Learning about black-boxes: A mathematical-technological model

André Greubel[1] and Hans-Stefan Siller[1]

[1]University of Wuerzburg, Germany, andre.greubel@uni-wuerzburg.de

Technology fastly becomes more relevant, both in real life and within mathematical and technological education. In everyday life, this technology frequently comes in form of a black-box, a system whose inner-workings are (partly) unknown to the user. In this paper, we argue that we should emphasize teachings about black-boxes in technological and mathematical education. This is because a black-box conception of technology is neglected by current teachings, but crucial for everyday life. To bridge this gap, we present a model that can be used as a basis for such teachings. We argue why the containing conception and analysis techniques are useful for mathematical and technological education. More precisely, we argue that such a model can be used as meta-informational knowledge to reflect upon technology, especially in cases with incomplete information.

Keywords: black-box conception, black-box analysis methods, mathematical-technological education, curriculum development, teaching models

Introduction

With the progression of the 21st century, technology becomes more relevant, both in real life and within mathematical and technological education. This is primarily because "[d]igital technology has the potential to open up new routes for students to construct […] new approaches to problem-solving" (Bray & Tangney, 2017). As such, technology is included more and more frequently in education.

This inclusion of technology can manifest itself in different ways. Firstly, one can differentiate between *tools* (used to automatically solve generic inner-mathematical problems) and *applications* (used to solve a specific real-world problem). Secondly, one can teach *with* and *about* technology. Teaching *with* technology focuses on teaching regular subject areas, but in a better way (whereas "better" can be understood in a huge variety of ways). Contrary to that, teaching *about* technology corresponds to an epistemic mediation of technology and focuses on goals internal to the user, e.g., his familiarity with or conception about technology (Rabardel & Bourmaud, 2003; Trouche, 2005).

Thus, teaching about technology can foster a reflective approach to the usage of technology (*critical reflection*). Notably, this critical reflection requires (some) insights into that technology. However, the mathematics of complex applications (like search engines, product recommendation, or self-driving cars) frequently exceeds the scope of typical school education. Additionally, many of these technologies hide their inner-workings from the user. Hence, at least parts of these applications must be treated as a black-box. Because of this, it is worth asking: "*What conceptions and techniques should be taught to students to foster their critical reflection of complex black-box applications?*"

In this paper, we contribute to this question with a normative approach. We first propose a theoretical model consisting of a black-box conception of technology and five analyzation techniques. Afterwards, we describe why we should teach this model to students. More precisely, we show how this model can be used as meta-informational knowledge to reflect about complex applications.

State of the Art of Teaching about Technology

Teachings about technology can be done both explicitly and implicitly: Students might get explicit instructions of how they should view and use technology, or they build their own model by their experience about their interaction with technology and its inclusion in their education.

Current Curricula Often Implicitly Build a White-Box Conception

As of right now, explicit teaching *about* technology is not part of most curricula. For example, in Germany, the central document guiding the creation of the state-local mathematics curricula states:

> "The inclusion of digital tools should foster the development of mathematical skills. The potential of such tools should support the exploration and understanding of mathematical relationships, the use of individual approaches to problem-solving, and reduce the focus of schematic activities in problem-solving." (KMK, 2015, p. 13, translated from German)

Thus, mathematical education focuses on mathematical *tools* and on teaching *with* technology; no explicit conception for technology is introduced. Hence, it is likely that students learn their conception implicitly, based on their learning and everyday experiences. Currently, the most prevalent theoretical model used to describe the inclusion of technology during learning is the technology enhanced modelling cycle (Greefrath et al., 2011). In this cycle, a mathematical model is translated into a technological model and the technological results are interpreted as mathematical results.

Thus, this model focuses on a white-box conception of technology: Students have full understanding of what the system does, how it is defined, why it is doing it, and know about all inputs and outputs.

Additionally, German computer science education complements this approach by teaching about how a system is doing something. This also includes lessons on how to implement such systems, e.g., using object-oriented programming. Notably, these courses also use a white-box conception where technology directly corresponds to a known model. Teaching about these models (e.g., a computer, the internet, databases, or an UML diagram) is an important part of the teaching (c.f. Röhner, 2016).

