The Argument Assessment Tutor (AAT)

MICHAEL H.G. HOFFMANN
Georgia Institute of Technology
m.hoffmann@gatech.edu

Intelligent tutoring systems provide learning opportunities that adapt to individual learning pathways. This contribution discusses challenges that any ITS faces, and it presents a first version of the newly developed "Argument Assessment Tutor" which familiarizes learners with a strategy to identify problems in bad arguments. The AAT allows practicing the use of seven criteria to assess the quality of arguments. The talk also discusses limitations of this approach and problems that need to be addressed.

KEYWORDS: argument assessment, argument quality, ARS criteria, intelligent tutoring system, ITS

1. INTRODUCTION

There is substantial empirical evidence that instruction that adapts to individual characteristics of learners—such as their prior knowledge, strategies, errors, and learning styles—is more effective than instruction that treats all learners as the same (Aleven, McLaughlin, Glenn, & Koedinger 2017; Aleven et al. 2018; Walkington 2013; Kulik, Kulik, & Bangert-Drowns 1990; Corbett, McLaughlin, & Scarpinatto 2000). Learners change as they learn, and that should be taken into account in teaching. As apprenticeship or mastery learning, individualized instruction has been known for a very long time. But to offer such a learning experience in the classroom—not to speak about the even larger scale of online learning—proves to be a challenge.

Starting already in the late 1970s, this challenge has been addressed by considering the possibility of "computer-based coaches" and the use of "artificial intelligence techniques" and "self-improving teaching systems." The first book about *Intelligent Tutoring Systems* (ITS) that used these terms was published in 1982 (Sleeman & Brown 1982). Today, ITSs are the focus of a lot of research and software development (see overviews by Peña-Ayala 2013 and Aleven et al.

2017; for the early history see Ohlsson 2016; an ITS conference series exists since 1988).

This contribution is about two questions. There is a widely shared assumption that ITSs are possible only for "well-defined domains where knowledge about the domain being taught can be explicitly modelled," such as mathematics, computer science, or chemistry. "For ill-defined domains, human tutors still by far outperform the performance of ITSs, or the latter are not applicable at all" (Gross, Mokbel, Hammer, & Pinkwart 2015, p. 413).[1] This leads to the first question: Can the process of assessing the quality of arguments be modelled precisely enough to allow the creation of an intelligent tutoring system? If that turns out to be the case, then the second question is: How could an ITS be designed that is able to provide intelligent, one-on-one, computer-based support to students as they learn how to assess the quality of arguments?

Since everybody should be able to *create* arguments of high quality and to *identify* weaknesses in given arguments, the ability to assess the quality of arguments is of crucial importance. Justifying claims by reasons—as the notion of "argument" is understood here—can be seen as the core of both scientific activity and deliberation in public and private spaces. Doing it well requires that people acquire the criteria needed to assess the quality of arguments and learn how to use them. Argument assessment is a skill whose development should be an essential part of education.

Most textbooks on critical thinking, informal logic, and argumentation provide useful material for learning how to assess the quality of arguments, including exercises. But they do not offer individualized feedback to learners as they struggle to acquire the necessary skills. It would be highly beneficial for education to have an automated system that works with each student like a human tutor by providing instruction; offering exercises; monitoring how an individual student is doing on the tasks selected; providing feedback both to successful problem solutions and to things that are not done correctly, including explanations of why certain solutions are not acceptable; and selecting further tasks based on an understanding of what the student needs to practice to realize their personal pathway to successful learning.

---

[1] There are attempts to develop ITSs also for ill-defined domains and problems (Lynch, Ashley, Aleven, & Pinkwart 2006), but all these approaches nevertheless employ justifiable, normative standards (more on normativity in Section 4 below); the systems' responses are not arbitrary. This means that these systems operate in areas that are well-defined at least to a degree that allows this kind of normativity. For the debate on how to define "ill-definedness," see Lynch, Ashley, Pinkwart, & Aleven 2009.

The Argument Assessment Tutor (AAT) that is described in this contribution already exists in the limited form of seven assessment tasks that can be done online at https://reflect.gatech.edu/aat. The system provides immediate feedback to a learner's attempts to identify problems in bad arguments. All assessments tasks are structured in form of a checklist which directs the learner's attention to seven criteria that should be used for quality assessment. The main learning goal of the AAT is to foster a deeper understanding of these criteria and to familiarize learners with this checklist. Learners are supposed to internalize the sequence of assessment steps that is structured by this checklist, that is, to develop a habit of assessing arguments along this particular sequence of steps.

