The Carneades model of argument invention

Douglas Walton and Thomas F. Gordon
University of Windsor, Canada / Fraunhofer Institute for Open Communications Systems (FOKUS), Berlin

Argument invention is a method that can be used to help an arguer find arguments that could be used to prove a claim he needs to defend. The aim of this paper is to show how argumentation systems recently developed in artificial intelligence can be applied to the task of argument invention. One such system called Carneades is featured. Carneades can be used to analyze arguments, evaluate arguments, to make an argument diagram, and to construct arguments from a database. Using some simple examples, the paper explains how Carneades works as a system of argument invention.

Keywords: Argument construction, argumentation schemes, argument visualization, artificial intelligence, finding an argument, proof standards, rhetorical argumentation

1. Introduction

The idea of building some method that could be useful to help an arguer find arguments that could be used to prove a claim he needs to defend was held to be important by the ancient Greek philosophers and rhetoricians (Kienpointner 1997). Aristotle and Cicero had systematic methods for helping a speaker find arguments to support a claim, but the technology to develop and implement a scientific method of argument invention was not available. Now that has changed. Artificial intelligence has recently experienced a rapid growth in argumentation technology. We now have multi-agent negotiation systems, argumentation-based models of evidential structures in legal reasoning, decision support systems using argumentation, and models of knowledge engineering structured around core concepts of argument to simplify knowledge elicitation. We also have automated argumentation systems used in computer-assisted collaborative learning, and single-user systems designed with the purpose of allowing a user to visualize, analyze, and evaluate an argument in a given text. Hence the infrastructure is there for

Pragmatics & Cognition 20:1 (2012), **1–31**. DOI 10.1075/pc.20.1.01wal
ISSN 0929–0907 / E-ISSN 1569–9943

the project of computational implementation of a working system of automated argument invention. This paper shows how close we are, and how some existing systems of argument mapping already have capabilities for carrying out limited tasks of argument invention. The Carneades Argumentation System, named after the Greek skeptical philosopher Carneades, is the first argument mapping tool with an integrated inference engine for constructing arguments from knowledge-bases. The Carneades system is a mathematical model of argumentation (Gordon and Walton 2006) that has a syntactical structure for representing and encoding arguments (Walton and Gordon 2005). It has also been implemented using a functional programming language and has an Open Source argument mapping graphical user interface that is available at no cost to users, and has a software library for building applications supporting other argumentation tasks.[1]

The version that presently exists can be used to analyze, construct, and evaluate arguments using defeasible forms of argument like argument from testimony, argument from analogy, argument from precedent, practical reasoning, and many other kinds of arguments (Gordon 2010).

The purpose of this paper is to explain with the aid of examples how Carneades works as a system of argument invention, and thereby show anyone interested in argumentation what the basic components of a computational system of argument construction should be, how they should be put together in a system, and how they can be applied to tasks where argument invention is a useful tool. Section 2 provides the needed background by briefly sketching the history of traditional attempts to build a method of argument invention, and by briefly introducing the reader to some of the recent software systems for argumentation that can be used to analyze and evaluate arguments, and have some capabilities for argument invention. The purpose of this paper is to show to anyone interested in argumentation how such a system works. Section 3 outlines the Carneades Argumentation System as a formal model. Section 4 clarifies the role of the audience in the model. Section 5 provides an example, and Section 6 explains more generally how Carneades works by applying it to this example and another one. Section 7 shows how the basic components of Carneades are put together to provide a working system of argument invention. Section 8 shows how the system is applied to the two arguments previously considered and Section 9 applies it to a legal example. A conclusion to the paper is provided in Section 10, briefly summarizing in general outline how the process of argument invention is carried out by Carneades.

2. Background

In traditional rhetoric, argument invention is the method for finding arguments that can be used to help an arguer prove some thesis he needs to prove in order to persuade an audience (Hohmann 2000). Aristotle in the *Topics* identified a great number of commonly used types of arguments he called topics (*topoi*, or places). More topics can be found in Aristotle's *Rhetoric*. Kienpointner (1997: 227) remarked that between three hundred and four hundred of them can be found. Bird (1962: 310) cites two scholars who enumerated the topics, one giving a count of 287, and the other a count of 382. Examples of topics are argument from witness testimony, appeal to authority, argument from justice (fairness), argument from analogy, argument from pathos, arguments from definitions, and arguments from consequences of actions. What these topics represent and what they are supposed to be used for has been controversial over the centuries. A comprehensive account of the history of topics, including a discussion of their relationship to argumentation schemes, can be found in Walton, Reed, and Macagno (2008: Chap. 8). Some have taken a topic to have a warranting function that enables an inference to be drawn from a set of premises to a conclusion. Others have taken a topic as a device to help an arguer search for an argument that could be used to support a claim he wants to prove (Kienpointner 1997). Viewed in this way, topics appear to be part of a system of argument invention. They appear to be very similar to, if not the same as, what are now called defeasible argumentation schemes (Walton, Reed, and Macagno 2008: 280). It is an interesting question for specialists in the history of rhetoric in Greek philosophy to determine what the relationship is between the ancient topics and modern argumentation schemes. The historical survey of the subject (Ibid.) covers the older history of the topics and modern theories of argumentation schemes, including those of Hastings (1963), Perelman and Olbrechts-Tyteca (1969), and Grennan (1997).

Another important notion of argument invention is that of stasis. The stasis system found in ancient rhetorical manuals is generally attributed to Hermagoras of Temnos (Hohmann 1989: 172), a teacher of rhetoric who lived in the second century B.C., and wrote a handbook (now lost) presenting a system of argumentation (Kennedy 1963: 303). An account of the system given by Quintilian tells us that the stasis in Greek (or status, in Latin) is the ultimate thesis (claim) that a sequence of argumentation is directed towards proving (Leff 1996). The motivating idea is that a sequence of argumentation in a speech has some unsettled issue it is meant to resolve, and the arguments in the speech should all be directed towards the resolution of this issue. For example, in a case where a defendant has been charged with theft, his accuser is claiming that he stole some item, and has the burden of proving this thesis. The defendant wins if the accuser fails to prove his claim.

Cicero built a system of rhetorical invention in *De Inventione* using topics as parts of a method of finding arguments for and against a contested claim (Leff 1983). The system emphasized the use of logical reasoning for the purpose of presenting and attacking arguments used to argue on both sides of a claim. This notion brings into play the requirement that argumentation used in a speech is only relevant if it provides reasons for or against the opposed propositions that make up the central issue of a dispute. To find relevant arguments in such a setting, for example in a court of law, Cicero advocated the use of topics representing different kinds of commonly used arguments that can carry weight as evidence in persuading a judge or audience to accept or reject an argument. One of the examples of argument invention studied later in this paper is a legal case of this sort.

Stasis theory and topics have provided key concepts that have proved to be useful for getting the outlines of a methodology for the field of rhetoric, a field that is centrally concerned with choosing among the arguments to decide which ones would be more persuasive, given information about the particular audience to be persuaded. Argumentation methods, like the use of argument diagramming tools, the deployment of argumentation schemes, and the building of formal models of argumentation in artificial intelligence, have now advanced to the point where they are useful for assisting with argument invention tasks. However, these systems (except for the one presented in this paper) have only indirectly addressed the problem of argument invention. Scheuer et al. (2010) have surveyed the state of the art on the development of software tools to support argumentation and to teach argument skills, and have provided a comprehensive account of what such systems are supposed to do and what their main features are.[2]

One of the significant differences between systems is whether collaboration between users, or even among a group of users, is supported (Scheuer et al. 2010: 57). Of the systems they discuss, they classify the following ones as offering built-in support for collaboration: Belvedere, gIBIS, QuestMap, Compendium, Digalo, AcademicTalk, InterLoc, DebateGraph, and Collaboratorium. They classify others as single-user systems, including Reason!Able, Rationale, Athena, Carneades, ArguMed, LARGO, SenseMaker, and Convince Me.

These argumentation systems are centrally concerned with the retrospective analysis of existing arguments by helping the user to construct an argument diagram that represents the structure of the original argument found in a text of discourse by representing its premises and conclusions, as well as other components like implicit premises. However, carrying out this retrospective task can help the user in the task of the construction of valid arguments by enabling him to find flaws in an argument that can be corrected. Additional mental demands of diagram creation may lead to more rigorous and well-conceived arguments, because strengths and weaknesses are easier to see (Buckingham Shum et al. 1997).

For example, although Araucaria is meant to analyze and evaluate argumentation found in a text of discourse cut and pasted into a text document, it can also be shown how it is useful for certain kinds of argument invention tasks. Araucaria has a menu of argumentation schemes, and when a user applies one of these schemes to a particular argument found in a text, the menu used for this purpose presents the list of appropriate critical questions matching that argumentation scheme. Critical questions identify weak points in argument, and therefore once the critical question is identified, it may suggest counter-arguments that could be used to attack the original argument. So for purposes of argument refutation, Araucaria can at least to some extent be helpful for argument invention. It can use critical questions to provide guideposts for inventing counter-arguments that can be used to attack a given argument.

