2011

Cognitive Effects of Argument Visualization Tools

Michael H.G. Hoffmann, *Georgia Institute of Technology - Main Campus*

Cognitive effects of argument visualization tools

MICHAEL H.G. HOFFMANN

School of Public Policy, Philosophy Program
Georgia Institute of Technology
685 Cherry Street, N.W.
DM Smith Building
Atlanta, GA 30332-0345
U.S.A.
m.hoffmann@gatech.edu

ABSTRACT: External representations play a crucial role in learning. At the same time, cognitive load theory suggests that the possibility of learning depends on limited resources of the working memory and on cognitive load imposed by instructional design and representation tools. Both these observations motivate a critical look at Computer-Supported Argument Visualization (CSAV) tools that are supposed to facilitate learning. This paper uses cognitive load theory to compare the cognitive efficacy of Rationale™ 2 and AGORA.

KEYWORDS: AGORA, cognitive load theory, computer-supported argument visualization, critical thinking, learning, Rationale.

1. INTRODUCTION

More than a hundred years ago, Charles S. Peirce developed the concept of "diagrammatic reasoning" to explain how discovery and learning—especially in mathematics—is possible in a process of five steps: constructing diagrams, experimenting with them, observing the results, checking their generality, and expressing them in general terms (Peirce 1976: 47-48). More recent research on historical examples of scientific discoveries and detailed studies of individuals that document conceptual innovation and change as being enabled by "model-based reasoning" refined Peirce's ideas about diagrammatic reasoning and created an entire new research field at the intersection of cognitive science, philosophy of science, and computer science (Nersessian 2008; Glasgow, Narayanan, & Chandrasekaran 1995; Magnani, Carnielli, & Pizzi 2010; Magnani & Nersessian 2002; and the *Diagrams* conference series, Berlin: Springer).

The crucial role of external representations has also been documented in research on child development. Susan Carey argued that children at the age of 3 ½ years have already mastered an understanding of how the count list "one, two, three, four, five, etc." represent numbers (Carey 2009). But how do they get to this point? Carey provides a precise description of a process which is known in educational research rather metaphorically as "bootstrapping." Although it is obviously impossible to pull oneself up by one's own bootstraps, Carey describes a sequence of five steps in which the culturally provided count list plays a central role. The beginning of the list seems to be learned initially without any meaning like the rhyme "Eeny, meeny, miney, mo." However, based on the fact that the numbers one through four are already learned as unrelated quantifier words in normal language acquisition, the meaningless number list can be used in a "bootstrapping" process that allows the child finally to associate each item in the list with

Zenker, F. (ed.). *Argumentation: Cognition and Community. Proceedings of the 9th International Conference of the Ontario Society for the Study of Argumentation* (OSSA), May 18-21, 2011. Windsor, ON (CD ROM), pp. 1-12.

a specific model of a set so that the last number in the list signifies the cardinality of the set. This bootstrapping, however, is possible only when children engage with symbol systems whose initial meaning consists in nothing more than the fun of declaiming words in games. Elizabeth Spelke and her colleagues have shown how the same bootstrapping mechanism can help to explain how children learn "natural geometry." Like "the system of number, the system of geometry that feels most natural to educated adults is a hard-won cognitive achievement, constructed by children as they engage with the symbol systems of their culture" (Spelke, Lee, & Izard 2010).

However, not only culturally evolved symbol systems enable the child to learn by providing a "scaffold" for the construction of meaning. Another form of scaffolding can be observed in the fact that children master more elements of the count list when they are counting objects than when counting without anything observable (Hasemann 2003; Caluori 2004). The structure of the things outside obviously helps children to develop cognitive routines and schemas.

A scaffold, however, does not only enable development, it also constrains and limits it. This has been shown in studies that provide evidence for a disproportionate number of so-called "split-five errors" in early addition and subtraction. These are errors which consist in a wrong answer that differs by 5 from the correct result, as in 18 - 7 = 6 (Domahs, Krinzinger, & Willmes 2008). Since virtually all children use some kind of finger counting at a certain stage of their development, and since the above-average number of split-five errors can hardly be explained by anything else, we can assume that the specific structure of the human hand has a significant cognitive effect. Since a disproportionate number of split-five errors has been observed even after finger calculation is used as a cognitive tool, it can be assumed that engaging with the hands shapes—for better or for worse—the routines and strategies that the learner stores in long-term memory. A striking example of this phenomenon has been reported by Ian Thompson. A boy, Scott, explained his correct answer for 6 + 7 = 13 as follows: "I took 5 out of the 6 and 5 out of the 7 and I was left with 3" (Thompson 1999).