Strengths of the White-Box Conception of Technology

With this (implicit) white-box conception, students are able to understand a fundamental property of technology: In order to build or use it, explicit and correct translation between these representations is necessary. Using technology in this way can increase the scope of solvable problems with automaton. This might be because a single approach is executed faster, or multiple approaches can be executed simultaneously. However, it is not possible that technology solves problems a human (with infinite time) cannot solve, since technology is seen as a mere extension of a mathematical model. As such, technology can only be as valid as the models used to create it. More precisely: any output is an inherent property of the used model or algorithm, and not of the method of execution.

This coherence to a mathematical model can motivate the discussions of various concepts like determinism (the output of the model is defined only by its behavior and input values) and edge-cases (situations that defy the assumptions of the model). It can also be used to introduce the important difference between verification (the technology accurately represents the mathematical model) and validity (the model accurately represents the real-world).

Limitations of the White-Box Conception of Technology

While this current, implicit white-box conception accounts for many important properties of technology, it fails to account for black-boxes, in which (at least important parts of) the inner-workings of the technology are unknown to the user. Notably, this limitation is not the fault of the current teaching methods like the technology-enhanced modelling cycle: They were never meant to build a full conception of technology. Regardless of fault, the lack of an explicit black-box conception is undesirable.

Firstly, while technological white-boxes are prevalent in education, black-boxes are far more prevalent in everyday life. This is primarily because every technology not created by the user is, at least initially, a black-box. While it might be common that (in education) all applications or models within a tool are created by students, this is not true for the majority of applications used in everyday life.

Secondly, the impact of these black-box applications on our everyday life is significant. They might affect the information we see (Google), the peer group we interact with (Facebook), dates we arrange (Tinder), the safety of our commute (self-driving cars or trains), our fitness (smart watch and fitness apps), and even our health (medical technology like automatic insulin pumps).

And lastly, black-boxes are often used in the workplace to hide mathematics (Williams & Wake, 2007). As such, designing or working with these black-boxes is a necessity in many jobs. Hence, there is an additional incentive to teach about black-boxes as vocational preparation.

Overall, black-box applications are very prevalent both in everyday life, and in the workplace. As they might have a significant impact on both, a reflective approach to their usage is desirable. However, as the conception of a black-box is currently not taught, there is little students can actually do to pursue such a reflective approach.

Proposition: A Model for Teaching about Black-Boxes and their Analysis

As such, we propose to bridge this gap by explicitly teaching about black-box technology in mathematical and technological education. We describe a conception of black-boxes and five techniques that can be used as basis for such teachings. These techniques differentiate in the amount of resources necessary (from low to high) and the amount of insight gained by utilizing them (from high to low).

Proposed Conception: Black-Boxes

Firstly, we denote the proposed conception students should have about black-boxes:

> Frequently, (parts of) the inner-workings of a given application are unknown to the user. This occurs naturally if the creator and user are not the same person. In this case, a user typically has no influence on the quality of the software if applied in a given situation. Similarly, the creator typically has no influence on the situations a software is applied.

The most important aspect a user must know about a technological black-box is, that it still has all of the properties of a white-box. Most importantly, its behavior and results are deterministic. Notably, this includes the output of pseudo random number generators, as they solely rely on their input value (the seed). This is also true if the user (or, in the case of neuronal networks, even the creator) cannot explain the exact functionality. This also highlights the most important limitation of any technology:

It cannot (magically) adjust to the requirements of a situation and evaluating the quality of the result requires (some) knowledge about the (mathematical) models used to acquire that result.

Nevertheless, it importantly is often still possible to use the technology, even with this lack of understanding from the user or the lacking intention of the creator for this use-case. Because of this, it is crucial that the quality of the software and its applicability to a given situation were thoroughly evaluated. Otherwise, wrong or misleading results might follow, whose consequences can be dire.

Proposed Technique 1: Accepting the Black-Box

The first approach is *accepting* the black-box. In this case, the technology is used without examination and evaluation. We will call such a technology a *true* black-box.

When using a true black-box, one has knowledge about the most important input and output of that black-box: The most important output corresponds to why we use the black-box (e.g., for a self-driving car: "moving towards a destination") and the most important input corresponds to how we use the black-box ("by inserting a destination into the user panel"). However, even this knowledge about the inputs and outputs might be incomplete or wrong.

At this point, it is important to note that true black-boxes both exist and have educational legitimacy.