The method is learning by doing. The AAT provides an argument and guides the learner through the assessment procedure by asking questions that invoke particular quality criteria. Each question is followed by a list of possible answers and, depending on the user's selection of an answer, the tutor either confirms the answer or provides an explanation why it is not correct. This way, students should internalize the use of the checklist by practicing its application in the assessment of arguments. Learning is supported by feedback that adapts to the student's growing expertise. Whereas a primary set of tasks guides the user step-by-step through the checklist, a second set is designed for learners who are already familiar with this list of criteria. After presenting an argument map, it starts immediately with the question: "What should be criticized in this argument? If you think that there are multiple problems, select the one highest in the list."

Such a "checklist tutor" seems to be a novel idea. Its general design could be useful for all teaching that focuses on familiarizing students with a structured sequence of cognitive activities. One example is the training of coders for research projects, or of medical professionals for diagnostic tasks (El Saadawi et al. 2008).

Even though this preliminary version of an AAT already exists, this does not mean that the first question guiding these considerations should be taken as answered by implication. The existing prototype only demonstrates that the seven quality criteria can be modelled in form of a checklist. This leaves two questions open. First, does this checklist cover what can be considered to be the core of argument assessment? There are already two other pilot systems that seem to be too limited: one focusing just on fallacies (Diana, Stamper, & Koedinger 2018), and the other one on the identification of weaknesses in graphical representations of legal arguments that students create after studying transcripts of oral arguments in court (Pinkwart, Aleven, Ashley, & Lynch 2006). The second question is whether the quality of the AAT model is good enough to support learning. So far, the main function of the prototype is to illuminate problems that need to be resolved before this line of research and development can be pursued further.

This contribution is divided into three parts. In Section 2 I am going to describe the challenges that any ITS faces. Section 3 will summarize a theory of argument assessment that I developed elsewhere, and it will show how a corresponding step-by-step method of argument assessment has been implemented in the existing AAT. In Section 4, finally, I will discuss some of the problems that still need to be addressed.

2. INTELLIGENT TUTORING SYSTEMS (ITS): CHALLENGES

Whereas a human tutor can rely on implicit knowledge about instruction and learning that comes with expertise, an artificial tutor requires explicitly formulated models of the processes involved in learning. Usually, an ITS architecture requires three cognitive models.

1. The *domain model* (also called expert knowledge) "contains the concepts, rules, and problem-solving strategies of the domain to be learned. It can fulfill several roles: as a source of expert knowledge, a standard for evaluating the student's performance or for detecting errors, etc. It is sometimes organized into a curriculum, a structure including all the knowledge elements linked together according to pedagogical sequences" (Nkambou, Bourdeau, & Mizoguchi 2010, p. 4).
2. The *student model* which describes the learner's "cognitive and affective states and their evolution as the learning process advances" (ibid.). In contrast to the domain model, the student model is a dynamic model. It needs to explain why a learner makes certain mistakes. Corresponding research goes back to the observation that student errors in learning mathematics are not random; they follow certain patterns that are conceptualized as inappropriate cognitive models of the domain to be learned. These insufficient models lead to "buggy" procedures and corresponding mistakes. As a consequence of this conceptualization of student errors, ITS researchers developed so-called "bug libraries, repertoires of cognitive models that deviated from the correct mathematical skills in such a way as to generate the erroneous answers observed empirically" (Ohlsson 2016, p. 460).
3. The *tutor model* connects the domain and student models. It is designed to provide help if a student requests a hint or to make certain decisions based on a student's input. All this is then presented in the user interface of the tutor which is sometimes counted as the fourth component of an ITS architecture. The tutor

> model determines "whether or not to intervene, and if so, when and how. Content and delivery planning are also part of the tutoring model's functions" (Nkambou et al. 2010, p. 4). For example, if the tutor is designed to provide immediate feedback on errors, then each "bug" in the student's cognitive model that becomes visible in a mistake needs to be answered by specifically designed feedback that presents, in some way, the corresponding component of the domain model to the student.