The cognitive power of external representations and objects—with regard to both enabling and constraining effects—is obviously relevant for the design of instructional materials and tools. We should design materials and tools that scaffold learning. They should enable bootstrapping without promoting cognitive schemas and routines that are irrelevant or that hamper further learning.

But how could that be achieved? We know that different representational notations that can be used to present the same problem—like a graph, a matrix, or a text—have different effects on learners' activities which justifies Daniel Suthers's concept of "representational guidance" (Suthers & Hundhausen 2003). But we also know that oftentimes high expectations—especially in the fast growing field of computer-supported learning—are disappointed. With regard to the efficacy of Computer Supported Argument Visualization (CSAV) tools, Carr reports that argument visualization did not improve legal argumentation by law students. Students using an argument visualization tool did not differ significantly from controls concerning the sophistication of their arguments, or their performance on the final exam (Carr 2003). Similarly, Bell observed that students introduced to the SenseMaker argument mapping software remained "generally confused about the argument map representation" and did not "develop any significant fluency with the tool" (Bell 2004).

Arguments are interesting from a cognitive point of view. An argument is obviously a complex entity. Following a common philosophical definition, an argument is a set of statements, including a claim and one or more reasons, in which the reasons jointly provide support (not necessarily conclusive) for the claim, or are at least intended to support the claim. That means, when we mentally represent an argument, we need to be able to engage simultaneously with a variety of things: with what is claimed in the various statements and with the relation, or a multitude of different relations, among these statements. A multitude of relations becomes relevant when we decide that a certain argument should better be divided into a chain or a network of connected arguments. Things are getting even more complex when we take into account that an arguer should also reflect on the quality of an argument.

Since arguments are cognitively demanding, I will use in the following, purely theoretical considerations Cognitive Load Theory to reflect on cognitive problems that Computer Supported Argument Visualization (CSAV) tools should address, and on possible solutions for these problems. The leading question is: How should CSAV tools be designed when the general goal is to develop tools that enable bootstrapping and efficient learning? More specifically, I will compare two very different tools that are both designed to foster critical thinking: "Rationale^{TM}" (version 2.0.8), a commercially available argument mapping software,[1] and the web-based "AGORA: Participate – Deliberate" system.[2] Rationale 2 can stand for a large variety of CSAV tools whose common feature is that they allow the visualization of "arguments" in the sense defined above, or even in a broader sense (including questions and ideas, for instance),[3] whereas AGORA seems to be the only CSAV tool that focuses only on deductively valid arguments.

I am interested here in the question of how effective both these CSAV tools can be for learning critical thinking given the following constraints: the cognitive complexity that we face when constructing arguments; the cognitive architecture that enables learning in humans; and natural limitations of the working memory. Since the last two constraints have been studied in research on cognitive load theory, this theory will be my starting point.

2. COGNITIVE LOAD THEORY

The core thesis of cognitive load theory is that the human working memory is limited: "We are unable to hold more than seven items of novel information in working memory (Miller 1956) and can probably process no more than four items (Cowan 2001)" (Sweller 2010: 37). These limitations are countered by a large long-term memory, for which no limitations are known, and a seemingly unlimited ability to process the content of the long-term memory without, or with only marginal, involvement of the working memory. It is assumed that information is stored in long-term memory in the form of cognitive schemas and that there are automated routines to process these schemas (Ericsson &

[1] See http://rationale.austhink.com/.

[2] See http://agora.gatech.edu/.

[3] See, for example, http://cohere.open.ac.uk/; http://carneades.berlios.de/; http://coala.gladisch.org/; http://araucaria.computing.dundee.ac.uk; http://www.argunet.org; http://www.athenasoft.org. For an overview see Scheuer, Loll, Pinkwart, & McLaren (2010).

Kintsch 1995). This means, problem solving can be done unconsciously, easily, and rapidly without much burden on working memory if cognitive schemas and routines that are necessary for processing are stored and automated in long-term memory (Sweller 2010).