To examine a simple example for purposes of illustration, let's consider the following argument: Bill is an expert in climate science; Bill says that it is doubtful that climate change is caused by carbon emissions; therefore it is doubtful that climate change is caused by carbon emissions. This argument fits the argumentation scheme for argument from expert opinion. When the text of the argument is inserted into Araucaria, and the scheme for argument from expert opinion is selected from the menu and applied to the argument, the argument diagram shown in Figure 1 can be produced.

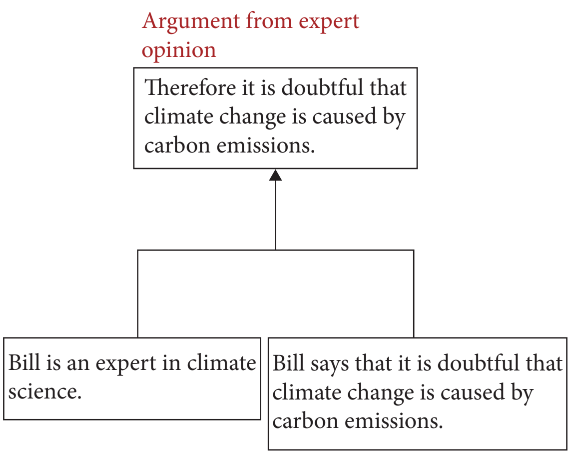

Figure 1. Araucaria Diagram for Expert Opinion Argument

The conclusion is shown at the top, and two premises are shown below in a linked argument configuration in which both premises are required to support the conclusion. The name of the argumentation scheme is shown above the conclusion, at the top of Figure 1, and the argument it applies to is shown by the shaded area around it.

In order to construct this diagram with Araucaria, one has to use a menu to select this particular argumentation scheme and apply it to the argument one is analyzing. In addition to representing the argument itself, this menu also displays a set of critical questions matching the argumentation scheme, in this instance the scheme for argument from expert opinion. This menu is shown in Figure 2.

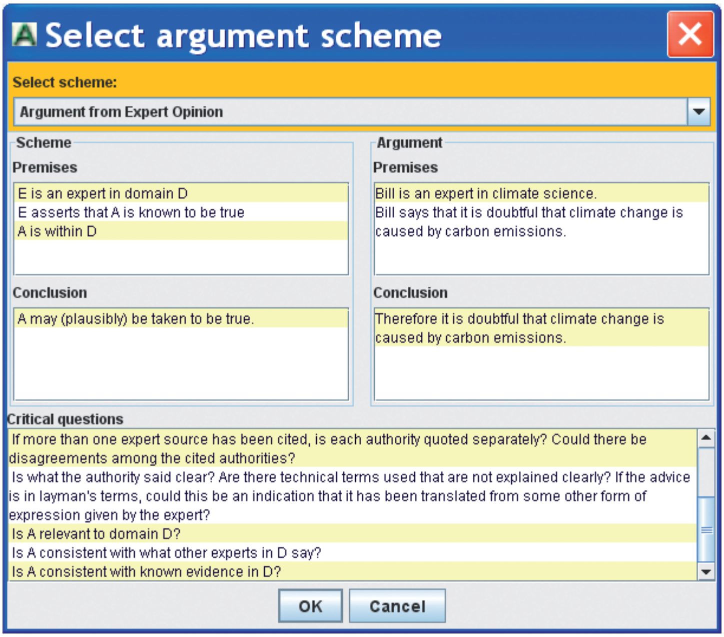

Figure 2. Araucaria Menu Showing Scheme and Critical Questions

Notice that one of the critical questions asks whether the statement asserted by the expert is consistent with what other experts in the same field of knowledge say. The asking of this critical question could elicit a statement that is common knowledge, namely the statement that most experts in climate science would disagree with the claim that it is doubtful that climate science is caused by carbon emissions. In other words, the evocation of this critical question might suggest a way of attacking the original argument by raising questions about whether what the expert has said represents a consensus in the field or not. Similarly, other critical questions, by suggesting applicable counterarguments, perform a limited function of argument invention.

It is reasonable to say that these systems can help a user to create a better argument. Scheuer et al. (2010: 55) rightly claim that "depending on intended purpose, different systems offer different modes of argument creation". However, the meaning of 'argument creation' in their sense falls short of argument invention in the fuller sense of systematically searching for new arguments an arguer could use to prove the central claim he has made in a dispute. This capability is present in a limited way in any of the systems that contain a set of argumentation schemes, like Araucaria or Reason!Able, because the critical questions attached to each scheme can be used to pinpoint weaknesses in the argument, suggesting objections that could be responded to with new arguments that the arguer may not anticipate prior to the asking of the question. But this capability is only a limited form of argument invention in the fuller meaning of the term proposed here. Carneades goes well beyond this limited capability by providing computational tools specifically designed to systematically search for and find new arguments that can be used to prove an arguer's claim.

3. The Carneades argumentation system

Carneades is a mathematical model of argumentation and a computational model (Gordon, Prakken, and Walton 2007) that defines mathematical properties of arguments and models the structure and applicability of arguments, the acceptability of statements, burdens of proof, and proof standards. Carneades has been implemented using a functional programming language, and has a graphical user interface currently under development as an Open Source software project freely available to users at the site http://carneades.github.com/ The current version (1.0) applies proof standards, manages the distribution of the burden of proof in argumentation, and can be used both to evaluate given arguments and to construct arguments to prove a claim. The initial work done in the project so far has built a working prototype of an Open Source argument mapping and visualization tool. This tool is comparable to the Araucaria system (Reed and Rowe 2005), and the numerous comparable systems of computer-supported argumentation surveyed by Scheuer et al. (2010), but the special feature of it explained and discussed in this paper is its capability for assisting with the task of argument invention.

In the next section we will provide an example showing how Carneades works, but in this section we outline Carneades as a formal model. We begin by formally defining argumentation schemes and the process of constructing arguments by instantiating schemes.

Definition (argumentation scheme). Let L be a higher-order language consisting of sets of predicate symbols, function symbols and constants, as well as a set of operators for recursively constructing terms and propositions from these symbols. Let V be a set of variables ranging over both formulas and terms. An **argumentation**

scheme is a tuple ⟨P, E, A, c⟩ where P ∈ L are its **premises**, E ∈ L are its **exceptions**, A are its **assumptions** and c ∈ L is its **conclusion**. All members of P, E and A must be literals, i.e., either an atomic proposition or a negated atomic proposition of the language L. The literals in P, E, and A may contain variables. The conclusion c may be a variable, ranging over an atomic proposition, or a literal which may contain variables.

As described in Gordon, Prakken, and Walton (2007: 884), exceptions and assumptions model the critical questions of argumentation schemes. They differ in how they distribute the burden of proof, more precisely the burden of persuasion, among the proponent and respondent. Intuitively, the respondent has the burden of proving exceptions, while the proponent must prove assumptions, just like ordinary premises, after the assumption has been questioned by the respondent.

Definition (argument). An **argument** is a tuple ⟨P, c⟩ where P ∈ L are its **premises** and c ∈ L is its **conclusion**. The conclusion c and all members of P must be ground literals. Let p be a ground literal. If p is the conclusion c of an argument, then the argument is an argument **pro** p. If the conclusion of the argument is the complement of p, then it is an argument **con** p.

Definition (stage of dialogue). A **stage** of dialogue is a tuple ⟨*arguments*, *status*⟩, where *arguments* is a set of arguments and *status* is a partial function mapping literals in the premises or conclusions of the arguments to their **dialectical status** in the stage, where the status is a member of {stated, assumed, questioned}. In every chain of arguments, $a_1, \ldots a_n$, constructible from *arguments* by linking the conclusion of an argument to a premise of another argument, a conclusion of an argument a_i may not be a premise of an argument a_j, if $j < i$. A set of arguments which violates this condition is said to contain a **cycle** and a set of arguments which complies with this condition is called **cycle-free**.[3] .

Definition (Instantiation of Argumentation Schemes). Let σ be a map of **substitutions** from variables to constants in L and let ⟨P,E,A,c⟩ be an argumentation scheme. By systematically substituting the variables in the argumentation scheme by their values in σ, the following arguments are constructed:

- ⟨σP ∪ σA, σc⟩, where σP denotes the term or proposition resulting from substituting variables in P with their values in σ. Let us call this argument a_1.
- For each exception in $e_1 \ldots e_n$ in E, a further argument is constructed, ⟨{e_i}, undercut(a_1)⟩, where “undercut” is assumed to be a predicate symbol in every language L.

In practice, arguments constructed from schemes are reified by assigning them identifiers and using these identifiers as constants to denote arguments in propositions about undercutters.

Essentially, in this model we use exceptions to interpret the critical questions of a scheme that place the burden of proof on the respondent as enumerating *some* ways to undercut arguments constructed using the scheme. This does not preclude other ways of undercutting the argument. The exceptions are not assumed to exhaustively list all possible ways to undercut the argument.