It is obvious that it takes substantial effort to create these three kinds of models. One way to simplify this task has been conceptualized as "constraint-based modelling" (CBM; Ohlsson 1993, 1994). Based on the observation that, in the process of learning, the domain knowledge plays a *normative* role for the learner—it tells them what they *should* do—Stellan Ohlsson suggested to conceive the declarative or propositional knowledge that is usually considered to be at the core of a knowledge domain as *constraints*, that is, as "knowledge elements that encode prescriptive rather than descriptive knowledge."

> The type of constraint used in CBM has the general form, "when such-and-such conditions are the case, then such-and-such other conditions ought to be the case as well" (or else something has gone wrong). For example, *when driving a car in New Zealand, the driver had better be driving on the left side of the road* (or else he or she violates the traffic laws of that country). Clearly, a speed limit is not a description of actual behavior, but a prescription. Formally, constraints of this sort take the form of ordered pairs of patterns, <Cr, Cs>, where each pattern is a conjunction of conditions. Cr is a *relevance criterion* that circumscribes the set of situations for which the constraint applies (*when driving in New Zealand*), and Cs is a *satisfaction criterion* that determines whether the constraint is satisfied (*drive on the left side of the road*). The set of constraints that apply to a problem type or in a particular task environment is called a *constraint base*. (Ohlsson 2016, p. 465; his italics)

The main advantage of Ohlsson's constraint-based modelling for the design of intelligent tutoring systems is that it does neither require a student model nor a tutor model. Everything needed can be derived from an analysis of the knowledge domain.

> Such an ITS would apply the constraints to each new problem state and flag violated constraints. Pre-formulated instructional messages would be associated with the constraints, and presented to the student when one or more

> constraints are violated. This is the core of the constraint-based approach. One notable advantage is that the constraint-based approach does not require empirical studies of students' errors or the compilation of bug libraries, because constraints encode correct domain knowledge. This seemed to me then, and seems to me still, a simpler and more elegant design for an ITS than to organize it round either a bug library or an expert model of the target skill. (Ohlsson 2016, p. 467)

The Argument Assessment Tutor (AAT) presented here follows, at least in its current, limited version, Ohlsson's CBM approach. The seven criteria of good arguments and the way these criteria are organized in the checklist provide a normative standard that can be spelled out in the form of constraints. When a student works on an AAT task, she is confronted with an argument of low quality. What is wrong with this argument is determined by one of the seven criteria. This is the criterion that is *relevant* for this particular argument, it represents Ohlsson's relevance criterion. The task of the learner is to "satisfy" the relevance criterion by correctly pointing out what is wrong with the argument. If she is not able to do so, she violates the constraints that are embedded in the tutor, and the tutor reacts accordingly.

3. THE ARGUMENT ASSESSMENT TUTOR (AAT)

The first challenge for designing an AAT is, thus, to get the normative standard right. What makes a good argument? What are the criteria that can and should be used to assess the largest number of possible arguments?

Unfortunately, it is not possible, in the context of this contribution, to provide a justification of the seven criteria that are used in the AAT. This is part of a larger project that is not yet completed. Suffice to say that these criteria form an extension of the "ARS criteria" formulated by Ralph Johnson and Anthony Blair. They focus on the idea that premises that support the conclusion of an argument should be *acceptable*, *relevant*, and *sufficient* (Johnson & Blair 2006 <1977>). The argument assessment approach that is implemented in the AAT does not only include four further criteria, but it also puts them into a particular sequence that can used as a checklist. The assessment procedure always starts with the question whether the formulation of the conclusion is clear enough. If it turns out that the conclusion of an argument is so badly formulated that it is impossible to judge whether reasons are relevant or sufficient, then the assessment can stop right there. In this situation there is no need to bother about the other criteria. Considerations like this one led to a certain ranking of the seven quality criteria, and to a particular assessment procedure that can simply stop at certain assessment points. The entire assessment procedure that I am currently using is summarized in the decision tree that is depicted in Figure 1.