According to cognitive load theory, "learning" is the process of forming these cognitive schemas and routines. However, biologically secondary learning, that is all learning of culturally dependent knowledge, presupposes instruction, most of which has to be processed consciously in working memory. Since instruction generally provides novel information, the learner has to deal with limitations of the working memory. He or she has to assimilate multiple information elements simultaneously to what is already given in long-term memory. John Sweller introduced the "borrowing and reorganizing principle" to describe how we acquire knowledge through instruction:

> Almost all information in long-term memory is obtained by imitating other people's actions or hearing or reading what others have said. … Nevertheless, the information borrowed is almost invariably altered and constructed. We do not remember exactly what we have heard or seen but, rather, construct a representation based on knowledge already held in long-term memory. (Sweller, 2010: 33)

This way, every instruction imposes a cognitive load on the working memory. Based on the fact that the capacity of the working memory is limited, it becomes crucial for the success of learning to organize instruction in a way that is cognitively effective.

In order to analyze and improve instruction, the theory distinguishes several forms of cognitive load: 1. "intrinsic" cognitive load which is determined by the material to be learned, in particular its so-called "element interactivity" which "refers to the number of elements that must be simultaneously processed in working memory to understand and learn material under instruction" (Sweller, 2010: 41); 2. "extraneous cognitive load" which is determined by the instructional design of a learning environment, but interferes with learning; and 3. "germane" or effective cognitive load that is also due to instructional design but enhances learning. Extraneous cognitive load results, for example, when a learner is confronted with a large number of interacting elements that are not related to learning, or when understanding new information requires significant changes in cognitive schemas that overstretch the conservatism of the long-term memory (Sweller 2010; Kalyuga 2010).

In the context of cognitive load theory, one of the most important functions of external representations is to "off-load" cognitive load (van Bruggen, Boshuizen, & Kirschner 2003; van Bruggen, Kirschner, & Jochems 2002). Since it should always be possible to cope with even the most complex task by breaking it down into components that can be dealt with in a sequence of steps, external representations can help to design such a sequence and to keep track of each step. Moreover, since element interactivity often refers to the influence that elements of a problem exert on each other, it helps to make those interactions explicit in external representations and to simulate what happens in externally visible form.

However, off-loading presupposes, first, that external representation systems are indeed capable to carry cognitive load and, second, that the user has already those cognitive schemas and routines available that are required for engaging effectively with external representations. For example, when we want to compare the price of two items in the grocery store that are packed in different sizes we can perform a calculation which is based on the rule of three (or proportion) on our phone. Doing so, however, presupposes two things: first, our calculator needs to be able to process our request based on the rules of the representational system that we choose in this case, that is arithmetic, and, second, we need to know the rule of three and how to apply it in the chosen system of representation. We can "off-load" to our phone the rule-based calculation, but not the routines that lead to our input.

Both the ability of representation systems to carry cognitive load and the ability of the learner to off-load certain cognitive schemas and routines mark the point where the problems with computer supported argument visualization tools begin.

3. COMPARING RATIONALE$^{\text{TM}}$ AND AGORA

Jan van Bruggen and his colleagues summarize the problem of CSAV tools as follows. Although argument visualization tools are supposed to reduce cognitive load by saving resources of the working memory, "a cognitive off-loading effect is not something that comes naturally with an external representation. On the contrary: characteristics of a representation seem to impose extraneous cognitive load and may lead to activities that do not foster deeper representation of the domain" (van Bruggen et al. 2003: 43). It is, however, probably impossible to assign specific cognitive loads—be it intrinsic, extraneous, or germane—to specific representations. On the one hand, it is well known that the actual cognitive load depends on the schemas and routines that a learner has already stored in long-term memory. "Depending on the schemas that have been acquired, material that is complex for one individual may be very simple for another. If a set of interacting elements have become incorporated into a schema, only that schema needs to be processed in working memory, not the interacting elements" (Sweller 2010: 41-42).

On the other hand, I think that any cognitive load that a particular representation either carries or represents (i.e., when it represents a situation in "the real world") depends on the learner's interpretation of this representation. That a "Diagram Is … Worth 10000 Words" (Larkin & Simon 1987) does not guarantee that every student sees the same 10000 words—or the ideas behind them—in a diagram. Like the first argument, this consideration implies that the actual cognitive load that a certain external representation imposes will always depend on the individual experiencing it.