Definition (Putting Forward Arguments from Schemes). When arguments constructed by instantiating a scheme are put forward in a dialogue, the stage of dialogue ⟨*arguments*, *status*⟩ is updated by adding the resulting arguments to *arguments* and changing the status of the assumptions σA in the argument ⟨σP ∪ σA, σc⟩ as follows:

- If a literal in σA is undefined or stated, then the literal is assigned the status "assumed" in the successor stage of dialogue.
- If a literal in σA is questioned in the stage, then it remains questioned in the successor stage of dialogue.

The intention of this model is for assumptions of arguments to be assumed only if and so long as they have not been questioned. Once assumptions have been questioned, they work as ordinary premises.

Next we define a structure for evaluating arguments, to assess the acceptability of propositions at issue. As in value-based argumentation frameworks (Bench-Capon and Sartor 2003; Bench-Capon, Doutre, and Dunne 2007) arguments are evaluated with respect to an **audience**, such as the trier-of-fact (judge or jury) in legal trials.

Definition (audience). An **audience** is represented by structure ⟨*facts*, *weight*⟩, where *facts* is a consistent set of literals accepted by the audience as being true, and *weight* is a partial function mapping arguments to real numbers in the range 0.0 … 1.0, representing relative weights assigned by the audience to the arguments.

This model of audiences is an abstraction useful for various purposes. It can represent information collected from a real audience, for example via interviews or surveys. But it can also represent assumptions made by an arguer during a dialogue about the audience, where these assumptions may be based on less than certain evidence. For example, evaluating arguments based on assumptions about the audience can be useful for helping an arguer to clarify which parts of his case need strengthening with further arguments. More clarification of the notion of audience in argumentation is given in Section 4.

Arguments in a stage of dialogue are evaluated relative to an audience in an **argument evaluation structure**.

Definition (argument evaluation structure). An **argument evaluation structure** is a tuple ⟨*stage*, *audience*, *standard*⟩, where *stage* is a stage of dialogue, *audience* is an audience and *standard* is a total function mapping atomic propositions to their applicable proof standards in the dialogue. A **proof standard** is a function mapping tuples of the form ⟨*issue*, *stage*, *audience*⟩ to Boolean values (true and false), where the *issue* is an atomic proposition.

Given an argument evaluation structure, the acceptability of a proposition can now be defined as follows.

Definition (acceptability). An atomic proposition p is **acceptable** in an argument evaluation structure ⟨*stage*, *audience*, *standard*⟩ if and only if *standard*(p, *stage*, *audience*) is true.

All the proof standards make use of the concept of argument applicability, so let us define this concept first.

Definition (argument applicability). Let ⟨*stage*, *audience*, *standard*⟩ be an argument evaluation structure. An argument ⟨P, c⟩ is **applicable** in this argument evaluation structure if and only if:

- every premise p ∪ P is a fact of the audience or, if neither p nor the complement of p is a fact, has been assigned the status "assumed" in the stage of dialogue or is acceptable in the argument evaluation structure, and
- no undercutter of the argument ⟨P, c⟩ is applicable, where an undercutter is any argument whose conclusion is undercut(a_i) and a_i is the constant assigned L denoting the argument ⟨P, c⟩ .

Now we are ready to define proof standards. For our purposes here, it should suffice to define two example proof standards, preponderance of evidence and dialectical validity. Other proof standards can be defined similarly. See Gordon and Walton (2009) for definitions of further standards, including clear and convincing evidence and beyond reasonable doubt.

Definition (preponderance of the evidence). Let ⟨*stage*, *audience*, *standard*⟩ be an argument evaluation structure and let p be a literal in L. *preponderance*(p, *stage*, *audience*) = true if and only if

- there is at least one applicable argument pro p in *stage* and
- the maximum weight assigned by the audience to the applicable arguments pro p is greater than the maximum weight of the applicable arguments con p.

Definition (dialectical validity). Let ⟨*stage*, *audience*, *standard*⟩ be an argument evaluation structure and let p be a literal in L. *dialectical-validity* (p, *stage*, *audience*) = true if and only if there is at least one applicable argument pro p in *stage* and no argument con p in *stage* is applicable.

The dialectical validity standard is suitable for aggregating arguments from general rules and exceptions, where any applicable exception is enough to override the general rule.

4. The audience

It has been integral to the notion of rational argumentation in dialectical systems that in order for an argument to be successful it must be based on premises that are commitments of the other party to whom the argument was directed (Walton and Krabbe 1995). This condition for the success of an argument is called the requirement of premise adequacy (Blair and Johnson 1987: 49), which states that the proponent of an argument must begin with premises that the audience is willing to concede, agree with, or accept, if he is to succeed in gaining the adherence of the audience to whom this argument is addressed. This requirement is a necessary condition for an argument to be successful, but not a sufficient condition. The argument must meet other requirements, for example it must be structurally correct according to some account of argument validity. A version of the premise adequacy requirement appears in the Carneades system for argument invention, because surely in order to find an argument that can successfully be used to rationally convince the party or audience to whom the argument is supposedly directed, an arguer needs to base it on premises that his audience has accepted, or shown some evidence of willingness to accept them. The audience in the Carneades argumentation system can be the proponent, in single-agent argumentation processes, the respondent, in two-party dialogues, or a third party, such as a judge or jury in legal cases.

Tindale (1990: 288) raised doubts about the requirement of premise adequacy. He wondered whether it could represent a form of epistemological relativism, for if an arguer exploits prejudices or outrageous views that his audience accepts, surely this argument ought not to be evaluated as one that should be judged to be good. On the other hand, it is recognized in argumentation studies that if the two parties in the dialogue, the proponent and the respondent, disagree about everything, and do not share any propositions that they both accept, or can agree on as acceptable, their attempt to engage in rational argumentation will be futile. Every persuasion dialogue is based on a conflict of opinions, meaning that the two parties disagree about some fundamental issue to be resolved. But if they disagree about everything, using rational argumentation in a persuasion dialogue to resolve this fundamental disagreement will inevitably fail. Hence it is one requirement of a critical discussion, a type of persuasion dialogue, that the two parties share what are called common starting points (van Eemeren and Grootendorst 1992: 9). Others in argumentation theory (Freeman 1991) have stated that common knowledge of the community of interlocutors cannot be dispensed with as a condition of premise acceptability, because arguments have to start somewhere. Freeman (1995: 270) also stated that premise acceptability requires that challengers in dialectical argumentation need to be representatives of the community of rational inquirers. In other words, participants in a dialectical exchange like that of a critical discussion need to generally presume that the party to whom their argument is directed will share background knowledge with the speaker, and agree not to dispute some propositions that can be accepted as common knowledge.

Clearly then the requirement of premise adequacy makes sense, and is an important part of the notion of a dialectical argument in which a respondent puts forward a particular argument to try to get a respondent (or audience) to come to accept some particular proposition that the proponent accepts but the respondent does not. We have to be careful however, in framing the requirements for what should be considered a successful or rationally convincing argument within a dialectical framework, in order to avoid the danger of epistemological relativism of a bad sort. Although the use of the term 'audience' automatically evokes this worry, seeing a respondent or audience as a participant in the dialectical framework should not lead anyone to cry wolf about dangerous relativism. In any dialectical framework where argumentation is being used for some purpose, for example to rationally convince another party to come to accept a proposition, a good argument must be based on premises that are (or can be made) acceptable to the party who is to be convinced. To put this in another way, it is a general feature of argumentation in such a dialectical framework that it be commitment-based (Walton and Krabbe 1995). This feature is a kind of rationality assumption requiring that the two parties in the argument exchange share what is called a commitment store (Hamblin 1971), a repository that contains all the propositions each one of them has gone on record as accepting in their previous exchange. As the discussion proceeds, propositions are inserted into this store or deleted from it according to procedural rules (protocols) that govern each move in the exchange. This notion of the commitment store is common to all dialectical systems. The notion of commitment can be taken as equivalent, or very closely equivalent, to the notion of acceptance (Walton 2010). Carneades, instead of being built around the notions of belief, desire, and intention, is built around the notion of commitment or acceptance. It is different from a belief revision system.

The notion of an audience, and the accompanying requirement of premise adequacy for arguments directed to that audience, become particularly important when building a system of argument invention. The arguer needs to build his argument with the goal of trying to get them to accept some designated proposition that represents his thesis to be proved by basing his arguments for the thesis on premises that his audience either accepts or can be gotten to accept through further argumentation. This doesn't mean that everything the audience accepts is true, or can be accepted generally as true. It also doesn't mean that if the speaker finds a successful argument that leads from premises that his audience accepts to the conclusion that his thesis to be proved should be acceptable to them, his argument should be accepted by any audience. It simply means that if this audience accepts the premises, and is therefore committed to them, then if the argument is valid (or structurally correct by application of argumentation schemes) it should also accept the conclusion. The situation is no different from any use of an argument. For example, in classical deductive logic, for an argument to be valid it is not necessary that all the premises are true. Validity only means that if the premises are true then the conclusion has to be true as well. So there is no epistemological relativity, at least of the worrisome sort, involved in a system of argument invention that operates on the basis of the requirement of premise adequacy.