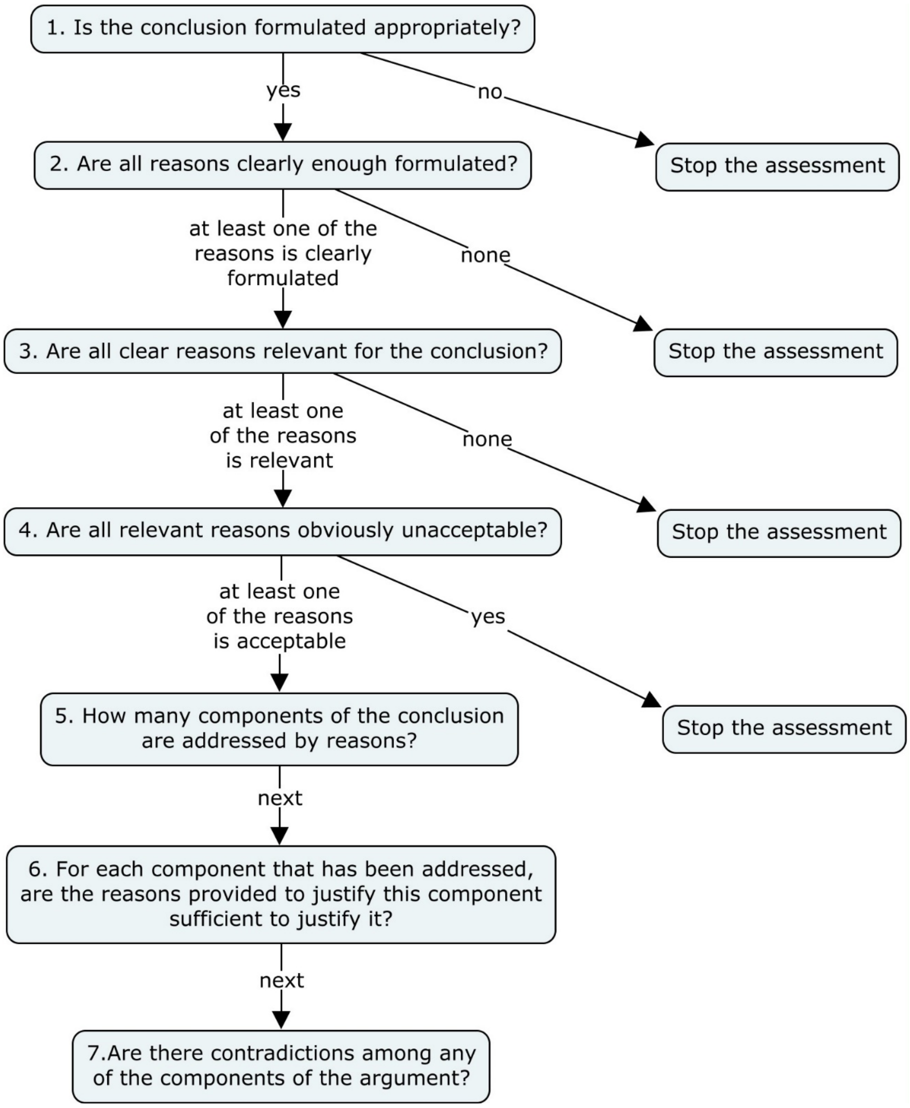

Figure 1 – A decision tree for the assessment of arguments

Most of this decision tree is used in the AAT as the "checklist" that structures the assessment procedure. Each of the seven criteria depicted here is described in greater detail in the instructions of the AAT tasks that are available at https://reflect.gatech.edu/aat. In the next section, I will provide some more detail only for one of these criteria to illustrate particular problems of this approach.

.

4. PROBLEMS WITH THE ARGUMENT ASSESSMENT TUTOR

A first important point to note is that Figure 1 presents the quality criteria only in the form of keywords. Even though the AAT has been designed with the goal in mind to familiarize the learner with these criteria in the process of using the system, a sufficient understanding of these criteria will require instruction that provides additional explanations, examples of their application, and a discussion of particular problems that can be expected.

The more important question, though, is the question whether all this is sufficient. Let me illustrate some of the additional problems with an example. Figures 2 and 3 show what the learner first sees when opening AAT 001 at https://reflect.gatech.edu/aat.[2]

Depending on the user's choice regarding the two options offered in Figure 3, the AAT will either react with "Unfortunately, your answer is wrong. The conclusion is NOT appropriately formulated," or it will show the next question: "Why is the conclusion not appropriately formulated?," followed by a list of options that includes all the possibilities depicted in Figure 3.

The important general point is that an ITS always requires that there is a clear threshold that divides acceptable student responses from unacceptable ones. If the question here were simply: "Is the conclusion formulated appropriately?" then this threshold would not be clear—we do not want, for example, that a learner considers the conclusion as inappropriately formulated based on the typo that can be seen in Figure 2. The seven possibilities that are offered are much more precise than the simple question.

