S. N. Pozdniakov, V. Dagienė (eds.), *Informatics in Schools. New Ideas in School Informatics*, Lecture Notes in Computer Science 11913
https://doi.org/10.1007/978-3-030-33759-9_22

Computer Science Problem Solving in the Escape Game "Room-X"

Alexander Hacke[1] ✉

(1) Didaktik der Informatik, University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Germany

✉ **Alexander Hacke**
Email: ahacke@uni-potsdam.de

Abstract
Problem solving is a key element of computer science. It is also a research topic within computer science education examining topics like processes, tasks and attitudes with regard to computer scientific approaches and contents. Our computer science escape game "Room-X" offers learners an insight into computer science and enables them to practice problem solving in an attractive and motivating environment. From a research perspective, Room-X allows us to observe learners of computer science while solving problems and to analyze their strategies involved. Video analyses are used to analyze behavioral patterns and to draw conclusions in order to promote problem solving skills in computer science and to further develop Room-X.

Keywords Computer science problem solving – Escape game – Out-of-school learning experience

1 Introduction

Problem solving is a key element of computer science and forms links with each of its sub-disciplines. In order to obtain a solid foundation for computer science education, secondary school and university students are required to deal explicitly with problem solving. However, little research is available on this topic. Consequently, little is known about how computer science problem solving can be taught in a purposeful manner. Nevertheless, problem solving is an inherent part of the German educational standards for computer science [2] and, in the form of computational thinking, of the K-12 CS standards [8]. Since computer scientific problem solving in German secondary schools is often treated in a rather theoretical manner, the topic is of little interest to many students. Often it is even taught implicitly, which does not do justice to the importance of the topic. Our approach to motivate secondary school

students for it and to practice problem solving strategies is the computer science-based escape game Room-X. It was specifically designed with computer science problem solving in mind. Room-X serves as a showcase for computer science (CS) and shows that CS education can also be helpful in a playful environment. In the first part of this paper, there will be a theoretical account of CS problem solving. Afterwards, a video analysis currently being conducted within the framework of Room-X will be described with the aim of finding out which major hurdles play a role in CS problem solving. At a later stage, this allows conclusions to be drawn as to which areas should play a significant role in teaching CS problem solving.

2 Computer Science Problem Solving

In cognitive psychology, *problem solving is described as the attempt to move from an initial state past a barrier to a target state* [9]. Problem solving requires a number of cognitive abilities which, according to Bloom, can be classified into six categories, with the top three ("Analyzing", "Evaluating", "Creating") being higher-order thinking and presupposing the lower three ("Remembering", "Understanding", "Applying") [1]. The processing of simple tasks can usually be represented by the lower three categories. Hence, it is necessary to understand the task (Understanding), to retrieve appropriate information and procedures from the long-term memory (Remembering) and apply them in the given context (Applying). Problem solving also requires higher-order thinking skills. It is important to analyze the problem, distinguishing important from unimportant details and revealing hidden aspects (Analyzing). Based on the analysis, a target-oriented strategy must be generated using appropriate heuristics, which, depending on the case, links known elementary procedures with new contexts (Creating). During the problem-solving process, this strategy must be constantly monitored for effectiveness and, if necessary, reconsidered (Evaluating).

The difference between a task and a problem lies in the fact that tasks require "only the use of known means in a known way to achieve a clearly defined goal", thus requiring only reproductive thinking. Problems, however, can only be solved with productive thinking. So a new or at least a modified solution has to be devised [4]. Whether it is a task or a problem depends on prior knowledge and is therefore not a fixed property.