On the one hand, accepting a black-box and using it as a true black-box is the default (and natural) behavior of many people. However, it is hard to discuss how desirable this is or to evaluate the consequences of that approach without explicit acknowledgement of the conception of a true black-box.

On the other hand, accepting a black-box can sometimes be the most reasonable choice: It might just be unfeasible or impossible to analyze a black-box at all. For example, many users will (likely) never have the ability (or feel the need) to analyze the algorithms used in self-driving cars themselves.

Proposed Technique 2: Testing the Black-Box

The first evaluation approach including interaction with the black-box is *testing* it, leading to a *tested* black-box. In this case, the technology is executed in a safe environment using specific inputs. If the observed output matches some expected output (to some precision), the technology is then applied to the real problem. Thus, if something fails, the resulting problems are limited to the safe environment.

For this technique, learners need to understand that it only works if the same inputs are used during testing and usage and if all relevant outputs are observed. It is important to fight the misconception "if all observed outputs are correct using one input, then all outputs are correct using any input". Notably, this can be difficult, as the user does not necessarily know about all inputs and outputs.

However, a tested black-box already offers some amount of information to the user: If the test was successful and seems to represent the future use cases, we can infer that the most important inputs and outputs are likely recognized at this point and that the future outputs will suit the future use-cases.

Proposed Technique 3: Integrating the Black Box

A more comprehensive approach is *integrating* the black-box: First, the black-box is modeled as an unknown function from some inputs to some outputs. Notably, "some inputs and outputs" does not necessarily mean "all relevant inputs and outputs". Then, outputs to selected inputs can be collected.

This systematic collection of inputs and outputs (rather than unsystematically collected tests), leads to more information as some generalizations can be made. For example, a user might notice that all outputs fall into some magnitude, regardless of input variation. Additionally, if we assume that the inputs are representative for any future use-case, we can infer that the product is likely safe to use.

Proposed Technique 4: Inferring the Black-Box

An even more comprehensive approach is *inferring* the behavior of the black-box. In this case, a re-modelling approach is taken: A mental model of the system is build, which is then verified by comparing coherence between this mental model and the actual behavior. As such, this techniques does not rely solely on some inputs and outputs of the black-box, but also takes the behavior into account.

The model-building process itself might use the observed dependency between inputs and outputs as basis for such a model and follow the steps of a modelling cycle. Regardless of approach, the resulting model can be represented in a variety of ways. This includes mathematical formulas ("speed of car = maximum allowed speed - 10"), verbal statements on different levels of abstraction ("The motor only starts if the battery is not empty", "The car follows the traffic laws"), algorithms ("if the cars sees a red light, then it breaks until standstill"), and meta-information ("This process takes 5s.").

The depth of both the mental model and the verification can vary significantly: The model can be a simple approximation (e.g., using a linear function) or be identical to the actual model. The verification can reach from a single execution in a common scenario to complex statistical tests using several carefully-constructed edge-cases (e.g., "how often does the motor starts if the battery is at 1%?").

Depending on the depths of the models and verification used, it is now *possible* to gain accurate knowledge about the exact lists of inputs and outputs, and the inner-workings of the technology. But this is no necessity: The model is only inferred and relies solely on a finite cutout of the observable behavior of the technology – contrary to any secured or proven knowledge of its inputs, inner-workings, or outputs. As Rice's theorem states that it is impossible to determine the exact functionality of a system without knowledge of its inner-working, this imples that this approach cannot lead to secured knowledge – even if the mental model used is actually the same as the real model (c.f. Rice, 1953).

Proposed Technique 5: Opening the Black Box

The fifth approach relies in *opening* the black-box. In this case, students not only examine its observable behavior, but also its inner-workings (*reverse engineering*). This approach can transform the black-box into a white-box. However, for comprehensive technology, accessing and understanding the implementation of a system is frequently outside the scope of education. For very comprehensive technologies, it might also be out of scope for any single person.

While it is frequently unpractical or impossible to open a black-box, understanding the limitation of the prior approaches only becomes possible if they are explicitly contrasted with this technique.

Additionally, some seemingly arbitrary aspects (like calling things that are treated as secured knowledge a "theory" in science, or arguing why mathematics can be seen as part of the humanities) only becomes comprehensible after understanding the epistemological difference between inferring and opening black-boxes.

Reasons for Teaching our Model

In this section, we show how the proposed model can be helpful and why to include them in curricula.