[2] Note that all arguments used in the AAT stem from college students who worked over the course of a semester on a so-called wicked problem (Rittel & Webber 1973; Hoffmann & Lingle 2015; students gave permission to use their maps for publications). However, they are all modified because they usually contained multiple problems. Confronting a learner with too many problems in a task like the one above leads to frustration if the intention of the AAT designer is to focus one problem whereas the learner discovers another one. This turns out to be an important problem for the design of an AAT.

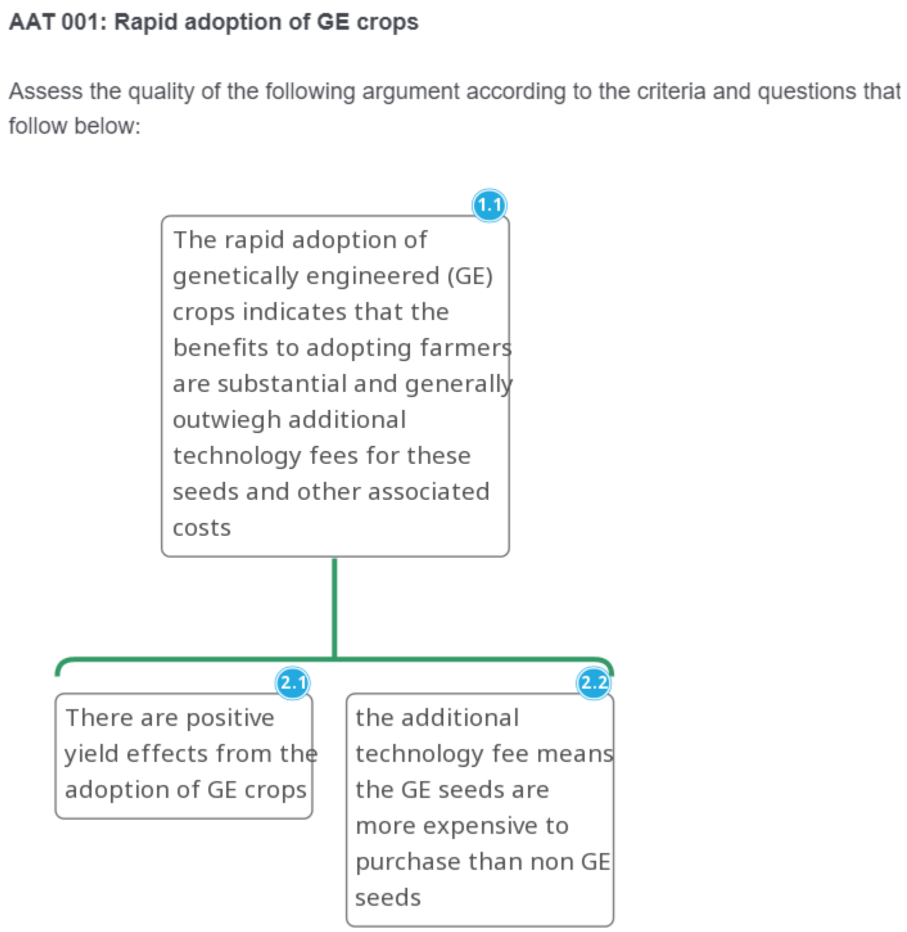

Figure 2 – The beginning of the assessment procedure in AAT 001

But there are several problems. The first one is that it is hard to justify why exactly these seven criteria are used and not others. Based on the fact that the tutor has the power to say "your answer is false" or "is correct," an AAT holds a strong normative position—not only with regard to this particular question, but for everything that is used to distinguish acceptable from unacceptable user responses. This normativity is not so much a problem for human tutors because you can still argue with your teacher. But for automated systems, this is a big problem. To address it, there should be at least some consensus in the scientific community about the criteria that determine an AAT's decisions. Moreover, since it might not be possible to identify possible problems right away, feedback from the users should inform ongoing revisions of the AAT design.