Carneades calculates on the basis of argumentation schemes and on the basis of which premises in an argument are accepted or rejected (or subject to doubt), whether the conclusion held to follow from those premises should be accepted or rejected (or held as questionable). To use the system, an arguer provides input on which premises should have given values to start with (accepted, rejected, or questionable), and then the system automatically calculates what value the conclusion should have. Also, Carneades does not just do this through a single one-step argument, but is typically applied to a complex argument consisting of a chain of argumentation where a conclusion in one sub-argument can be a premise in another one. When used for argument invention, it searches around for such an argumentation chain (a path through the argumentation tree) that ends in the proposition the arguer seeks to prove, and has at least some premises that the audience accepts. If it finds such a path, the job is finished, and Carneades can tell the user which other premises need to proved. If it finds no such complete path, it gives advice on what positions could be useful to help find such a path.

5. An example

The first example we will consider was found on a Debatepedia page (accessed Sept. 20, 2010) presenting many pro and con arguments on the issue of whether

Wikipedia is a reliable and socially beneficial resource.[4] It is possible to use one stage of the argumentation in the dialogue containing a small segment of the argument to illustrate how the Carneades tool for argument invention can be used in an example of this sort. The ultimate conclusion, the goal to be proved or disproved, is the proposition that Wikipedia is reliable. For purposes of illustration we will take the case where there is one pro argument, that is, one argument supporting the ultimate conclusion, and one con argument, that is, one argument attacking the ultimate conclusion.

Pro Argument
Encyclopedia Britannica is reliable, and Wikipedia is as reliable as Encyclopedia Britannica, therefore Wikipedia is reliable.

Con Argument
Wikipedia's openness makes it subject to errors. My first argument for this claim is that Wikis cannot be reliable if anyone can edit them, unless editing changes are checked by the editors. My second argument is that Wikipedia articles can be written by nonexperts. My third argument is that anyone can insert text into a Wikipedia article.

This example is expressed in a simplified form. In the natural language text on whether Wikipedia is valuable from which this simple example was abstracted, there are many other sub arguments, missing premises, questions of interpretation and so forth. For example, a study by experts in the journal *Nature* was claimed to have found that Wikipedia is almost as reliable as *Encyclopedia Britannica* in some respects, and this appeal to expert opinion was brought forward and criticized by both sides extensively (Giles 2005). One could imagine the Debatepedia argumentation being analyzed in greater depth. One could also imagine raising doubts about the argument by critical questioning concerning the issue of how often editing changes are checked by the editors and how this process is carried out. However, our aim for purposes of illustration of the argument invention system is to keep the example argument simple, so that anyone can understand the general principle of how it works even in a case like this.

We begin by showing how all the argumentation, including both the pro and the con arguments, can be represented as an argument map using Carneades, shown in Figure 3.

Argument A1 is an example of an argument with a single premise. Argument A2 has two premises that go together to support the conclusion that Wikipedia is reliable. It has the form of a linked argument. Arguments A1 and A2 are represented as two separate arguments about the issue of whether Wikipedia is reliable.

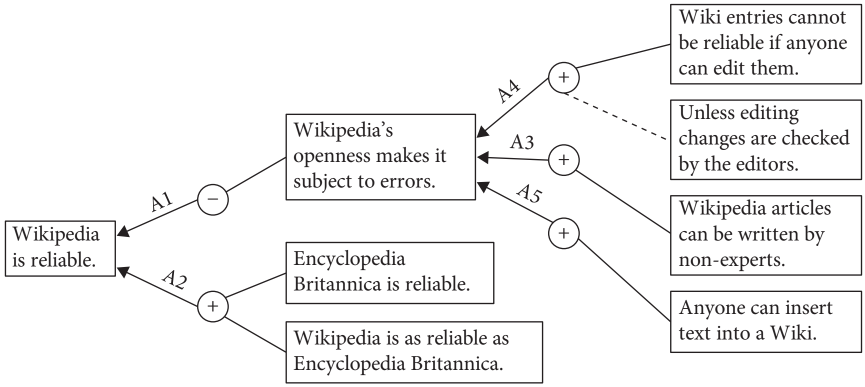

Figure 3. Carneades' map showing the arguments in the Wikipedia example

A1 is a con argument, with the conclusion that Wikipedia is not reliable, and A2 is a pro argument, with the conclusion that Wikipedia is reliable. This pattern of argumentation is called a convergent argument in informal logic. Also notice that arguments can be chained together using Carneades, so that the conclusion of one argument can also function as a premise in a subsequent argument. The pro argument is displayed using by putting a plus sign in the node that represents the argument. Argument A2 is an example of a pro argument. A con argument is displayed using a minus sign in the argument node. Argument A1 is an example of a con argument.

One premise in the argument A4 is joined to its argument node by a solid line, whereas the other premise is joined to the same argument node using a broken line. To understand this notation we have to have some idea of how Carneades represents different kinds of premises to model the asking of critical questions corresponding to an argumentation scheme (Walton and Gordon 2005). The critical questions do not behave in this regard in a uniform manner, and it is possible to have two different theories about how burden of proof should be managed. On the one theory, when a critical question is asked, the burden of proof shifts to the proponent's side to answer it. On the other theory, merely asking the question is not enough to defeat the proponent's argument until the questioner offers some evidence to support the question. To provide the capability to distinguish between these two kinds of critical questions and to manage the problem of burden of proof that they pose, Carneades distinguishes three types of premises in an argument scheme, called ordinary premises, assumptions and exceptions. The ordinary premises must be proved to hold. Assumptions are additional premises that hold without proof until they have been questioned, after which they must be proven in the same as ordinary premises. They are assumed to be acceptable unless called into question. Exceptions are modeled as premises that hold unless they are proven with evidence to back them up. An argument can be undercut by proving one of its exceptions. This is difficult to state in a way which is not confusing, because in the formal model an exception premise holds unless the opponent proves it does not hold.

In argument diagrams, ordinary premises are shown with solid, black lines. Assumptions are represented using dotted black lines. And exceptions are represented as dashed lines. You could express the difference between exceptions and assumptions by saying that the questioner has the burden of proof to back up his question in the case of an exception, whereas in the case of an assumption, the arguer has the burden of proof to back up the assumption once it is questioned or challenged by the other side.

As shown in Figure 3, argument A4 has two premises. The top premise, which says that Wiki entries cannot be reliable if anyone can edit them, is an assumption, meaning that if it is challenged by a questioner, the burden of proof is on the arguer who put forward A4 to prove this premise, or at least to give some evidential support for it. Otherwise argument A4 fails. The bottom premise is taken to represent an exception, so the burden of proof is on the side who is questioning the argument, the party who disagrees with the claim that Wikipedia's openness makes it subject to errors. The word 'unless' redundantly put in the text box for the bottom premise, has been added to make its role as an exception even clearer.[5]

6. How Carneades works

Carneades uses formal models of argumentation schemes to support the construction of arguments in two different ways. The first way instantiates schemes using information in a database (or knowledge base). This method is fully automatic, assuming the relevant information has already been entered into the database. In general of course this assumption does not hold. It would indeed be an enormous task to put all information into a database, especially if this is to be done in advance of any concrete need for the information in some case. The second way to use argumentation schemes in Carneades to help users to construct arguments is more interactive. The system asks the users questions, in a dialogue, to gather the information needed to instantiate the schemes. That is, this second method gathers the information as it is needed, in a goal-directed way, without requiring the user, or anyone else, to enter all the relevant information into a database in advance. Again, both methods are supported by Carneades, and they can both be used together.

Although the focus of this paper is on how Carneades can be used to construct new arguments, and not on how it is used to evaluate arguments, it is necessary to say a few words about argument evaluation in order to see how the system carries out the task of argument construction. Arguments are evaluated on the basis of proof standards (Gordon and Walton 2009). As shown in the formal model in Section 2, it is assumed that each statement in an argument has some proof standard that needs to be met in order for the argument to be considered applicable. Each of these proof standards is ranked in order of strength, clear and convincing evidence being a higher standard than those presented in Section 3. There is also an even higher standard of proof beyond a reasonable doubt (Gordon and Walton 2009: 447). This standard is applicable in legal argumentation, but we do not have space to discuss it here.

Carneades is set automatically to the standard of preponderance of evidence when the user opens the graphical interface, but the menu allows the standard of proof set for any statement to be changed. One particular feature that Carneades has is the capability for inputting a standard of proof for a statement that is appropriate. The proof standard to use is determined by the dialogue type and its procedural rules (protocol), not by the audience. For example, in a legal court proceeding, the proof standard to use depends on the type of proceeding. In a civil case, preponderance of the evidence is used. In a criminal case, beyond a reasonable doubt is used. Any information about whether this particular audience accepts, questions, or rejects a statement can also be handled as input to an argument. When any changes are made to an existing argument graph, Carneades automatically changes the acceptability of other statements on the graph that follow from the statements that have been changed. In general, a statement is acceptable only if the pro-argumentation supporting it outweighs the con argumentation against it by a margin meeting the standard of proof set for that statement.