Reason 1: Intermediate Stages of Black-Boxes as Meaningful Conceptions

First and foremost, our model introduces "intermediate stages" of black-boxes (e.g., tested or integrated black-boxes) as meaningful conceptions. Without such an explicit conception, one might think about black-boxes as "incomplete white-boxes", i.e., a white-box that is left to be opened or inferred. Indeed, this conception was dominant in initial black-box research (e.g., Glanville, 1982) and remains relevant in many recent research projects. See (Krell & Hergert, 2019) for an overview of approaches.

However, this conception is not useful in practice: It is not always desirable to open or infer a black-box. Instead, depending on the time, skill, and jurisdiction of the student, the complexity of the technology under consideration and its available implementation, documentation, explanation, and licenses, it might be impossible, unfeasible, illegal, or uneconomic to apply some of these techniques.

This is quite apparent for self-driving cars or search engines: Access to their source code is limited because of intellectual property rights. Additionally, the complexity of their source code likely exceeds the capability for analysis for most (even trained) single individuals. Similarly, building and verifying an accurate mental model for such a technology also is both very hard and time-consuming.

As such, it is naïve to argue "one should always open or at least infer a black-box". In reality, opening or inferring certain black-boxes is not something most individuals will opt to do. But notably, this does not imply that one should not analyze a system at all. Instead, the intermediate stages show alternative and frequently valid courses of action or analysis one can take.

Reason 2: Stages of Black Boxes as Meta-Informational Knowledge

Secondly, our model highlights aspects of the trade-off between invested resources and gained insights. Thus, the techniques and their potential insights and requirements act as meta-informational knowledge: It can help to build an informed opinion on "whether one knowns enough about a black-box for a specific use-case, given the resources one is able or willing to invest".

As such, these stages can help to assess ones knowledge and identify limitations. Notably, this includes assessing ones knowledge about aspects where one has incomplete or uncertain information. Notably, the assessment of such information is an important step while "deciding what do believe or do", a phrase often used as definition of critical thinking (Ennis, 1987).

This is especially important for political participation: If one assesses that it not possible or feasible to analyze a black-box oneself, one has to ask about the implications. Most importantly, it might be desirable that somebody else analyses the black box. How to design such systems of evaluation can be an important part of a political debate, because different legitimate interests of stakeholders might collide: For example, a user has the desire to use a safe product even if he cannot verify its safety for himself. However, a company in a free market wants to keep certain implementation details as a company secret to keep their advantage over competitors and a government might try to reduce public expenses for institutional verification. Thus, a compromise between these legitimate interests has to be designed and evaluated. This requires insights into the technical aspects of potential solutions.

Illustration of the Benefits: Application of our Model in a Thought Process

To illustrate the prior benefits, we want to come back to the example of self-driving cars. More precisely, we want to exemplarily show how our model can guide the thought process of coming to a meaningful conclusion for the question “How likely is an emergency to occur while using this car?”

With our explicit model, we can first structure our knowledge: We can list available techniques for analysis and consider how they could be applied to the current application (the self-driving car):

<u>True Black Box</u>: Initially, one does not know anything about a self-driving car, expect for its basic functionality. It takes a destination and afterwards drives to this destination. Treating a self-driving car as a true black-box is easy, but reveals nothing about its safety.

<u>Tested Black Box</u>: A self-driving car can be tested in a specifically designed training course. If the course requires some representative situations (like interacting with different car or reading traffic signs), we can conclude that the car is safe to drive at least in some common situations.

<u>Integrated Black-Box</u>: To integrate the self-driving car, we can observe whether the car produces correct output (i.e., drives correctly and according to the traffic laws) for most anticipated inputs. If we chose a comprehensive set of inputs (e.g., a representative selection of all roads of the corresponding country and in a variety of different weather and traffic situations) and observe no (or few) invalid outputs, we can infer that the car is probably safe to use for most everyday cases we want to use it for

<u>Inferred Black-Box</u>: To infer more information about the self-driving car, we could build a set of models based on its behavior. For example, one can build the sub-model “If the car sees a red light, it breaks until standstill” and verify its correctness by building a representative amount of situations (including different distractors like weather and traffic). Afterwards, one can combine several sub-models (and verify this combination) to acquire an accurate mental model of the self-driving car. With this method, we can infer the safety of the car if the mental model indicates safe behavior.