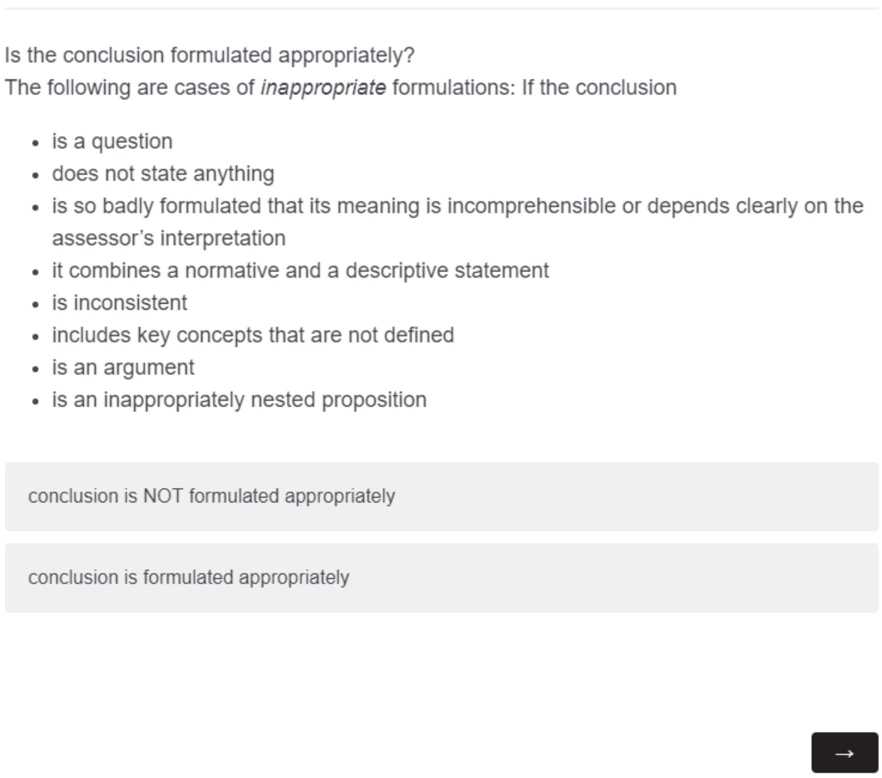

Figure 3 – The first question in the checklist with more specific options of what counts as "not appropriately formulated." Note that "an inappropriately nested proposition" refers to a conclusion such as "Dr. Wiseman claims that dental hygiene is important." If the reasons support that Dr. Wiseman formulated such a claim, then the conclusion is appropriately nested; but if the reasons justify why dental hygiene is important, then it is inappropriately nested.

A second problem concerns the fact that even though the eight specifications of "inappropriately formulated" provide more precision than the general question, the comprehension of anything that is provided by the tutor depends on the background knowledge that a user brings to the task. To be clear: this is not about not knowing the meaning of things like an "inappropriately nested proposition" in the example of Figures 2 and 3. If a user does not know that, this should simply provide a motive to look it up in the instructions. Problematic are situations in which the designer of an AAT applies a certain rubric differently to a particular case than its user. Is the "key concept" "genetically engineered" clear enough or does it require a definition? What exactly does it mean for a particular formulation that its meaning "is incomprehensible or depends clearly on the assessor's interpretation"?

Differing background knowledge is probably the most important challenge for learning with an argument assessment tutor; it is a fundamental challenge for any assessment of an argument (Hoffmann 2018). It is crucial, in particular, for assessing the sufficiency of reasons for a particular component of the conclusion, but also for determining the acceptability of reasons. A part of this challenge can certainly be addressed by instructions that define all seven criteria in more detail, but it should be clear that this problem can never be completely resolved.

The third and fourth problem could be considered as sub-problems of the one relating to background knowledge: confirmation or myside bias regarding the content that is covered by a task, and the possibility that a user looks at a given argument from a perspective that is not anticipated by the designer, or simply alien to him or her. In both cases the problem is that the user might have a point in assessing an argument in a certain way that should not simply be dismissed as wrong by the implicit authority of the system.

The norm-setting authority of an AAT is related to the problem of normativity that we discussed as the first problem above. It has to be acknowledged that learning requires accepting the authority of the tutor to set the norms of what counts as good and bad. If a user perceives certain norm-setting reactions of the tutor as arbitrary or unjustified, then the motivation to engage with the system and to learn will be at stake. If learners lose trust in the authority of an AAT, then this system fails as a learning tool.

A fifth problem poses a challenge for the designer of an AAT. In an effort to come up with clear-cut cases for the tasks, there is a risk of trivializing the assessment so that not much gets learned. This problem is exacerbated by the fact that the question of what counts as "trivial" depends, obviously, on the age or preparation of the learner.