Carneades provides a number of "assistants" for helping users with various argumentation tasks, including a "find arguments" assistant for inventing arguments from argumentation schemes and facts in a knowledge base, a "instantiate scheme" assistant for constructing or reconstructing arguments by using argumentation schemes as forms, and a "find positions" assistant for helping users to find minimal, consistent sets of statements which, if accepted by the audience, would make a goal statement acceptable. Figure 4 shows a screen shot of the Carneades graphical user interface with the "find positions" assistant. The argument assistants are started using the "assistant" menu shown at the top left of the figure. Detailed information about a selected statement, "Anyone can insert text into a wiki", is shown in the statement "inspector" at the lower-left of the figure. Here we can see that the status of the statement is "accepted" and that it has been assigned the preponderance of evidence proof standard. We can also see, at the bottom of the statement inspector, that neither the statement nor its complement is currently acceptable. How the Wikipedia example is represented as an argument diagram

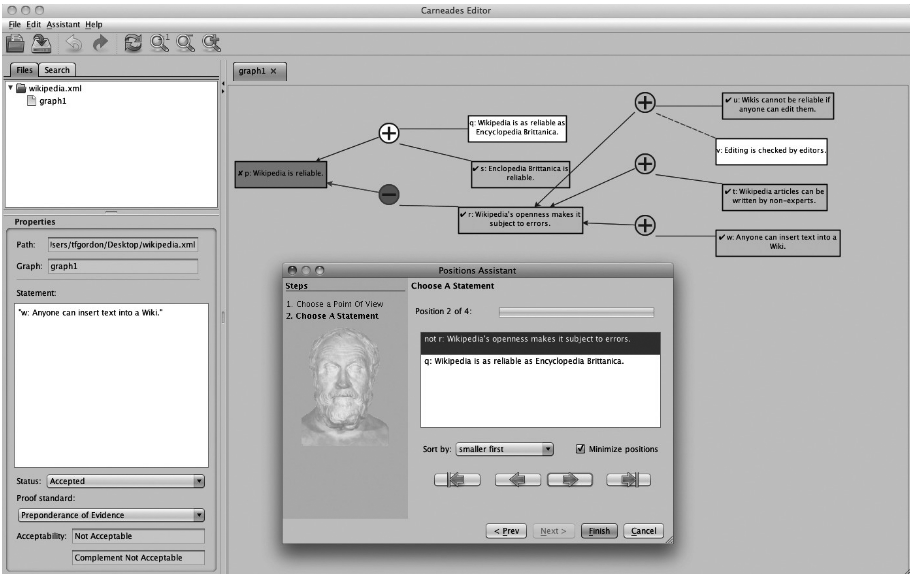

Figure 4. Screen shot of the Carneades menu and argument visualization

in the current version of the system is shown at the top-right of the window. The dialogue box of the "find positions" assistant is shown floating just below the argument diagram.

An argument is said to be applicable if all its premises hold. The menu of argumentation schemes that the user can find with the argument assistant can be applied to the argument map built by the user. Whether or not a premise holds depends on its type (ordinary, assumption, exception), status (stated, questions, accepted, rejected) and acceptability. For example, one way for an ordinary premise to hold is for it to be accepted. But in the case of an exception, the reverse is true. An accepted exception does not hold. This is because an exception is a kind of negation. The exception holds unless it is proven (or accepted).

When representing argumentation visually on a Carneades argument map, each statement has a status (stated, questioned, accepted, rejected), and a symbol appears in the box for each statement indicating its status as follows. If a statement is accepted or acceptable, a check mark appears in front of it and the text box has a green fill. If the complement of a statement is accepted or acceptable, an X mark appears in front of it, and the text box has a red fill. These indicators are redundant, so that the status of the statement is shown to someone who is colorblind. A question mark in front of statement indicates that the statement has been called into question. If the statement has not been questioned and neither the statement nor its complement is accepted or acceptable, there is no symbol in front of it, and the text box has white background (no fill).

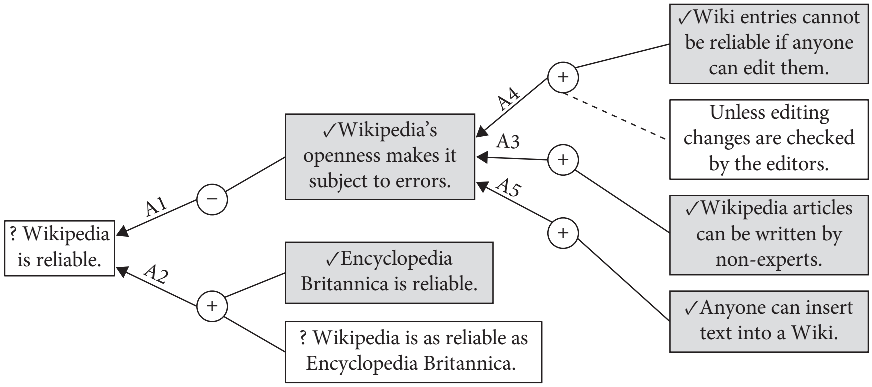

Figure 5. Audience input shown in the Wikipedia example

The Wikipedia example can be used to show how argument evaluation is carried out by Carneades. Suppose we have the situation shown in Figure 5 where the arguer has the goal of proving the statement that Wikipedia is reliable. The audience has accepted all the leaf statements shown with filled boxes, on the right-hand side of the diagram. There are only three statements that are not accepted or acceptable. The audience questions the statement that Wikipedia is reliable, and questions the statement that Wikipedia is as reliable as Encyclopedia Britannica. The statement that editing changes are checked by the editors, an exception of argument A4, is merely stated but not accepted.

Now the question is this. Is the argument a deadlock, where arguments A1 and A2 are now equally persuasive? Or is it the case that one of these arguments is stronger than the other?

Notice that A2 has one premise that is accepted, but the other premise is questioned. Thus A2 is not applicable. It appears, however, that A1 is applicable. A1 has three arguments supporting it, A3, A4, and A5. A3 and A5 are both single-premised arguments, and in each instance the premise is accepted. A4 has two premises. One of them has been accepted. However, as noted above, the premise that editing changes are checked by the editors is represented as an exception. Hence the burden of proof is on the side who wants to undercut the argument, the party who disagrees with the claim that Wikipedia's openness makes it subject to errors. So the exception of argument A4, shown in the white box, does not need to be proved by the arguer in order for the argument to be applicable. This means that the con argument, A1, is applicable, while the pro argument A2, is not and therefore the ultimate conclusion that Wikipedia is reliable is not proved. However, if both the pro and the con argument had been acceptable, then the outcome would have depended on the proof standard assigned to the statement that Wikipedia is reliable. As noted above, by default the preponderance of evidence standard is used. With this proof standard, if the pro argument, A2, would be given greater weight than the con argument, A2, by the audience, then the pro argument would win, and Carneades would automatically change the status of the conclusion that Wikipedia is reliable from questioned to acceptable, and change the fill color of its text box to green and replace the question mark in front of the statement that Wikipedia is reliable with a check mark.

7. How the knowledge base is used

Carneades can not only help to construct an argument map of a given argument and provide a way of evaluating that argument as stronger or weaker than competing arguments for the purpose of persuading an audience to accept a conclusion. It can also use knowledge bases to help a user to search for new arguments to support his claim or attack a competing argument that might refute the claim. The aim of the argument invention system is to assist an arguer preparing an argument for presentation to an audience by constructing arguments. To find premises and new arguments based on them, the arguer can draw from a knowledge base of argumentation schemes and facts representing information useful for persuading the audience. The schemes representing knowledge of the domain in the knowledge base must be programmed manually by an expert.[6] Alternatively, Carneades provides an "instantiate scheme" assistant for using argumentation schemes to construct or reconstruct arguments from information which has not be represented formally in a knowledge base. The information needed to instantiate the schemes may be discovered using an external information services, such as Google. The three main components of the system are the arguer, the information service, and the argument evaluation structure already described in the previous sections. Using the argument evaluation structure, the agent forms an initial impression of his audience to make assumptions about what statements the audience will accept, and takes into account proof standards that define how strong an argument needs to be in order to be acceptable.

For example, suppose I want to bolster my argument A2 to show that Wikipedia is reliable so that I can build a stronger argument that will attack and possibly even defeat the opposed argument A1. What should I do? Should I make a further argument pro my claim that Wikipedia is reliable? For example, I could put forward arguments supporting some premise of one of my previous arguments, such as the claim that Wikipedia is as reliable as Encyclopedia Britannica. Or I could make another argument pro my claim that Wikipedia is reliable. Or maybe I can find a way to attack my opponent's con argument, A1. Maybe her argument is particularly vulnerable in some way, and I should choose that point of attack as my next move.