In order to be successful in problem solving, it is necessary to have confidence in one's own abilities. Also, the attitudes to the specific problem and towards problem solving in general are largely responsible for how effectively a problem solver can use the means at his or her disposal (cf. [12]). The term "problem" needs to be narrowed down in terms of problem solving. A psychology-based definition states that *a problem exists when, in a situation where a particular goal is to be achieved, an obstacle or barrier prevents it* [9]. Problems can be categorized in many ways. For example, they can be differentiated by how clearly a target state to be reached is defined, or by classifying them as so-called *simple* or *complex* problems. In the case of *complex* problems, the surrounding conditions change during the course of the solution attempt, which requires a continuous reassessment of the solution approach. In addition, a large number of variables come into play, with many of them being interdependent. *Complex* problems include situations like managing a business or managing a global crisis. *Simple* problems, on the other hand, have stable conditions and comparatively less relevant variables. However, other than the name suggests, they are not easy to solve, either. Problems within a computer scientific context usually fall into the category of *simple*

problems. This means that the general conditions do not change or change only slightly during the solution attempt and the number of variables to be considered is within manageable limits. Of course, embedded in a real-world situation, they can also be part of a *complex* problem. In the context here, however, the focus will be on *simple* problems, since otherwise it is no longer clear whether the problem is of computer scientific or other nature.

> **Definition:** A computer science problem exists when, in a situation with stable conditions, an obstacle or barrier prevents a particular goal requiring a computer science-based solution approach from being achieved.

Computer Science Methods. It now has to be clarified which methods are to be attributed to computer science and which are not. Various attempts have been made in the past to characterize the nature of CS, as demonstrated by Grillenberger [5]. Thus, the theoretical-argumentative and the empirical approach stand opposite to each other. Depending on the focus in terms of subject area and perspective, quite different models or catalogs arise as to what is attributed to computer science. Irrespective of the methodology, the computer science method used for problem solving should be found at least in one of the widely accepted approaches, be it in the Fundamental Ideas of computer science by Schwill [13] or in the Great Principles of Computing by Denning [3]. If, for example, one sticks to the theoretical-argumentative point of view of the catalog of Fundamental Ideas as a description of the essence of CS, then it must be possible to trace computer science problem solving back to at least one of these ideas. This means, for example, that the use of an algorithmic paradigm such as Divide and Conquer or the use of the tree representation can be counted among the computer science methods, since they can be found in the list of Fundamental Ideas.

The question then arises as to whether the problem at hand is of a computer scientific nature or only the problem-solving process chosen or whether both parts may be attributable to computer science. Similar to the approach by Humbert and Puhlmann subdividing computer science phenomena into three categories [6], problems can also be classified according to their relation to computer science:

1. *The problem is not of computer science nature.* Problems of a purely philosophical nature where a computer scientific approach is not appropriate or out of place.
2. *The problem is indirectly of a computer science nature.* Problems that have a real-world character but are inherently computer scientific and can therefore be solved by a CS problem-solving strategy.
3. *The problem is directly of a computer science nature.* Problems that require a problem-solving strategy with computer-scientific principles.

By classifying problems in this way, it becomes obvious that a problem of the third category like finding the closest pair of points that can be solved using the divide-and-conquer algorithm, can certainly also be part of category two, namely as a computer-scientific part of a real world problem. For example, a problem involving the distribution of tasks to employees or vehicle scheduling in suburban traffic systems is often an integer programming problem that can be solved with the branch-and-bound algorithm of computer science. The

problem space of such a problem then consists not only of the computer science problem, but also of the fact that the computer-scientific character must first be recognized. However, problems of category two may also be solved in a non-computer-scientific manner. For example, depending on the scenario, in the case of a distribution problem either CS optimization might be necessary or a simple random distribution might suffice. It also becomes apparent that the initial situation of the problem provides information about the likelihood of an involvement of computer science in the problem-solving process.

2.1 Problem Solving Versus Computational Thinking

One of the concepts that seem similar to computer science problem solving at first glance is *Computational Thinking (CT)*. In 2006, Jeannette Wing's term came into play to describe a particular way of thinking.