It should be possible, however, to cope with these difficulties when we focus, first, on the beginning learner who—as we can assume—does not yet have the routines necessary to interpret and handle representations adequately and, second, when we focus on representation *systems*, not individual representations. Rationale$^{\text{TM}}$ and AGORA are "systems" of representation in so far as the architecture of the software determines what can be represented according to which rules and constrains. Under both these conditions, we can compare Rationale$^{\text{TM}}$ and AGORA with regard to the question of how both *systems* allocate cognitive load. This can be done on the level of the software architecture without empirical research—even though it would be best, of course, if these conceptual considerations could be supported by empirical evidence.

Both Rationale$^{\text{TM}}$ 2 and AGORA are argument mapping tools whose development is guided by the thesis that constructing arguments and argumentations in graphical form promotes critical thinking. In the case of Rationale's predecessor "Reason!Able," this thesis has been supported by several empirical studies that found considerable improvement of students' performance on the California Critical Thinking Skills Test (CCTST; cf. Twardy 2004; van Gelder, Bissett, & Cumming 2004). The main differences of the two software systems can be summarized as follows:

- AGORA guides the construction of arguments and argumentations by specific, system-generated prompts in a step-by-step process, whereas in Rationale^{TM} 2 the user is free to construct arguments and argumentations through templates and options which can be selected from a "Building Panel."
- Whereas the AGORA system generates automatically deductively valid arguments based on user input, the Rationale^{TM} 2 user can choose between two modes. In the "Reasoning" mode, "contentions" can be defended by simple reasons and debated by objections and rebuttals, and in the "Advanced Reasoning" mode, reasons for, and objections against, those contentions can be complemented by "co-premises" so that the construction of deductively valid arguments is possible.

In general, the AGORA system constrains the user's freedom more than Rationale^{TM} 2 does, but provides more guidance. Both systems are based on different educational philosophies. While Rationale^{TM} 2 is designed to represent arguments and argumentations in the biggest possible variety in order to come as close as possible to the way people actually argue in real live, AGORA attempts to familiarize users with a rather narrow, normative standard of argument construction in order to stimulate reflection and creativity; here it is more important to revise arguments as long as it takes to construct the strongest possible argument. This does not mean, however, that only a small sub-set of possible arguments can be represented in AGORA. It is well-known that almost all non-deductive arguments can be transformed into deductive ones by introducing an additional premise that simply states that the set of all reasons implies the conclusion. For example, the inductive argument: "All the ravens that I saw in the past were black, therefore all ravens are black" can be transformed into a *modus ponens* argument by adding the premise: "If all ravens I saw in the past were black, then all ravens are black." Of course, this additional premise is not really convincing and should be criticized. But that is exactly the point in the AGORA approach: Making those premises visible that *would* make conclusions necessarily true if all the premises were true helps the user to understand that most arguments should be revised to make them more plausible. Thus, the user should see that the argument above could be improved, for example, by saying:

- All the ravens that I saw in the past were black
- If all ravens I saw in the past were black, then probably all ravens are black
- Therefore, probably all ravens are black

This means that even though the expressional means are limited in the AGORA system by the fact that only seven argument schemes are available, the system is no less expressive than other argument visualization systems. The difference between the two systems is only that Rationale^{TM} 2 aims at representing everyday arguments directly, whereas the AGORA system challenges the user to participate in a reflective dialogue. After the user specifies reason and claim and selects an appropriate deductively valid argument scheme, the system creates automatically the missing premise. This premise again should stimulate a reflection on its acceptability and about possible objections, which again should lead to a revision of the formulations used as reason and claim.

Both Rationale^{TM} 2 and AGORA are designed on the assumption that students should learn a set of rules for the construction of arguments, and that they should be enabled to apply these rules correctly on a regular basis. The main function of these rules is to enable students to assess the quality of arguments. When they reconstruct arguments from learning material, or construct arguments on their own, they should have some criteria available to evaluate the quality of these arguments, that is the acceptability of reasons and the strength of support that the reason, or a combination of reasons, provides for the claim. Another important function of the rules of argument construction is to enable students to identify assumptions that are "hidden" in arguments. Often objections against an argument are not directed against the reasons that are explicitly stated, but against the inferential relation between reasons and conclusion; this relation, however, is mostly not explicit. Rationale$^{\text{TM}}$ 2 and AGORA formulate rules that enable students to visualize those hidden assumptions.