Besides these problems that still pose significant challenges for the further development of an AAT, there are also a few problems that are already addressed by its current design. The first one is the tendency of argument assessors to stop the analysis of an argument right after a first problem has been spotted. The step-by-step guidance of the tutor motivates a more systematic and thorough approach. The second problem is the tendency to see quality issues everywhere. This is countered by the specifications for each criterion. They should help to develop a sense of what is really important.

Overall, the question whether the AAT model of the knowledge domain "argument assessment" is good enough to support learning is still an open question. The answer will depend on empirical studies, but also on some agreement in our community about the question whether the currently adopted normative standards are adequate, and how they could be improved.

5. CONCLUSION

Although this contribution presented already a certain design of an Argument Assessment Tutor—which answers at least a part of the second question that guided these efforts—the first question, whether such a tutor is possible at all as a tool for learning, remains unanswered. What is required, at this point, is a broader discussion within the community of argument theoreticians and informal logicians about the problems raised above, and then observations of its use and effects on learning.

Assuming that such research and deliberation does indeed lead to promising results, the next big step for the further development of an AAT would be to think about a more "intelligent" tutoring system. The literature on ITS makes a distinction between two or three levels on which these systems can be "adaptive" to learning: (1) on the level of the steps that are needed to complete a given task; (2) on the level of tasks, where the ITS is challenged to select the task that is most beneficial for a particular student; and (3) on the level of designing an ITS for a particular pedagogical challenge that has been identified across large numbers of students. The first level has been called "inner loop" or "step loop" because the tutor needs to be prepared to give feedback and hints on each step within a task; the second level "outer loop" or "task loop" (Vanlehn 2006); and the third level "design loop." As Aleven et al. (2017) write with regard to the latter:

> A system is adaptive at design time if it is designed in a way that is responsive to the learning demands that the domain produces that are largely the same for many learners (e.g., challenges or hurdles that are the same across learners). (p. 524)

The current AAT version is adaptive on the level of particular steps—because it responds differently in reaction to each step that a user takes when going through a task—and on the design level, because it has been designed in response to the need to provide student-tailored instruction to foster argument assessment skills. What is missing is adaptability on the task level.

At this point constraint-based modelling (CBM), on which the current AAT design is based, reaches its limits. As Fournier-Viger, Nkambou, & Nguifo (2010) stress, "one of the principal limitations of the CBM approach" is that it does "not support tutoring services such as to

… suggest the next steps to perform to the learner" (p. 86). In order to be able to select the task that is most helpful for a particular learner at a particular point in her development, it is necessary to keep track of individual student performance. This does not only require a "student model" but also the ability to track, over time, how well an individual student acquires each of the skills that are addressed by the tutor. The next goal is, thus, to develop a tutoring model that selects assessment tasks based on an analysis of individual learning needs.

ACKNOWLEDGEMENTS: This research has been supported by a grant from the National Science Foundation's "Cyberlearning and Future Learning Technologies" program (Award 1623419). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Aleven, V., McLaughlin, E. A., Glenn, R. A., & Koedinger, K. R. (2017). Instruction based on adaptive learning technologies. In R. E. Mayer & P. A. Alexander (Eds.), *Handbook of Research on Learning and Instruction* (2nd ed., pp. 522-559). New York: Routledge, Taylor & Francis Group.

Aleven, V., Sewall, J., Andres, J. M., Sottilare, R., Long, R., & Baker, R. (2018). Towards adapting to learners at scale: integrating MOOC and intelligent tutoring frameworks. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale* (pp. 14): ACM.

Corbett, A., McLaughlin, M., & Scarpinatto, K. C. (2000). Modeling student knowledge: Cognitive tutors in high school and college. *User modeling and user-adapted interaction, 10*(2-3), 81-108.

Diana, N., Stamper, J., & Koedinger, K. (2018). An Instructional Factors Analysis of an Online Logical Fallacy Tutoring System. In *International Conference on Artificial Intelligence in Education* (pp. 86-97): Springer.