Wing characterizes CT as problem-solving, system design and the understanding of human behavior through the use of fundamental computer science concepts. It contains mental tools that reflect the breadth of computer science [14]. Since there is no precise definition of Computational Thinking, ISTE and CSTA have sought to narrow down what *CT* is, based on the feedback from people involved in computer science education. *CT* is characterized as a problem-solving process with at least the following properties:

- formulate problems in such a way that we can solve them with the help of computers.
- logical organization and analysis of data.
- Representation of data by means of abstraction (e.g. models, simulations).
- automation of problem solving through algorithmic thinking (sequence of ordered steps).
- identify, analyze and implement possible solutions with the goal of using the most efficient and effective combinations of steps and resources possible.
- generalization and transfer of the problem solving process to a multitude of other problems [7].

The list of properties of Computational Thinking is based on the assumption that a problem from another science should be solved by means of computer science and that this solution should be implemented and automated. CS problem solving plays a role in this process, of course. However, the focus of CS problem solving, as described in this document, is not on the implementation and automation of a solution, but on the previous step, i.e. thinking through CS problems and creating structured solutions. This process does not necessarily require the computer, nor does the automation of problem solving and the generalization and transfer of the problem-solving process take center stage.

---

3 Problem Solving in Escape Games

Escape rooms (also known as live escape games, exit rooms, and other similar terms) are a special kind of escape game in which players as a team are locked inside a room. With the help of clues and puzzles inside they try to escape the room in a limited time. In most cases, there is also a mission to fulfill, such as disarming a bomb, solving a criminal case or stealing an object. The topics for such games are extremely diverse and are based on exciting settings, e.g. chemical laboratories, prison tracts or agent offices. Escape games also provide incentives for educational contexts. For example, Nicholson [10] describes the benefits of

using such games in the classroom as a welcome change from working on the computer, the need for team collaboration, and motivational aspects as the basis for active learning and social constructivism. Escape games provide a great opportunity to train problem-solving skills. The concept of such games incorporates the essential features of a *simple* problem and thus makes the players problem solvers. Within a certain period of time, they have to move from an initial state (the room and the hints provided) to a destination state (usually: to leave the room). This is not possible without further ado because one or more obstacles (riddles, the door cannot be opened easily, etc.) are put in the way. Therefore, they must use heuristic procedures (for example, formation of sub goals, search space limitation, visualization), creatively plan a solution strategy and constantly check this strategy for meaningfulness during execution. In addition, escape games are well-suited as their playful adventure character helps to keep the motivation of the participants high and to mask any negative attitudes to problem solving that may exist. Possibly the contained problem solving will not even be perceived as such.

Group Effects. In addition to the usual hurdles of a problem solver, according to Rosenstiel [11] team interaction creates various additional obstacles. For example, the so-called group think may play a role. The opinion of the majority then becomes the binding factor and deviating ideas are suppressed. A further influence is possible through the effect that a very talkative person might have on the group. The latter is granted a higher influence on group decisions. He or she may therefore consciously or unconsciously lead the group, as long as he is not recognized as a "busybody". Furthermore, the decision-making quality does not increase proportionally with the size of the group. On the contrary, it even decreases with group sizes of ten or more participants due to constraints in communication and interaction. [11] Even though Room-X does not allow for more than six participants, the group effect may affect the result. There are also known negative group effects when the team cohesion is disturbed. The individual may then not exhaust his full potential but rather orient himself towards the maximum performance of the group. He or she does not want to be the only one, who works, if the others do not participate. Nobody really feels responsible and certain team members may choose not to work at all and still profit from the work of the others. However, the group also has positive effects. Usually there is communication, i.e. the observation of the processes is simplified, which is essential to gain knowledge about problem solving situations. In groups, the participants are usually less inhibited and motivation is higher. Also, the performance of the individual participants can be increased by the team feeling.