With regard to simple arguments, the Tutorial of Rationale$^{\text{TM}}$ 2 provides six rules:[4]

(1) The "Complete Sentence Rule": Use complete sentences, not words or phrases.
(2) The "Declarative Sentence Rule": Use declarative sentences, not other kinds of sentences such as questions.
(3) The "Golden Rule": Every simple argument has at least two co-premises (linked premises).
(4) The "Rabbit Rule": Any significant term or concept which appears in the contention must also appear in one of the premises.
(5) The "Holding Hands Rule": if something appears in a premise but not in the contention, it must appear in another premise.
(6) The "No-Danglers Principle": No significant word, term, phrase or concept should hang loose; it needs to appear in two different statements of the argument.

The "Rabbit Rule"—which is based on the idea "that you can't pull rabbits out of hats just by magic"—and the "Holding Hands Rule" are central because they imply the "Golden Rule" and the "No-Danglers Principle," as can be seen in Figure 1 which is from the Rationale Tutorial.

AGORA, in contrast, focuses on five rules that students are supposed to learn and to apply:

(1) Represent your main argument—and every sub-argument that might be controversial—in a form that realizes a deductively valid argument scheme.
(2) Given that an argument is a set of statements that contains one claim and at least one reason, each of these statements should be formulated as one complete sentence and should be placed in one text box.
(3) Only descriptive or normative statements, or those that combine both, are permitted in an argument.
(4) Reflect on the acceptability of all your premises, and provide further arguments for those whose acceptability is either not evident or controversial.
(5) Revise the formulations of your statements and the structure of your argument as long as it takes to create the best possible argument.

[4] See http://austhink.com/reason/tutorials.

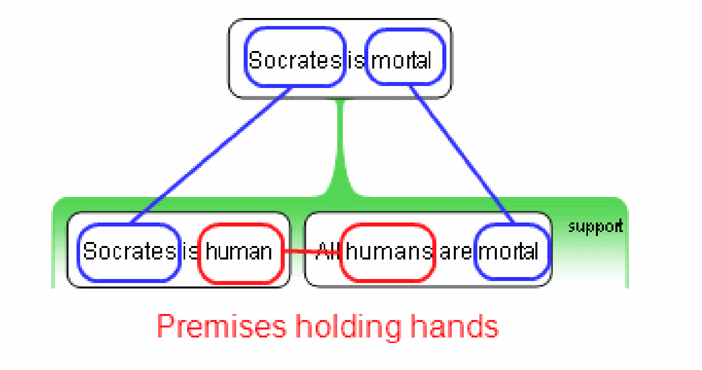


Fig. 1.An example for a correct application of Rationale's "Rabbit Rule" (marked in blue) and its "Holding Hands Rule" (marked in red).

While both sets of rules overlap with regard to the need to provide "well-formed formulas" in an argument, there are two obvious differences: first, the AGORA rules include rules that are intended to motivate reflection on the acceptability of premises and ongoing improvement of the argument; and second, realizing the first rule of the AGORA system implies, necessarily, the realization of Rationale rules 3 to 5, but not *vice versa*.

Less obvious differences, however, concern the distribution of cognitive load in both systems:

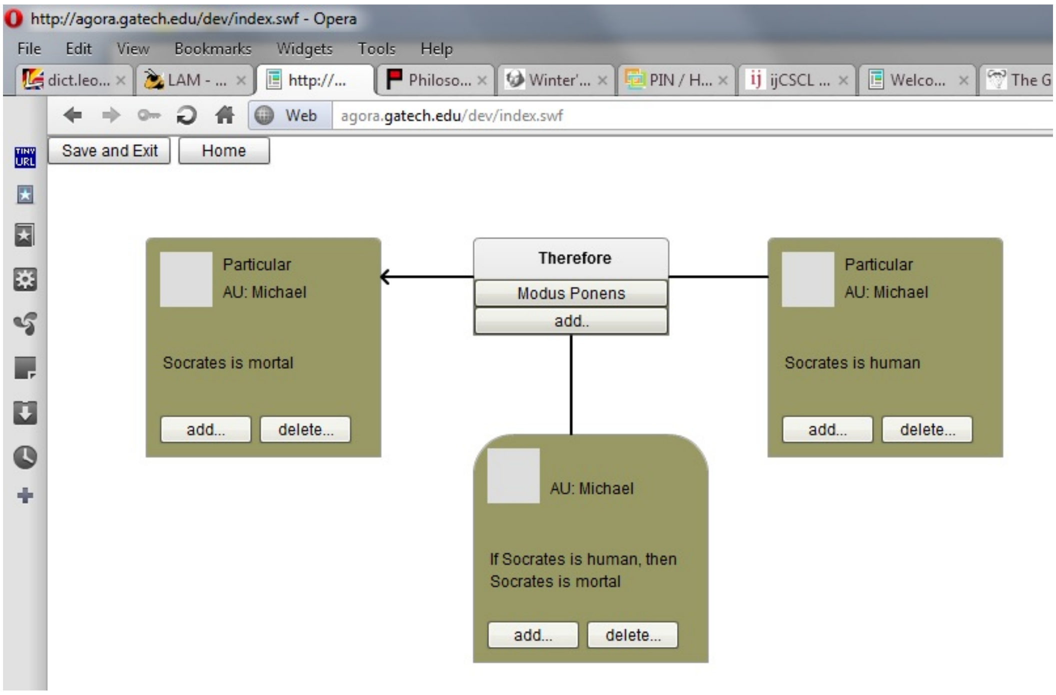
- The user of Rationale^{TM} 2 has to learn the rules of argument construction in advance and by instruction, whereas the central first rule of AGORA—which implies the central Rationale rules 3 to 5—is implemented in the software architecture.[5]
- The Rationale^{TM} 2 user has to develop a strategy of how to realize the six rules when constructing an argument while the AGORA software offers two possible strategies that allow the user to break down any task in a sequence of steps, and that guides him or her step by step to realize the first AGORA rule of argument construction automatically.

This works as follows in the AGORA system.[6] Every time the user wants to create or add a new argument, she is prompted to choose one of two strategies by either selecting "What is the main claim of your argument?" or "Click here if you want to use a specific argument scheme." If she selects the former, a text box appears and she is prompted to enter her claim. After she clicks "Done," a new text box appears which is connected to the first one by a directed arrow and the connector "therefore." This new text box prompts her to enter a reason. After she has done this, she will be prompted to choose an argument scheme from a list. Each scheme from the list is connected to another list which shows several English formulations for the missing premise which are all logically equiv-

[5] The sixth Rationale rule is also automatically realized in the AGORA software. Whereas the monotonicity of standard deductive logic would allow "danglers," AGORA realizes a non-monotonic and defeasible form of deductive logic.

[6] See the prototype at http://agora.gatech.edu/. The AGORA system is not yet completed, but its complete functionality is described in "Logical Argument Mapping (LAM): A Manual" at http://lam.spp.gatech.edu/.

alent. (For *modus ponens,* for example, the list includes “if p then q,” “p implies q,” “p only if q,” “p is a sufficient condition for q,” etc.). When she selects one of these English formulations, the software creates automatically a deductively valid argument on her screen. Figure 2 shows how the Rationale 2 argument presented in Figure 1 looks like in AGORA.



Particular
AU: Michael
Socrates is mortal

Particular
AU: Michael
Socrates is human

AU: Michael
If Socrates is human, then Socrates is mortal

Fig. 2.The AGORA version of the Rationale 2 argument in Figure 1.[7]

The system-generated argument in Figure 2 makes clear that the Rationale rules 3 to 6 are automatically realized by the AGORA software. Each modification of a formulation in one of the text boxes and each change of the argument scheme preserves the argument’s deductive validity and, thus, maintains conformity to the rules. In Rationale^{TM} 2, by contrast, following these rules requires, first, that the user selects the “Advanced Reasoning” mode (because only in this mode co-premises can be added), second that she adds indeed co-premises to her reasons or selects a template in which text boxes for co-premises are provided, and third that she formulates the statements for the contention, the reason, and the co-premise in a way that satisfies the rules.

Based on the different software architectures that are provided in Rationale^{TM} 2 and AGORA, the following two arguments can be made. First, in the process of learning critical thinking, working with Rationale^{TM} 2 imposes a higher cognitive load on learners’ working memory than working with AGORA. That is, a learner who does not already have stored the schemas and routines in long-term memory that are necessary to construct arguments according to the six Rationale rules or the five AGORA rules, will face a

[7] The automatically created premise at the bottom corresponds to the premise “All humans are mortal” in the previous figure, but is less general. In the completed version of AGORA the user can apply “generalization” to the argument to transform this premise into “if something is human, then it is mortal.”

higher cognitive load in Rationale™ 2 than in AGORA. The reasons for this first claim can be summarized as follows.