El Saadawi, G. M., Tseytlin, E., Legowski, E., Jukic, D., Castine, M., Fine, J., . . . Crowley, R. S. (2008). A natural language intelligent tutoring system for training pathologists: implementation and evaluation. [journal article]. *Advances in Health Sciences Education, 13*(5), 709-722. doi: 10.1007/s10459-007-9081-3

Fournier-Viger, P., Nkambou, R., & Nguifo, E. M. (2010). Building intelligent tutoring systems for ill-defined domains. In R. Nkambou, J. Bourdeau & R. Mizoguchi (Eds.), *Advances in intelligent tutoring systems* (pp. 81-101). Berlin: Springer.

Gross, S., Mokbel, B., Hammer, B., & Pinkwart, N. (2015). Learning Feedback in Intelligent Tutoring Systems. *KI - Künstliche Intelligenz*. doi: 10.1007/s13218-015-0367-y

Hoffmann, M. H. G. (2018). The elusive notion of "argument quality". *Argumentation, 32*(2), 213–240. doi: 10.1007/s10503-017-9442-x

Hoffmann, M. H. G., & Lingle, J. A. (2015). Facilitating Problem-Based Learning by Means of Collaborative Argument Visualization Software. *Teaching Philosophy, 38*(4), 371-398. doi: 10.5840/teachphil2015112039

Johnson, R. H., & Blair, J. A. (2006 <1977>). *Logical self-defense*. New York: International Debate Education Association.

Kahan, D. M. (2013). Ideology, motivated reasoning, and cognitive reflection. *Judgment and Decision Making, 8*(4), 407-424.

Kulik, C.-L. C., Kulik, J. A., & Bangert-Drowns, R. L. (1990). Effectiveness of mastery learning programs: A meta-analysis. *Review of educational research, 60*(2), 265-299.

Lynch, C. F., Ashley, K. D., Aleven, V., & Pinkwart, N. (2006). Defining ill-defined domains; a literature survey. In *8th International conference on Intelligent Tutoring Systems. Proceedings of the First International Workshop on Intelligent Tutoring Systems for Ill-Defined Domains* (pp. 1–10), Jhongli, Taiwan.

Lynch, C. F., Ashley, K. D., Pinkwart, N., & Aleven, V. (2009). Concepts, Structures, and Goals: Redefining Ill-Definedness. *International Journal of Artificial Intelligence in Education, 19*(3).

National Center for Education Statistics. (2007). The Nation's report card: Writing 2007. National Assessment of Educational Progress at grades 8 and 12 (pp. 67 p.). Retrieved from http://nces.ed.gov/nationsreportcard/pdf/main2007/2008468.pdf

Nkambou, R., Bourdeau, J., & Mizoguchi, R. (Eds.). (2010). *Advances in intelligent tutoring systems*. Berlin: Springer.

Ohlsson, S. (1993). The interaction between knowledge and practice in the acquisition of cognitive skills *Foundations of knowledge acquisition: Cognitive models of complex learning* (pp. 147-208): Springer.

Ohlsson, S. (1994). Constraint-based student modeling. In J. E. Greer & G. I. McCalla (Eds.), *Student modelling: the key to individualized knowledge-based instruction* (pp. 167-189). Berlin, Heidelberg: Springer.

Ohlsson, S. (2016). Constraint-Based Modeling: From Cognitive Theory to Computer Tutoring – and Back Again. *International Journal of Artificial Intelligence in Education, 26*(1), 457-473. doi: 10.1007/s40593-015-0075-7

Peña-Ayala, A. (2013). *Intelligent and adaptive educational-learning systems : achievements and trends*. Heidelberg: Springer.

Pinkwart, N., Aleven, V., Ashley, K. D., & Lynch, C. F. (2006). Toward legal argument instruction with graph grammars and collaborative filtering techniques. In M. Ikeda, K. D. Ashley & T.-W. Chan (Eds.), *Intelligent Tutoring Systems, 8th International conference, ITS 2006, Jhongli, Taiwan. Proceedings* (pp. 227-236). Berlin: Springer.

Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences, 4*, 155-169.

Sleeman, D., & Brown, J. S. (1982). *Intelligent tutoring systems*. London ; New York: Academic Press.

Vanlehn, K. (2006). The behavior of tutoring systems. *International journal of artificial intelligence in education, 16*(3), 227-265.

Walkington, C. A. (2013). Using Adaptive Learning Technologies to Personalize Instruction to Student Interests: The Impact of Relevant Contexts on Performance and Learning Outcomes. *Journal of Educational Psychology, 105*(4), 932-945. doi: 10.1037/a0031882