---

4 Room-X: An Escape Game for Computer Science Lessons

The escape game Room-X was set up at our university to give students in a limited time frame a motivating insight into various topics of computer science and to promote the institute of computer science. In this context, groups of students who would like to play the game come to visit us regularly. This gives us the opportunity to observe them while solving problems. Their mission is to spy on the tasks of the next computer science exam, which is stored on a password-protected tablet in the classroom of Mr. Schroeder. The exam must be photographed, otherwise the mission is not completely fulfilled and is considered as failed. The password can be found using the items in the room. In addition, the teacher has activated the alarm system of the classroom door. In order to escape unnoticed, the team must find out

the numerical code of the key vault containing the remote control of the alarm system. The team in Room-X is monitored throughout the session by a camera inside the room, so they can be helped if necessary by the game supervisor passing tips into the room. The game lasts 60 min. When the time expires, the alarm system triggers. Opening the door prematurely also triggers the alarm system and leads to disqualification and abortion of the mission. First, the team get all the information about the scenario, the processes and the rules of Room-X in a separate room. The use of the whiteboard and notepads and pens in the room is explicitly permitted. The teams are advised that the game supervisor can contact people in the room during the game. Then the team is led into Room-X, the timer is started, the door is closed and the alarm system is activated. By recognizing and solving characteristic CS tasks and skillfully combining the clues found, it is finally possible to unlock the tablet, open the key vault and deactivate the alarm system. The team members then have the opportunity to discuss their experiences with each other and to learn backgrounds and solutions to the individual puzzles in the room.

In order to prevent the dissemination of the solution for the room, only a rough description of the puzzles is given here:

- The tablet is secured with a password and has a sticky note on it with the words "PW: Holidays! (HexHex)" → What's that supposed to mean?
- On the wall is a piece of paper with a cryptic message: "zpcyidyqr rmbyw dccjq jgic y pmai gl kw qrmkyaf." → What does that mean?
- On the teacher's desk there is an SD card with the inscription *SECRET*. If you insert it into a digital camera lying around, you will see photos of different objects (for example, a huge device with the inscription *Z3* and a strangely colored map) → Are there any decisive hints in the pictures?

---

5 Room-X and Computer Science Problem Solving

Aiming at observing the students' approach to solving computer science problems in Room-X, it will first be examined to what extent the game demands or requires computer science problem solving. For this purpose, the room with its associated tasks and puzzles is analyzed below with reference to the definitions and thinking skills mentioned above. In a subsequent video analysis, the strategies of the participants are identified and analyzed with regard to the properties that lead to success to derive conclusions for fostering problem-solving strategies.

5.1 Description of Problem and Problem Solving in Room-X

The starting situation faced by the participants corresponds to a *simple* problem according to the above-mentioned problem definition, because the following characteristics can be found: At the beginning of the game the team is in the initial state, which consists of the room with its hidden clues and the hints given by the game supervisor. The group cannot easily proceed to the target state, because of various obstacles (door code, tablet password). The surrounding conditions do not change during the search in the problem space, if time pressure is disregarded. Also, the number of variables that must be handled in the course of the game is manageable and the inter-dependencies between them are low. Accordingly, the

problem is not part of the *complex* category. Examining the path through Room-X reveals that it includes many elements that only need lower-order thinking skills. That is, there are a number of *tasks* to be solved in the room. For the most part it is not possible to establish the connection between the tasks, or to recognize what the solution of a task might be good for in order to progress in the game. This has less to do with the complexity of the strategy to be found than with the fact that the proposed solution is in some places too artificial. Connections do not always follow a recognizable pattern and are thus not predictable. So, the way to the target state often requires brute force, teamwork and luck.

5.2 Computer Science in Room-X

The problem to be solved in Room-X is not computer scientific in itself. However, the path through the problem space contains a number of *tasks* of a computer science nature. The individual tasks on the topics of encryption, logic and automata theory require computer-scientific and general sub-strategies for problem solving, such as following a path, reproducing algorithms, representation and recognition of a model, verification and purposeful combination of results as well as continuous documentation. Room-X therefore does not contain an overarching CS problem-solving strategy, but sufficient sub-strategies in order to allow conclusions to be drawn as to how well the students are prepared for CS problem solving.

