students agreed that they like this method, even though this teaching method is a behavioristic and formal one, and teacher-direct as well. It is just highly dependent on the teacher and the quality of information provided, as the training teachers clearly reported.

Reading Textbooks and User Manuals

Reading user manuals and textbooks seem to be still important to the students, since 27 students were satisfied with this method. This is an informal method that supports students' interactions with textbooks, and constructivist learning. In contrast to the Learning pyramid, which places this method as one of the lowest in terms of learning effect, the students did not reject it. However, as highlighted by training teachers and students' responses to open-ended questions, the learning effect of this method depends on the quality of the textbooks provided and other written material made available.

Individual Problem Solving

This method refers to students who are supposed to learn entirely by themselves, that is to say those preferring individual learning. Twenty four students liked this informal method that supports constructivist learning. However, it seems that a pedagogy that promotes constructivist learning does not automatically work by itself, without the help or guidance received from a

teacher or more competent peer. Indeed, students still need help from the teacher because of the complexity of the software tool being used, or programming tasks. This is probably the reason why this method was not prioritized by the students.

Group Work under the Guidance of the Teacher

According to the students, group work, even under the guidance of the teacher, does not seem to have the same learning effect in comparison with similar methods, such as project work. Twenty one students agreed that they like this method. This result is quite surprising, because group work is supposed to have a learning effect of 50%. Work group is a formal method, and takes place within the normal conditions of a classroom. Moreover, it is supposed to foster student-student interactions and socio-cultural learning similar to project work. However, training teachers reported that unless group work is linked to project work and software development, it will not have the desired learning effect. If group work focuses more on discussing theoretical aspects of informatics than solving practical exercises, it will not attract and motivate students.

Using Study Material Made Online by the Teacher

Although 21 students liked this method, they did not consider it as important to their learning. The reason is that informatics at the secondary level is a practical subject, so it is probably not very appropriate for the teacher to put a lot of theoretical material on the internet. Unless this method is very directly aimed at particular tasks to be solved, the course web site will quickly become a list of online documents with little or without leaning effect for the student.

Present Solutions to Problems in Front of the Class

Eighteen students seem to like this method. However, students do not seem to like presenting solutions to problems to the whole class, even though this method is the most effective one from the point of view of the Learning pyramid in terms of learning benefits (90% learning effect). This method is also supposed to foster informal and socio-cultural learning. Nevertheless, this method is not the students' preferred one, on the contrary. According to the trainee teachers, the most important reason why they did not like this method is that it requires confidence and trust to be able to present solutions in front of others.

Homework Assignments and Ask Friends, Siblings, or Parents for Help

Finally, only 10 students were globally satisfied with this method. Like presenting solutions for the class, this informal method is also supposed to foster socio-cultural learning. However, it is unrealistic for most students to expect that they can get help at home with homework or assignments by friends, siblings, or parents. Trainee teachers indicated that the informatics subject uses specialized applications and is based on relatively complex logic. As a result, "ordinary people" who are not employed in software-related jobs have little chance to help students. Moreover, even software people must have been involved in the specific application to be of greater help.

Related Work

There are very few similar research studies on teaching methods in school informatics. A recent international study in several countries (Schulte et al, 2012), used 13 methods or approaches for teaching computer science at school. The methods can be rated on three different levels (Ibid, pp. 32). Four methods have been considered as important (first level):

1. Classroom based teaching
2. Using standard applications like text processing

3. Students work individually or in small groups on small tasks at the computer and
4. Programming
5. Projects

Teaching methods of moderate importance (second level) are:

6. Using editors
7. Students work individually on projects
8. Students work in small groups on projects discussions
9. Discussion
10. Lectures, and
11. Using integrated development environments (IDE)

Two methods are considered as unimportant:

12. Reading and role plays

As it can be seen, this study used other teaching methods, or similar methods with different designations. Even if there are some similarities, it is difficult to compare this work with this study for many reasons. The differences may be explained by different cultural traditions and approaches to informatics. For example, in some countries algorithmic thinking is more important than practical applications. As a result, teaching methods are rated differently depending on the topics being taught.

Implications for Teaching School Informatics

Summarizing, it appears from the results and discussion that the teaching of informatics at the secondary level should support all types of learning: social-cultural, constructivist, and behaviorist (or teacher-directed). First of all, project work, which combines social-cultural learning, formal and informal learning, as well as student collaborations, seems to be the best method to teach school informatics. However, the emphasis on the benefits of constructivist pedagogy, that is individual learning, is not fully supported by this work. On the contrary, the results reveal the importance of teacher-directed instruction that engages students in meaningful learning, as the research literature clearly reveals (McGhie-Richmond, Underwood, & Jordan, 2007). Indeed, direct teaching should not be underestimated when it comes to instruct students explicitly in what to do to solve exercises or how to use software tools. In addition, teaching informatics should be supported by high-quality textbooks and motivating exercises. Furthermore, demonstrations of software functionalities used to create meaningful applications are still very important in classroom settings. Clearly, students who spend much of their time doing exercises, repeating their lessons and being instructed by teachers or working under the direct supervision of teachers, make better progress than those who are expected to learn entirely by themselves (individual learning). However, teaching methods in themselves not have the desired learning effect without qualified teachers in terms of pedagogical strategies and technical knowledge.