Carneades has three argument assistants that can help a user to answer these questions: the “find positions” assistant, the “instantiate scheme” assistant and the “find arguments” assistant. Let’s begin by explaining the “find positions” assistant first. The find positions assistant helps uses to answer the question ‘What should be my next goal?’. A goal is a proposition that an arguer chooses to work on next by looking for arguments pro the proposition. A position, according to Ballnat and Gordon (2010), is a minimal, consistent set of literal propositions which, if accepted by the audience, would enable a statement at issue to become acceptable by satisfying its proof standard. Ballnat and Gordon (2010) have devised a cyclic procedure, shown in outline in Figure 6, for selecting goals, searching for information to use for constructing arguments pro the goal from argumentation schemes, selecting an argument to put forward, reevaluating the arguments and then, returning to the beginning of the cycle, recomputing the positions and selecting the next goal to work on. This cyclic procedure is carried out by the human users of the Carneades system, using the argument assistants of Carneades as tools. The “find positions” assistant computes positions using abductive reasoning. When there is more than one set of positions, or more than one statement in a position, the next problem is to choose among them.

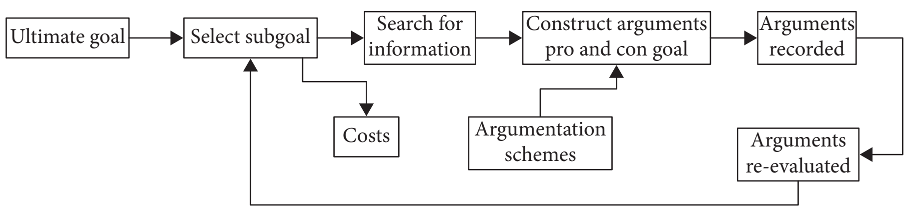

Figure 6. Procedure for computing alternative sets of goals

Ballnat and Gordon (2010) use a cost function to define a preference ordering over positions that estimates how difficult it would be to construct arguments sufficient to persuade the audience to accept the propositions of the position.

The find positions assistant can now be used by anyone who wishes to apply it to examples of arguments. It yields four positions in the Wikipedia example. The first position is {Wikipedia is reliable}. This position is uninteresting. It merely says the users should try to convince the audience of his claim, but doesn’t provide any real help by suggesting how to do this. The second position is {not: Wikipedia’s openness makes it subject to errors; Wikipedia is as reliable as Encylopedia Brittanica}. This second position is more interesting and more helpful, because it tells us that both of these claims need to be proven. It tells us, in other words, that the arguer needs to persuade the audience of both claims. Proving that Wikipedia is as reliable as Encylopedia Britannica would make argument A2 applicable, giving us one good pro argument. But this isn’t enough, since the rebuttal, A1, is still applicable and both arguments have equal weight. Proving that it is not the case that Wikipedia's openness makes its subject to errors would defeat this rebuttal. The last two positions go into more detail, showing ways to disprove the claim that Wikipedia's openness makes it subject to errors by telling what needs to be done to defeat all three of the arguments supporting this claim.

Carneades views argumentation as a process for reasoning with limited resources in an efficient and rational manner. The argumentation that is the primary focus is defeasible, meaning that it is subject to defeat or refutation as new information comes in. This view requires that the argumentation be open to new information that is continually coming in and which requires revision of one's arguments as this information is taken into account.

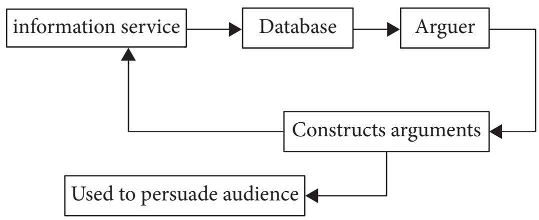

Figure 7. The process of finding new arguments

Moreover, Carneades can make use of a database (knowledge base) of rules and facts to invent arguments, with the help of the "find arguments" assistant. In the example, the "find arguments" assistant of Carneades would look through its database to find information that could be used to construct arguments to support A2 or to attack A1.

8. Application to the two examples

The "find arguments" and "instantiate scheme" assistants are designed to help users to invent arguments. They can help users to search for new arguments that can used to supply needed support for parts of one's existing argument that need to be made stronger, or to attack weak points in a competing argument that attacks or defeats one's argument. Let's suppose that there is a statement in the database saying that a study in the journal *Nature* found that Wikipedia is as reliable as Encyclopedia Britannica. Let's suppose that it is in the database because it has been assumed to be accepted by the audience. Let's also suppose that the audience accepts the statement that the journal *Nature* is an acceptable source for expert opinions. The argumentation scheme for argument from expert opinion can be included in a set of argumentation schemes loaded into Carneades for use by the "instantiate scheme" assistant. Then these two statements can be used as premises in an applicable argument from expert opinion to prove that the previously questionable statement that Wikipedia is as reliable as Encyclopedia Britannica. This situation is shown in Figure 8. Argument A6 is applicable since its two premises have been accepted by the audience. Thus, the conclusion that Wikipedia is as reliable as Encyclopedia Britannica is now acceptable. Since the statement is acceptable, it is now shown in a filled text box in Figure 8.

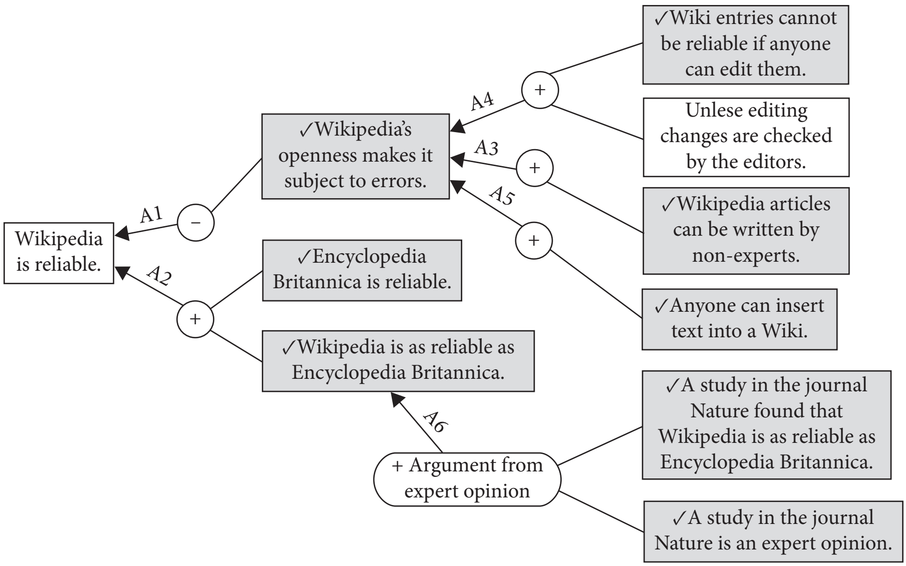

Figure 8. Finding a new argument in the Wikipedia example

Now the question is this. Is the argument a deadlock, where arguments A1 and A2 are now equally persuasive? The answer depends on the proof standard for the statement 'Wikipedia is reliable' and the weights assigned by the audience to the arguments. As noted above, by default the preponderance of evidence standard is used. With this proof standard, the stronger argument wins. In the example, the pro argument A2 and the con argument A1 are both acceptable and have the same weight. Thus, the "Wikipedia is reliable" statement does not satisfy the assigned preponderance of evidence standard and is not acceptable. The argument A2 is applicable because both of its premises hold. The "Encyclopedia Britannica" premise holds because it has been accepted by the audience. The "Wikipedia is as reliable as Encyclopedia Britannica" premise holds because it is acceptable, since it is supported by an applicable argument from expert opinion, A6. Argument A6 is applicable because both of its premises are accepted by the audience. However, the counterargument A1 is applicable because its only premise, "Wikipedia's openness makes it subject to errors" is acceptable, since it is support by three arguments, all of which are applicable. The ordinary premises of these arguments are accepted by the audience. The exception of argument A4, "Unless editing changes are checked by the editor", has not been accepted by the audience, and is thus shown with a white background. But since this is an exception, the burden of proof is on the other party to prove it. It doesn't need to be proved by the arguer. Again, given this information, the claim that "Wikipedia is reliable" is not acceptable, since it does not satisfy the preponderance of evidence standard which has been assigned to it.

The question remains how Carneades would deal with the climate science example shown in the Araucaria argument diagram in Figure 1. Araucaria could only deal with this instance of argument invention in an oblique way by prompting a critical question that might suggest some ways of attacking the argument. With Carneades, the "find arguments" assistant can be used to search a knowledge base for arguments that could be used to either support or refute the argument from expert opinion in this case. Suppose there is a statement in the database asserting that Bill's research is funded by industries that have financial interests at stake. As shown in Figure 10, Carneades can find that statement in the database and then use it as an exception in the argument from expert opinion. Let's say that this statement is in the database of statements that is accepted by the audience. And let's say that both of the premises of the argument from expert opinion are also accepted by the audience. Then we have the situation represented in Figure 9.