As a second step, we can use this knowledge about the techniques to decide what to do: Using the system as a true black-box does not provide sufficient information about its safety. Since vehicle malfunction can be fatal, testing in a single environment also seems inadequate. However, both integration and inferring can give valuable information about its safety. Nevertheless, they require detailed knowledge and many resources (like verifying the behavior of the car in multiple real-world situations). As such, one might opt to not do this oneself, but rather vote for an obligation (by the producing company or by government regulators) to verify its safety using integration or inferring.

Overall, we used this model to progress from a simple “we don´t have any idea about its safety” to a more sophisticated “There should be an external entity that verifies its safety. An important aspect of this verification is the amount of hours driven in representative situations (rather than test scenarios)”.

Notably, this assessment benefits from knowledge about the techniques in the model. On the one hand, this is because they highlight the difference between behavior in a test vs. a representative environments (test vs. integration). On the other hand, the metric “amount of hours driven in representative situations” is a consequence following from the application of integration or inferring. Hence, it might be overlooked if only thinking in conceptions of “true” and “opened” black-boxes.

Summary and Conclusion

In this paper, we proposed a model of black-boxes in STEM education and argued why it accounts more for the special requirement of mathematical and technological education. This is primarily because technological system relevant for everyday life (especially those based on complex mathematical models like neuronal networks) are frequently too complex to open with typical school education.

Then, we argued why we should include teachings about black-boxes in mathematical and technological education. This is because, firstly, the conception of intermediate stages of black-boxes (rather than “incomplete white-boxes”) are useful constructs if opening a black-box is impossible or unfeasible. As discussed before, this is often the case with complex mathematical technology. Secondly, our model can act as meta-informational knowledge to assess ones reflection about a system that is too complex for a full analysis. This also enables the generation of new knowledge. Thirdly, our models highlights that one does not necessarily know about all inputs and outputs of a black-box.

However, the proposition of this model can only be seen as a first step: How to design curricula and lessons based on this model is an important question for future research. Furthermore, it would be desirable to unite our technology-focused model with existing models in science education to achieve a more general “STEM approach to black-boxes” usable in interdisciplinary teaching.

References

Bray, A., & Tangney, B. (2017). Technology usage in mathematics education research. *Computers & Education*, 114, 255–273. https://doi.org/10.1016/j.compedu.2017.07.004f

Ennis, R. (1987). A taxonomy of critical thinking dispositions and abilities. *WH Freeman/Times Books/Henry Holt & Co*.

Glanville, R. (1982). Inside every white box there are two black boxes trying to get out. *Behavioral Science*, 27(1), 1–11. https://doi.org/10.1002/bs.3830270102

Greefrath, G., Siller, H.-S., & Weitendorf, J. (2011). Modelling Considering the Influence of Technology. In *Trends in Teaching and Learning*. https://doi.org/10.1007/978-94-007-0910-2_32

Sekretariat der Ständigen Konferenz der Kultusminister [KMK] (Ed.). (2015). Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife [Educational Standards for A-Levels in Mathematics]. *Appel & Klinger Druck und Medien GmbH*.

Krell, M., & Hergert, S. (2019). The Black Box Approach. In *Towards a Competence-Based View on Models and Modeling in Science Education*. https://doi.org/10.1007/978-3-030-30255-9_9

Rabardel, P., & Bourmaud, G. (2003). From computer to instrument system: a developmental perspective. *Interacting with Computers*, https://doi.org/10.1016/S0953-5438(03)00058-4

Rice, H. G. (1953). Classes of Recursively Enumerable Sets and Their Decision Problems. *Transactions of the American Mathematical Society*, 74(2), 358. https://doi.org/10.2307/1990888

Röhner, G. (2016). Bildungsstandards Informatik für die Sekundarstufe II. [Educational Standards for Higher Secondary Education in Computer Science]

Trouche, L. (2005). Instrumental Genesis, Individual and Social Aspects. In *The Didactical Challenge of Symbolic* (pp. 197–230). https://doi.org/10.1007/0-387-23435-7_9

Williams, J., & Wake, G. (2007). Black Boxes in Workplace Mathematics. *Educational Studies in Mathematics*, 64(3), 317–344. https://doi.org/10.1007/s10649-006-9039-z