Conclusions

This work is a contribution to the pedagogy of informatics as a subject in secondary schools. It is a study in which training teachers used 10 teaching methods in 2 different secondary schools were studied. The number of participants and the sample size (2 trainee teachers, 40 school students from two classes) may not be sufficient to adequately support the generalization of the results. Hence, new cycles of experimentations and evaluations of the teaching methods in future studies are warranted to generalize the findings of the present work, using both quantitative and qualitative methods to ensure more validity and reliability. However, even if it is impossible to draw any general conclusions from the teaching experiments that were performed in 2012, it can be ascertained that the training teachers from two different classes made a real attempt to assess

teaching methods in their respective classrooms. Their work can be considered as a real contribution to the knowledge base of school informatics. But still, there is a need for more training in using various and innovative teaching methods, application of learning theories and didactics concepts, acquisition of theoretical knowledge in conjunction with good application examples in classroom settings. Hence, a lot of work remains to do, and pedagogical changes and innovations in the informatics subject in secondary schools can be achieved only through the iterative and continuous cycle of experimentations and evaluations in varied school contexts.

References

- Apiola, M., & Tedre, M. (2012). New perspectives on the pedagogy of programming in a developing country context. *Computer Science Education*, 22(3), 285-313.
- Avidov-Ungar, O., & Iluz, I. E. (2014). Levels of ICT integration among teacher educators in a teacher education academic college. *Interdisciplinary Journal of E-Learning and Learning Objects*, 10, 195-216. Retrieved January, 2015 from <http://www.ijello.org/Volume10/IJELLOv10p195-216Avidov0892.pdf>
- Armoni, M. (2011). Looking at secondary teacher preparation through the lens of computer science. *ACM Transactions on Computing Education*, 11(4), Article 23, 23-1 – 23-38.
- Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45-73.
- Ben-Ari, M. (2004). Situated learning in computer science education. *Computer Science Education*, 14(2), 85-100.
- Bennedsen, J., & Caspersen, M. E. (2012). Persistence of elementary programming skills. *Computer Science Education*, 22(2), 81-107.
- Brodahl, C., Fagernes, M., & Hadjerrouit, S. (2007). Applying and evaluating understanding-oriented ICT user training in upper secondary education. *Issues in Informing Science and Information Technologies*, 4, 473-490. Retrieved June 12, 2014 from <http://proceedings.informingscience.org/InSITE2007/IISITv4p473-490Brod287.pdf>
- Bryman, A. (2012). *Social research methods* (2nd ed.). Oxford University Press.
- Cápay, M, Magdín, M., & Tomanová, J. (2012). Education of primary and secondary school teachers in informatics supported by digital technologies. DIVAI 2012 - 9th International Scientific Conference on Distance Learning in Applied Informatics, 53-61.
- Dagdilelis, V., Satratzemi, M., & Evangelidis, G. (2004). Introducing secondary education students to algorithms and programming. *Education and Information Technologies*, 9(2), 159-173.
- Dagdilelis, V., & Xinogalos, S. (2012). Preparing teachers for teaching informatics: Theoretical considerations and practical implications. *WiPSCE '12*, November 8-9, 2012, Hamburg, Germany, 78-81.
- Driscoll, M. P. (2000). *Psychology of learning for instruction* (2nd ed.). Boston: Allyn and Bacon.
- Danielson C. (2002). *Enhancing student achievement: A framework for school improvement*. Alexandria: Virginia: Association for Supervision and Curriculum Development.
- Duffy, T. M., Lowyck, J., & Jonassen, D. H. (1993). *Designing environments for constructive learning*. Berlin: Springer-Verlag.
- Fee, S. B., & Holland-Minkley, A. M. (2010). Teaching computer science through problems, not solutions. *Computer Science Education*, 20(2), 129-144.
- Hadjerrouit, S. (1998). A constructivist framework for integrating the Java paradigm into the undergraduate curriculum. *ACM SIGCSE Bulletin*, 30(3), 105-107.
- Hadjerrouit, S. (1999). A constructivist approach to object-oriented design and programming. *ACM SIGCSE Bulletin*, 31(3), 171-174.