(1) Every argument that satisfies the first AGORA rule or Rationale rules 3 to 6 poses an intrinsic cognitive load of at least three interacting elements because at least three statements need to be formulated so that they satisfy what is called in Rationale™ the Rabbit Rule and the Holding Hands Rule.

(2) In addition to this intrinsic cognitive load, the six Rationale rules pose germane cognitive load on the working memory of the learner (because they support the learning of how to construct good arguments). In Rationale™ 2, however, the learner, who does not yet have stored the rules in long-term memory and automated their application, has to deal with the intrinsic load of the argument and the germane load of the six rules simultaneously. Although the software provides a template that distinguishes clearly between "contention," "reason," and "co-premise," so that off-loading the cognitive load connected with Rationale rule 3 is possible (the "Golden Rule" about the necessity of co-premises), the application of the five remaining rules is neither supported by the software, nor does the software provide any feedback when these rules are violated.

(3) In AGORA, by contrast, a good deal of the cognitive load that a learner has to handle to construct a good argument is off-loaded to the software. First, the two strategies provided by the software allow a clear distinction of several tasks that are rationally connected. Since user input, in the sequence of steps described above, is prompted for each task individually, the learner is confronted with only one component of intrinsic load at a time. (For the selection of an argument scheme, which is cognitively more demanding, the beginner would simply play around with the schemes offered to see how they work.) Second, the central first rule of argument construction is automatically realized by the software; there is neither a need to understand this rule nor to keep it in mind for possible applications. Third, I would assume that anything that is created by the software challenges the learner immediately to reflect on its acceptability, and that this reflection would lead—if the acceptability is problematic—either to further arguments or a revision of what has been constructed so far (realizing thus AGORA rules 4 and 5).

My second argument refers to the possibilities of extraneous cognitive load on the one hand and bootstrapping on the other. Whereas the software architecture of the AGORA system guarantees that the user formulates deductively valid arguments, and whereas the validity of the seven argument schemes that will be implemented in the AGORA system[8] is evident and can be proven, the Rabbit Rule and the Holding Hands Rule in Rationale™ are more complicated than it might seem. We saw already that both rules are fulfilled by the famous syllogism about Socrates' mortality in Figure 1. But both rules are also fulfilled by the following arguments:

[8] *Modus ponens*; *modus tollens*; disjunctive syllogism; not-both syllogism; equivalence; conditional syllogism; and constructive dilemma. See http://tinyurl.com/54qkq4.

(1) Socrates is mortal, (supported by) Socrates appears in Plato's *Republic* and (co-premise) some of those who appear in Plato's *Republic* are mortal.

(2) Socrates is mortal, (supported by) Socrates' fame guarantees immortality and (co-premise) all fame that guarantees immortality is mortal.

These examples demonstrate that arguments of different (or dubious) quality can be constructed which nevertheless fulfill the Rabbit Rule and the Holding Hands Rule. Since the function of these rules is to enable students to evaluate the quality of arguments, and since Rationale^{TM} does not provide any rule that could be used to rank the quality of the three arguments about Socrates' mortality, it seems to be clear that the Rationale rules are not sufficient to assess the quality of arguments. This poses a problem for learning, because we can easily imagine that students come up with alternatives like these. If so, they might start to reflect on the adequacy of the rules of argument construction, or they might be getting confused, instead of learning how to construct a good argument.

While there seems to be a risk of extraneous cognitive load that results from Rationale's rules, and while it is possible to use the Rationale^{TM} 2 software without ever learning the rules, or learning how to apply them, I would assume that the AGORA approach allows learning by bootstrapping. Since the first and central AGORA rule is automatically realized by the software, the user can learn its underlying concept of deductive validity simply by constructing arguments. In Rationale^{TM} 2, the same effect could only be achieved by a lot of instruction.

4. CONCLUSION

Even though it will be hard to determine the cognitive load imposed by concrete, external representations, Cognitive Load Theory can be used to compare the respective load that different argument visualization *systems* impose on the beginning learner. Especially for learners who do not know anything about what an argument *is*, its structure, and its quality, it might be better to start with a system that provides more guidance and that allows the bootstrapping of those skills that can then be used for a wider range of argument types.