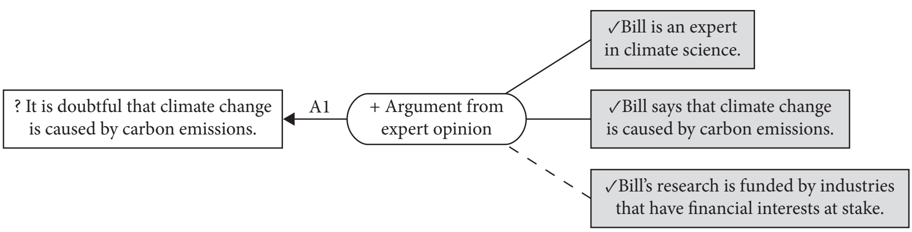

Figure 9. How Carneades models the climate science example

What Figure 9 shows is that argument from expert opinion is an argumentation scheme in Carneades, and both of the premises of the argument are accepted. Both premises are in darkened text boxes, and the scheme for argument from expert opinion shown in Figure 9 has been applied to reconstruct argument A1. However we need to notice also that the exception, the statement that Bill's research is funded by industries that have financial interests at stake, is also shown in a darkened box, meaning that it also has been accepted by the audience. If this statement was not in a darkened box, but was in a white box showing it was not accepted, then the conclusion that it is doubtful that climate change is caused by carbon emissions would be acceptable. However, it is not, as shown in Figure 9. The statement with a question mark in front of it, shown in a white box, is questioned by the audience and not acceptable.

What this example shows is that Carneades has a more direct and comprehensive method for argument invention than other leading argument evaluation systems (for example, Araucaria). Like Araucaria, it does use critical questions matching schemes to find weak spots in an argument that can be questioned, or that can be followed up by producing a counterargument that is suggested by the question. But it can also find positions to work from to improve an existing argument, and it can search for and find new arguments from its database that might be used either to attack or support an existing argument. In this sense it is a full-fledged system of argument invention.

9. Application to a legal example

In this section we show how the Carneades system for argument invention can be applied in a more detailed way to a real legal case that concerns software copyright licensing. The issue in the case is whether the Carneades engine may use a particular software license or not (Gordon 2011). The example illustrates legal argumentation on both sides of an issue concerning open source license compatibility issues.

A software copyright license can grant developers the right to use the software in their own programs, for example by making modifications to the source code of the software. Whether or not a license is required can depend on whether the use creates a "derivative work". This issue depends on how the legal concept of a "derivative work" is interpreted by the courts. One rule is that software derived from software licensed using a so-called "reciprocal" license must be published using a license that is compatible with the license of the original software. There are two conflicting views about whether or not linking software to library code creates a derivative work. According to the lawyers of the Free Software Foundation, linking does create a derivative work. A well-known legal expert on open source licensing issues, Lawrence Rosen, takes the opposing point of view, arguing that linking is not sufficient to create derivative works.

The issue in this instance is whether a project may publish its software using a particular open source license, the Eclipse Public License (EPL). To put the question another way, the issue is whether the license preferred by the developers is compatible with the licenses of the software used by the project. The ultimate conclusion of the sequence of argumentation is statement contained in the text box at the far left, 'The Carneades engine may use the EPL license'. The pro-argument at the top is based on the default license rule, which says that copyright owners are

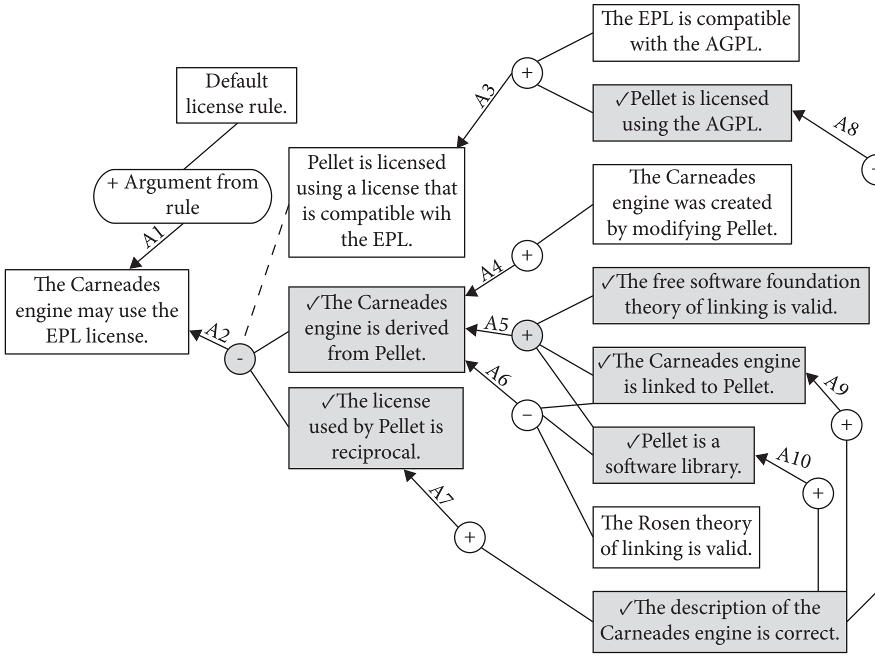

Figure 10. Argument map of the Copyright License example

free to use any license they wish. This rule was used to construct a pro-argument supporting the ultimate conclusion that the Carneades engine may use the EPL license, but the rule is defeasible, subject to exceptions. One of the libraries used by the Carneades inference engine, Pellet, is licensed using the Affero General Public License (AGPL). The con argument, represented by a circular node containing a minus sign, was constructed using an exception to this rule. This large con argument, as can be seen in Figure 10, raises sub-issues, one of which is whether or not linking to a software library creates a derivative work. The reciprocity rule states that software must be licensed using a license that is compatible with the reciprocal licenses of any software from which it is derived. As a chain of argumentation, Figure 10 represents the sub-arguments that are connected together in a sequence of argumentation supporting and attacking the ultimate conclusion shown at the left. Looked at in this way it represents an argument map showing the pro and contra argumentation that needs to be considered when evaluating the argument in order to decide whether the conclusion is acceptable or not.

However, there is another way to use this argument map. The "find positions" assistant of Carneades uses abductive reasoning to compute minimal sets of statements which, if accepted would make the ultimate conclusion acceptable or not acceptable. So instead of evaluating arguments, the find positions assistant reasons backwards to compute minimal sets of statements, called "positions", that, if proven or accepted by the audience, would cause the desired ultimate conclusion to be provable, satisfying its proof standard. The positions computed by the assistant provide the user with information about which statements require further supporting arguments or evidence in order to make his case. In the example, one of the positions computed by the assistant consists of a single statement, about whether or not the Free Software Foundation theory of linking is valid. If arguments can be put forward sufficient to prove to the audience that this theory of linking is not valid, then the argument pro Carneades being derived from Pellet would not be applicable, thus undermining the only argument con the main claim that the Carneades engine may be licensed using the EPL.

Once the issues have been selected, with the help of the "find positions" assistant, other Carneades tools can be used to search for and construct arguments for these issues. In addition to the tool which automatically constructs arguments from a knowledge base, there is an interactive tool, called the "instantiate scheme" assistant that helps the use to apply argumentation schemes, for example argument from expert opinion, to construct additional arguments and add them to an existing argument. Both of these tools support argument invention and can be used together. The tool which constructs arguments from a knowledge base is useful when it is feasible to model the required knowledge, for example by modeling the rules of copyright law, as in this example. The instantiate scheme tool is more appropriate for arguments which fall outside the scope of any available domain models.

10. Conclusions

The survey of the historical literature has been very brief in this paper, but this body of literature has shown that since the time of the Greek philosophers and rhetoricians, components for approaching argument invention with stasis and topics have been known. However, these components have not been articulated precisely enough or put together in a systematic enough way to be very useful by themselves. However, now they have been integrated, the next step of applying computational methods to the task of argument invention can be seen as a significant enterprise already yielding useful tools.

It is concluded that the prospects of developing an automated system of argument invention are very good. It has been shown that the argumentation software systems surveyed briefly in the second section had not taken up the task of argument invention directly, but have concentrated on other equally important tasks like argument analysis and evaluation. These tools are increasingly used for the retrospective task of analyzing existing arguments by a single user, and for computer supported-collaborative argumentation to accomplish group tasks of various kinds. Inference engines for nonmonotonic logics, such as Nute's Defeasible Logic (1994), can be used to invent arguments and have existed for many years. What is new in Carneades is the integration of such an inference engine in an argument mapping tool. Only with the advent of Carneades have these argument mapping software systems now begun to see argument invention as a central task. While we emphasized that some single-user argument evaluation systems have a significant capability for argument invention, Carneades goes much further to develop a much more comprehensive and well developed set of tools for finding positions to support an argument, and even for searching around for new arguments that could supplement an existing network of argumentation in order to help it prove a claim.