Exploring the Effect of Teaching Methods

- Hadjerrouit, S. (2008). Using a learner-centered approach to teach ICT in secondary schools: An exploratory study. *Issues in Informing Science and Information Technology*, 5, 233-259. Retrieved May 24, 2014 from <http://proceedings.informingscience.org/InSITE2008/IISITv5p233-259Hadj424.pdf>
- Hadjerrouit, S. (2009a). Teaching and learning school informatics: A concept-based pedagogical approach. *Informatics in Education*, 8(2), 227-250.
- Hadjerrouit, S. (2009b). Didactics of ICT in secondary education: Conceptual issues and practical perspectives. *Issues in Informing Science and Information Technology*, 6, 153-178. Retrieved May 24, 2014 from <http://iisit.org/Vol6/IISITv6p153-178Hadjerrouit605.pdf>
- Hammond, M. (2004). The peculiarities of teaching information and communication technology as a subject: A study of trainee and new ICT teachers in secondary schools. *Technology, Pedagogy and Education*, 13(1), 29-42.
- Hromkovič, J., & Steffen, B. (2011). Why teaching informatics in schools is as important as teaching mathematics and natural sciences. In I. , & R.T. Mittermeir (Eds.). *ISSEP 2011, LNCS 7013*, 21–30. Springer-Verlag: Berlin.
- Hubwieser, P. (2004). *Didactics of informatics: Basics, concepts, examples* [in German. Original title: Didaktik der Informatik: Grundlagen, Konzepte, Beispiele]. Springer Verlag: Berlin.
- JIE & ACM WGIE. (2013). *Informatics education: Europe cannot afford to miss the boat*. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education, April 2013.
- Imsen, G. (2009). *The teacher's world* [In Norwegian. Original title: Lærerens verden]. Oslo: Universitetsforlaget.
- Kolikant, Y. B-D. (2010). Innovative teaching in computer science: What does it mean and why do we need it? *Computer Science Education*, 20(2), 73-78.
- Kumar, R. (2006). Informatics curricula and teaching methods. In: L. Cassel and R. A. Reis (Eds.) (2003) Boston: Kluwer Academic Publishers. Book review. *Educ Inf Technol*, 11, 93–97.
- Learning pyramid. (n.d.) National Training Laboratories. Retrieved February 12, 2014 from <http://www.washingtonpost.com/blogs/answer-sheet/files/2013/02/pyramid.png>
- McDougall, A., & Boyle, M. (2004). Students' strategies for learning computer programming: Implications for pedagogy in informatics. *Education and Information Technologies*, 9(2), 109-116.
- McGhie-Richmond, D., Underwood, C., & Jordan, A. (2007). Developing effective instructional strategies for teaching in inclusive classrooms. *Exceptionality Education Canada*, 17(1), 27-52.
- McGonigal, K. (2005). Teaching for transformation: From learning theory to teaching strategies. *Newsletter*, 14 (2).
- Minaidi, A. & Hlapanis, G. H. (2005). Pedagogical obstacles in teacher training in information and communication technology. *Technology, Pedagogy and Education*, 14(2), 241-254.
- Ogden, T. (2004). *Quality school* [In Norwegian. Original title: Kvalitetsskolen]. Oslo: Gyldendal Akademisk.
- Piaget, J. (1971). *Genetic epistemology*. New York: W.W. Norton.
- Repstad, K., & Tallaksen I. M. (2006). *Variert undervisning – mer læring*. Bergen: Fagbokforlaget.
- Schubert, S., & Schwill, A. (2004). *Didactics of informatics* [In German. Original title: Didaktik der Informatik]. Berlin: Spektrum Akademischer Verlag.
- Schulte, C. (2002). Towards a pedagogical framework for teaching programming and object-oriented modeling in secondary education. *Proceedings of SECIII 2002*, Dortmund, July, 22-26.
- Schulte, C., et al (2012). Computer science at school / CS teacher education – Koli Working-group report on CS at School. *Proceedings of Koli Calling '12*, November 15-18, 2012, Tahko, Finland, 29-38.

- Sousa D. A., (2001). *How the brain learns: A classroom teacher's guide* (2nd Ed.). Thousand Oaks, CA: Corwin Press.
- Steffe, L. P., & Gale, J. (Eds.). (1995). *Constructivism in education*. New Jersey: Lawrence Erlbaum Associates.
- Taber, K. S. (2011). Constructivism as education theory: Contingency learning, and optimally guided instruction. In J. Hassaskhah (2011). *Educational Theory*, 39-61. Nova Science Publishers Inc.
- Vygotsky, L.S. (1978). *Mind and society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.
- Webb, M. E. (2002). Pedagogical reasoning: Issues and solutions for the teaching and learning of ICT in secondary schools. *Education and Information Technologies*, 7(3), 237-255.

Biography



Said Hadjerrouit is Professor of Informatics Education and Computer Science at the University of Agder. He holds a Ph.D. (1992) in Medical Expert Systems and Artificial Intelligence, and a Master (1985) in Software Engineering from the Technical University of Berlin (Germany). He has been working at the University of Agder since 1991. Hadjerrouit has teaching experience in informatics and society, philosophical and ethical issues of computing, object-oriented programming, Web engineering, software development, databases, didactics of informatics, ICT in mathematics education, and ICT-based learning.

His current research focuses on didactics of informatics, digital tools in mathematics education, e-Learning, technology-enhanced learning, Web-based learning resources, Web 2.0 technology, and social software. Hadjerrouit has published more than 130 articles in international journals and conferences proceedings.