REFERENCES

Bell, P. (2004). Promoting Students' Argument Construction and Collaberative Debate in the Science Classroom. In: Linn, M.C., Davis, E.A., and Bell, P. (eds). *Internet environments for science education* (pp. 115-143). Mahwah, NJ: Erlbaum.

Caluori, F. (2004). *Die numerische Kompetenz von Vorschulkindern*. Hamburg: Kovac.

Carey, S. (2009). Where our Number Concepts Come From. *Journal of Philosophy* 106(4), 220-254.

Carr, C.S. (2003). Using computer supported argument visualization to teach legal argumentation. In:. Kirschner, P.A, Buckingham Shum, S.J., and Carr, C.S. (eds). *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-making* (pp. 75-96). London: Springer.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences* 24, 87-114.

Domahs, F., Krinzinger, H., & Willmes, K. (2008). Mind the gap between both hands: Evidence for internal finger-based number representations in children's mental calculation. Cortex, 44(4), 359-367.

Ericsson, K.A., and Kintsch, W. (1995). Long-term Working Memory. *Psychological Review* 102(2), 211-245.

Glasgow, J., Narayanan, N. H., and Chandrasekaran, B. (eds). (1995). *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. AAAI Press.

Hasemann, K. (2003). *Anfangsunterricht Mathematik*. Heidelberg: Spektrum.

Kalyuga, S. (2010). Schema Acquisition and Sources of Cognitive Load. In: Plass, J.L., Moreno, R. and Brünken, R. (eds). *Cognitive Load Theory* (pp. 48-64). Cambridge; N.Y.: Cambridge University Press.

Larkin, J.H., and Simon, H.A. (1987). Why a Diagram Is (Sometimes) Worth 10000 Words. *Cognitive Science* 11(1), 65-99.

Magnani, L., Carnielli, W., and Pizzi, C. (eds). (2010). Model-Based Reasoning in Science and Technology. Abduction, Logic, and Computational Discovery.

Magnani, L., and Nersessian, N.J. (eds). (2002). *Model-Based Reasoning: Science, Technology, Values*. New York: Kluwer Academic/Plenum Publisher.

Miller, G.A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review* 93, 181-186.

Nersessian, N.J. (2008). *Creating scientific concepts*. Cambridge, Mass.: MIT Press.

Peirce, C.S. (1976). The Carnegie application (1902). In: Eisele, C. (ed.). *The New Elements of Mathematics by Charles S. Peirce* (Vol. IV, pp. 13-73). The Hague-Paris/Atlantic Highlands, N.J.: Mouton/Humanities Press.

Scheuer, O., Loll, F., Pinkwart, N., and McLaren, B.M. (2010). Computer-Supported Argumentation: A Review of the state of the art. *International Journal of Computer-Supported Collaborative Learning,* 5(1), 43-102.

Spelke, E., Lee, S.A., and Izard, V. (2010). Beyond Core Knowledge: Natural Geometry. [Review]. *Cognitive Science* 34(5), 863-884.

Suthers, D.D., and Hundhausen, C.D. (2003). An Experimental Study of the Effects of Representational Guidance on Collaborative Learning Processes. *Journal of the Learning Sciences* 12(2), 183-218.

Sweller, J. (2010). Cognitive Load Theory: Recent Theoretical Advances. In: Plass, J.L., Moreno, R, and Brünken, R. (eds). *Cognitive Load Theory* (pp. 29-47). Cambridge; NY: Cambridge University Press.

Thompson, I. (1999). Mental calculation strategies for addition and subtraction. *Mathematics in School,* 28(5), 1-4.

Twardy, C. R. (2004). Argument Maps Improve Critical Thinking. *Teaching Philosophy* 27(2), 95-116.

van Bruggen, J.M., Boshuizen, H.P.A., and Kirschner, P.A. (2003). A Cognitive Framework for Cooperative Problem Solving with Argument Visualization. In:. Kirschner, P.A, Buckingham Shum, S. J., and Carr, C.S. (eds.). *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-making* (pp. 25-47). London: Springer.

van Bruggen, J.M., Kirschner, P.A., & Jochems, W. (2002). External representation of argumentation in CSCL and the management of cognitive load. *Learning and Instruction* 12(1), 121-138.

van Gelder, T., Bissett, M., & Cumming, G. (2004). Cultivating expertise in informal reasoning. *Canadian Journal of Experimental Psychology-Revue Canadienne De Psychologie Experimentale* 58(2), 142-152.