5.3 Applied Problem-Solving Strategies in Room-X

In a qualitative video analysis, we aim to examine whether the sub-strategies mentioned above can be observed and whether an influence on the success of problem solving can be derived. This raises the following questions for the video analysis:

RQ1. What typical behavioral patterns can be observed during a problem-solving process in Room-X?
RQ2. What impact do observed behavioral patterns have on success in problem solving in Room-X?

Regarding the questions, the following assumptions can be deduced based on the considerations and definitions in Sect. 2 as well as the properties of Room-X: It is assumed that the teams will search the objects in the room for clues of all kinds. The team will split up according to their preferences or prior knowledge according to the tasks. They will try to solve the individual tasks, use the whiteboard and notepads as a means of visualizing or representing the findings, communicate with each other and evaluate findings in the team. It is assumed that the following behavioral patterns lead to success: systematic search for clues, correct solution of the individual tasks, visualization and representation of the hints and results, involvement of all team members and evaluation and combination of hints and results.

5.4 Conducting the Video Analysis

For the video analysis, video material of 38 groups of five to six people each is available, which corresponds to about 200 participants. The material is high-definition video with sound from a surveillance camera on the ceiling of the room. This monitoring is usually used by the game supervisor to control the game. The video footage was examined for the

participants' success in problem solving. The following behaviors were isolated in advance and operationalized (deductive approach):

1. Correct solution of the individual tasks: One or more team members solve one of the tasks and find a correct solution.
2. Involvement of all team members: All participants are focused on the problem, that is, looking for clues, giving advice, helping others, solving tasks.
3. Systematic search for clues: The room is thoroughly examined for clues from one end to the other, ideally independently by several people.
4. Visualization and representation of hints and results: The board or a notepad is used to record intermediate results, hints, findings and questions as soon as they are available.
5. Evaluation and combination of hints and results with each other: results are mutually checked; they are related to each other verbally or in writing on the whiteboard.

Observed Behavioral Patterns. After qualitative evaluation of approx. 70% of the video material, tendencies regarding the first question can be identified. Every group begins by applying a brute-force heuristic strategy: all team members swarm out, scatter in the room, leaf through books, etc. This corresponds to the expected search of the objects in the room for clues. Speed and thoroughness of this process vary greatly. The fast teams need about 13 min, the slow teams up to 30 min (average: 19 min). During the search, various tasks are discovered and usually immediately attempted to be solved. Tasks that seem too difficult for one person are left behind or someone else is consulted. For example, this happens when a conversion to another number system must be carried out. Finding the correct solutions to the CS tasks varies from eight to 31 min (average: 18 min). There are also a number of teams that cannot solve all the tasks.

The whiteboard is used to document individual results. However, the documentation of found hints is often sparse and visualization is rare. Every now and then hints get lost in the communication process of the team members and then have to be rediscovered.

In addition, behavioral patterns become visible that suggest that individual team members are demotivated, which means that sometimes there are participants who often look out of the window or stand around indifferently.

Promising Behavioral Patterns for Solving Problems. The assumptions regarding promising behavioral patterns can also be largely confirmed by the video material: Teams that use the whiteboard in a more structured fashion are usually more successful. For example, dashes for the number of digits of the password were written on the board, which can be seen as a meaningful representation of a sub goal. Teams without recognizable structured sketches on the whiteboard could still be successful, provided they still wrote down a lot in their notepads or kept the results circulating verbally. There was little use of computer scientific graphical aids, e.g. trees or graphs, but that was to be expected. More successful teams also have at least two structured task solvers. These do not jump from task to task but remain focused on one task and use paper and pencil.