Using Carneades, the arguer only presents his case when the resources provided by the information service have been exhausted. If that has not happened, the arguer tries to improve his case by asking questions and searching for information to construct arguments. This is a very interesting juncture to represent when modeling argumentation, because in many cases one of the most important decisions to be made, the problem is whether to draw a conclusion on the basis of the evidence one already has or to delay drawing a conclusion in order to continue collecting evidence before reaching a decision. Having reached this point of concluding that the resources provided by the information service have been exhausted, the arguer selects which arguments to put forward in order to best persuade the audience to accept the thesis that he needs to prove. In the sequence of argumentation in the Carneades system therefore, there is a continuous loop as the arguer keeps collecting new information from the information service and uses that information to construct new arguments. Only once these resources are exhausted does he break out of this loop and present his case to the audience based on the argument now formed. As the arguer proceeds through the loop, he uses abductive reasoning to find alternative positions to support his argument, and he evaluates which is the best position to move forward with from that point.

Notes

1. http://carneades.github.com/

2. An earlier version of the Carneades argument mapping tool that did not yet support argument invention was included in this survey.

3. The restriction prohibiting cycles is not as egregious as it might seem at first glance, since rebuttals, i.e., conflicting pro and con arguments, do not cause cycles. Nonetheless, we are currently looking for a way to remove this limitation of the model.

4. http://debatepedia.idebate.org/en/index.php/Debate:_Is_Wikipedia_valuable%3F

5. This way of visualizing exceptions cannot be used in the general case, since the statement could be used in several arguments, as a different kind of premise in each argument. This is why they type of premise is visualized by the kind of link (solid, dotted, dashed) between the statement and argument, and not as an attribute of the statement's box.

6. The facts (i.e., statements accepted by the audience) are entered in the Carneades Editor (shown in Figure 4) by the user before the schemes can be applied to construct arguments. But in a new version currently being developed, the user can also be prompted to enter relevant facts, interactively in a dialogue, as the schemes are applied.

References

Aristotle. 1937. *The Art of Rhetoric*. J. Henry Freese (trans). Loeb Classical Library. Cambridge, MA: Harvard University Press.

Aristotle. 1939. *Topics*. E.S. Forster (trans). Loeb Classical Library. Cambridge, MA: Harvard University Press.

Ballnat, S. and Gordon, T.F. 2010. "Goal selection in argumentation processes". In P. Baroni, F. Cerutti, M. Giacomin, and G. R. Simari (eds), *Computational Models of Argument: Proceedings of COMMA 2010*. Amsterdam: IOS Press, 51–62.

Bench-Capon, T.J.M., Doutre, S., and Dunne, P.E. 2007. "Audiences in argumentation frameworks". *Artificial Intelligence* 171(1): 42–71.

Bench-Capon, T.J.M. and Sartor, G. 2003. "A model of legal reasoning with cases incorporating theories and values". *Artificial Intelligence* 150(1–2): 97–143.

Bird, O. 1962. "The tradition of the Logical Topics: Aristotle to Ockham". *Journal of the History of Ideas* 23(3): 307–323.

Blair J.H. and Johnson, R.H. 1987. "Argumentation as dialectical". *Argumentation* 1(1): 41–56.

Buckingham Shum, S.J., MacLean, A., Bellotti, V.M.E., and Hammond, N.V. 1997. "Graphical argumentation and design cognition". *Human-Computer Interaction* 12(3): 267–300.

Freeman, J.B. 1991. *Dialectics and the Macrostructure of Arguments.* Berlin: Foris.

Freeman, J.B. 1995. "The appeal to popularity and presumption by common knowledge". In H.V. Hansen and R.C. Pinto (eds), *Fallacies: Classical and Contemporary Readings*. University Park, PA: Penn State Press, 263–273.

Giles. J. 2005. "Internet encyclopaedias go head to head". *Nature* 438: 900–901. doi: 10.1038/438900a; [Published online 14 December 2005]

Gordon, T.F. 2005. "A computational model of argument for legal reasoning support systems". In P.E. Dunne and T.J.M Bench-Capon, *Argumentation in Artificial Intelligence and Law,* IAAIL Workshop Series. Nijmegen: Wolf Legal Publishers, 53–64.

Gordon, T.F. 2010. "The Carneades argumentation support system". In C. Reed and C.W. Tindale (eds), *Dialectics, Dialogue and Argumentation*. London: College Publications, 145–146.

Gordon, T.F. 2011. "Analyzing open source license compatibility issues with Carneades". *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law* (ICAIL-2011). New York: ACM Press, 50–55.

Gordon T.F. and Walton, D. 2006. "The Carneades argumentation framework". In P.E. Dunne and T.J.M. Bench-Capon (eds), *Computational Models of Argument: Proceedings of COMMA 2006*. Amsterdam: IOS Press, 195–207.
Gordon, T.F., Prakken, H., and Walton, D. 2007. "The Carneades model of argument and burden of proof". *Artificial Intelligence* 171: 875–896.
Gordon, T.F. and Walton, D. 2009. "Proof burdens and standards". In . I. Rahwan and G. Simari (eds), *Argumentation and Artificial Intelligence*. Berlin: Springer, 239–260.
Grennan, W. 1997. *Informal Logic*. Kingston: McGill-Queen's University Press.
Hamblin, C.L. 1971. "Mathematical models of dialogue". *Theoria* 37: 130–155.
Hastings, A.C. 1963. *A Reformulation of the Modes of Reasoning in Argumentation*. Unpublished Ph.D. Dissertation. Northwestern University, Evanston, Illinois.
Hohmann, H. 1989. "The dynamics of stasis: Classical rhetorical theory and modern legal argumentation". *American Journal of Jurisprudence* 34: 171–197.
Hohmann, H. 2000. "Rhetoric and dialectic: Some historical and legal perspectives". *Argumentation* 14: 223–234.
Kennedy, G. 1963. *The Art of Persuasion in Greece*. London: Routledge.
Kienpointner, M. 1992. *Alltagslogik: Struktur und Funktion von Argumentationsmustern*, Stuttgart: Fromman-Holzboog.
Kienpointner, M. 1997. "On the art of finding arguments: What ancient and modern masters of invention have to tell us about the *Ars Inveniendi*". *Argumentation* 11: 225–236.
Leff, M. 1996. "Commonplaces and argumentation in Cicero and Quintilian". *Argumentation* 10: 445–452.
Leff, M. 1983. "The topics of argumentative invention in latin rhetorical theory from Cicero to Boethius". *Rhetorica* 1: 23–44.
Nute, D. 1994. "Defeasible logic". In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3: *Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford: Oxford University Press, 353–395.
Perelman, C. and Olbrechts-Tyteca, L. 1969. *The New Rhetoric: A Treatise on Argumentation*. Notre Dame: University of Notre Dame Press.
Reed, C. and Rowe, G. 2005. *Araucaria, Version 3.1*. Available free at http://www.computing.dundee.ac.uk/staff/creed/araucaria/.
Scheuer, O., Loll, F., Pinkwart, N., and McLaren, B.M. 2010. "Computer-supported argumentation: A review of the state of the art". *Computer-Supported Collaborative Learning* 5 (1): 43–102.
Tindale, C.W. 1990. "Audiences and acceptable premises: Epistemic and logical conditions". In van Eemeren et al (eds), *Proceedings of the Second International Conference on Argumentation*. Amsterdam: SICSAT, 288–295.
Tindale, C.W. 1999. *Acts of Arguing: A Rhetorical Model of Argument*. Albany, NY: State University of New York Press.
Walton, D. 2010. "A dialogue model of belief". *Argument and Computation* 1: 23–46.
Walton D. and Gordon, T.F. 2005. "Critical questions in computational models of legal argument". In P.E Dunne, and T.J.M Bench-Capon (eds), 103–111.
Walton, D., Reed, C., and Macagno, F. 2008. *Argumentation Schemes*. Cambridge, Cambridge University Press.

Authors' addresses

Douglas Walton
Centre for Research in Reasoning,
Argumentation and Rhetoric (CRRAR)
University of Windsor
2500 University Ave. W., Windsor
Ontario N9B 3Y1
Canada

dwalton@uwindsor.ca
www.dougwalton.ca

Thomas F. Gordon
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin
Germany

thomas.gordon@fokus.fraunhofer.de
www.tfgordon.de

About the authors

Douglas Walton is a Canadian academic and author, well known for his many widely published books and papers on argumentation, logical fallacies and informal logic. He presently holds the Assumption Chair in Argumentation Studies and is Distinguished Research Fellow of the Centre for Research in Reasoning, Argumentation, and Rhetoric (CRRAR) at the University of Windsor, Canada. Walton's work has been used to better prepare legal arguments and to help develop artificial intelligence. His books have been translated worldwide, and he attracts students from many countries to study with him. A festschrift honoring his contributions, *Dialectics, Dialogue and Argumentation*, (2010. College Publications, edited by C. Reed and C. W. Tindale), shows how his theories are increasingly finding applications in computer science.

Thomas F. Gordon conducts research on argumentation technology in the fields of artificial intelligence and law, legal informatics, and computational models of argument. He heads a research group on governance portals at the Fraunhofer Institute for Open Communications Systems (FOKUS) in Berlin and holds an honorary professorship for argumentation technology at the Institute of Computer Science of the University of Potsdam. The goal of this research is to provide software tools for governance tasks in the policy life cycle: agenda setting, analysis, policy development and legislation, implementation (including administrative rulemaking and electronic service delivery), and monitoring.