Group behavior. The analysis of the video material also revealed that teams work very

differently. There are teams whose members communicate a lot with each other, those in which the members seem to be relatively indifferent, and sometimes even teams that work destructively. Certainly, there are group effects involved as described in Sect. 3. The following role types were identified during the analysis: *the leader* (someone who distributes tasks to others), *the coordinator* (someone who writes down intermediate results and ensures that information is passed on, but does not distribute tasks), *the loner* (someone who works alone on tasks for a longer period of time), *the inactive* (someone who is idle for a longer period of time), *the follower* (someone who essentially only follows and watches other team members), *the worker* (someone who gets to work and tries to solve tasks) and *the supporter* (someone who, for example, assists a worker and helps him to solve tasks). These types are not disjoint and there are also role changes. Also, not all types are present in every team. It seems to be more effective to have a coordinator rather than a leader on the team. In fact, at least five teams with a leader present could not finish the game successfully. On the other hand, the fastest team is one with two coordinators. It can also be seen that unsuccessful teams usually have one to five followers or inactive members. Successful teams consist exclusively of coordinators, supporters and workers. It can already be seen that behavior suggesting convergent or divergent thinking is not tied to a specific role. The evaluation must still show whether a certain way of thinking is more likely to belong to a certain role type or not.

Aspects Hampering Success. The game contains some wrong tracks, which all in all lead quite effectively to the group losing sight of the essentials and thus increase the problem space. In addition to these deliberate measures, there were several other hurdles for the participants. An often-recurring phenomenon is the lack of meaningful documentation of intermediate results and the lack of communication within the team. As a result, interim results are lost and it can happen that the notes on the whiteboard, which are mostly lacking any explanation and are usually written by different team members, cause additional confusion. In this regard, there is a tendency not to pursue all hints consistently. More than half of the teams leave at least one discovered item unused for minutes or do not use it thoroughly. Another point is the lack of evaluation of results. Results that appear cryptic are accepted without cross-checking and lead to avoidable errors for at least six teams. Furthermore, the CS tasks lead to failure for approx. 20% of the teams. The reasons are manifold. There are at least two teams that are unwilling to do the tasks. Several teams lack the basic knowledge, which is why they need a very long time to familiarize themselves with the matter. In addition, hints attached to the tasks are often ignored, which leads to incorrect results.

---

6 Analysis of Behavioral Patterns in Room-X

Known behavioral patterns from research about problem solving (cf. [4]) apparently also become visible in Room-X, as far as the ongoing evaluation of the data shows.

Convergent and Divergent Thinking. Teams that are able to combine convergent thinking with divergent thinking are more efficient. It also seems to be helpful if there are team members who are focused on solving the computer science tasks and those who are able to recombine the results with other details. However, it is not very effective to switch frequently between the individual tasks. (Data shows, that teams with more than two convergent thinkers are usually successful. However, teams with a chaotic approach are

usually unsuccessful.)

Convenience. Room-X is not a *complex* problem, but the presence of the team members and the different puzzles require some mental work to keep the overview and to recognize what the structure of the overall problem and its solution looks like. It is therefore a matter of gaining "system knowledge". Teams with good coordination, communication and documentation have an advantage here as described in Sect. 5.4, since the thinking capacities of the team members are tied up in tasks and can only partially deal with the overall view.

Overstraining. The cognitive system reacts to tasks that seem too complicated by preferring other, simpler tasks, even if they seem less relevant or even when the importance of the difficult task is recognized. This effect can also be observed in Room-X. There are teams who postpone or ignore computer science tasks to the end, but instead complete all simple search tasks very quickly. In Room-X this is especially visible for teams with little CS background knowledge.

Protection of the Sense of Competence. It is also not uncommon to observe actions in which participants ignore clues they have found if these did not fit into the solution strategy they had devised. This effect apparently protects the problem solver from getting into a feeling of inability to act if a hint does not fit into the plan. Thus, the hint is rather put aside, than that it ruins the solution plan.

---

7 Threats of Validity

As the video analysis is still in progress, there are shortcomings in the area of test validity. The objectivity of implementation can be assumed since the situation as an escape game has no direct test character and is carried out independently and without the influence of the researcher. Due to the number of runs, effects that can occur due to irregular external conditions are reduced. Evaluation and interpretation still require analysis by at least one other person in order to achieve a certain degree of objectivity. The same therefore also applies to the reliability of the evaluation and interpretation of the video material. The operationalized characteristics and behaviors must be made available to the second analyst as a manual and interpreted according to fixed rules in order to achieve the highest possible degree of validity.

---

8 Conclusion and Perspective

Room-X is an escape game with computer scientific tasks that requires general problem-solving strategies. An advantage of the current concept is that no prior knowledge of specific problem-solving procedures is required. Thus, it can address a broad audience as it only expects skills most tenth-grade learners already have. With regard to the individual tasks, it is advantageous that even teams with little or no prior knowledge can be given a motivating insight into ideas of computer science.

In order to be able to adjust the computer science content according to prior knowledge and skills of the visiting groups, the development of interchangeable modules is planned, which emphasize the usefulness of the computer scientific ideas and the meaning of computer science methods.

First indications for the promotion of problem-solving skills can be derived from the

results of the video analysis. As for RQ1, many typical basic problem-solving strategies such as systematic search, team collaboration, documentation could be observed as predicted. As for RQ2, the successful teams are more motivated and determined and are better at documentation and communication. They search more thoroughly and work on the tasks more concentrated. In short, they work in a more structured way. In terms of documentation, the low tendency of the teams towards structured representation is particularly noticeable. The causes for this must therefore be examined more closely.

Ideally, problem solvers should use a cleverly chosen strategy to move purposefully through the problem space. The video analysis showed, however, that no planning phase takes place, but instead participants start with the search for clues, since the current concept does not require a planning phase. In order to strengthen the problem-solving aspect in the future, the current procedure must be replaced. One possible approach is to examine computer science concepts in terms of their structure and integrate them as a (sub) strategy.

References

1. Anderson, L.W., Krathwohl, D.R.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York (2001)
2. Arbeitskreis Bildungsstandards SII: Bildungsstandards Informatik für die Sekundarstufe II. Supplement to LOG IN 183/184 (2016)
3. Denning, P.J.: Great principles of computing. Commun. ACM **46**(11), 15–20 (2003)
4. Dörner, D., Kreuzig, H.W., Reither, F., Stäudel, T.: Lohhausen: vom Umgang mit Unbestimmtheit und Komplexität. Huber (1983)
5. Grillenberger, A.: Von Datenmanagement zu Data Literacy: Informatikdidaktische Aufarbeitung des Gegenstandsbereichs Daten für den allgemeinbildenden Schulunterricht. Dissertation, Freie Universität Berlin (2019)
6. Humbert, L., Puhlmann, H.: Essential ingredients of literacy in informatics. In: Magenheim, J., Schubert, S. (eds.) Informatics and Student Assessment - Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics. LNI Seminars, vol. 1, pp. 65–76. GI, Bonn (2004)
7. International Society for Technology in Education (ISTE): Operational definition of computational thinking for K-12 education (2011)
8. K-12 Computer Science Framework Steering Committee: K-12 Computer Science Framework. Technical report, Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative, New York, NY, USA (2016)
9. Müsseler, J., Rieger, M.: Allgemeine Psychologie, 3rd edn. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-642-53898-8
10. Nicholson, S.: Creating engaging escape rooms for the classroom. Child. Educ. **94**(1), 44–49 (2018)
11. Rosenstiel, L.: Grundlagen der Organisationspsychologie: Basiswissen und Anwendungshinweise, 7th edn. Schäffer-Poeschel, Stuttgart (2011)
12. Schoenfeld, A.H.: Reflections on problem solving theory and practice. Math. Enthusiast **10**(1), 9–34 (2013)
13. Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik **25**(1), 20–31 (1993)
14. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)