Computer-Supported Collaborative Learning
DOI 10.1007/s11412-009-9080-x

Computer-supported argumentation: A review of the state of the art

Oliver Scheuer · Frank Loll · Niels Pinkwart · Bruce M. McLaren

Received: 4 June 2009 / Accepted: 7 December 2009

Abstract Argumentation is an important skill to learn. It is valuable not only in many professional contexts, such as the law, science, politics, and business, but also in everyday life. However, not many people are good arguers. In response to this, researchers and practitioners over the past 15–20 years have developed software tools both to support and teach argumentation. Some of these tools are used in individual fashion, to present students with the "rules" of argumentation in a particular domain and give them an opportunity to practice, while other tools are used in collaborative fashion, to facilitate communication and argumentation between multiple, and perhaps distant, participants. In this paper, we review the extensive literature on argumentation systems, both individual and collaborative, and both supportive and educational, with an eye toward particular aspects of the past work. More specifically, we review the types of argument representations that have been used, the various types of interaction design and ontologies that have been employed, and the system architecture issues that have been addressed. In addition, we discuss intelligent and automated features that have been imbued in past systems, such as automatically analyzing the quality of arguments and providing intelligent feedback to support and/or tutor argumentation. We also discuss a variety of empirical studies that have been done with argumentation systems, including, among other aspects, studies that have evaluated the effect of argument diagrams (e.g., textual versus graphical), different representations, and adaptive feedback on learning argumentation. Finally, we conclude by summarizing the "lessons learned" from this large and impressive body of work, particularly focusing on lessons for the CSCL research community and its ongoing efforts to develop computer-mediated collaborative argumentation systems.

O. Scheuer (✉) · B. M. McLaren
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Campus D3_2, 66123 Saarbrücken, Germany
e-mail: oliver.scheuer@dfki.de

F. Loll · N. Pinkwart
Department of Informatics, Clausthal University of Technology,
Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany

Published online: 16 January 2010

Keywords Collaborative argumentation · Argumentation systems · Argument visualization · Analysis and feedback · Empirical studies of argumentation systems

Introduction

A key human skill, utilized across many domains and activities, in a variety of situations, is the ability to *argue*. Think of an attorney in a courtroom, a scientist positing a new theory, a politician arguing for a new policy proposal, or an employee trying to convince his or her boss to buy a new software package. In all of these scenarios—and countless others in professional and everyday life—the protagonist must cite important facts, argue in support of conclusions that derive from those facts, counter the claims of others in a principled way, and fully use his or her powers of persuasion—and yet do all of this according to accepted guidelines or rules of argumentation—to convince others of their position or justify a conclusion. Unfortunately, many people are poor arguers (Tannen 1998). They often do not know when they are merely expressing an opinion versus making a claim based on facts. They do not try to rebut the arguments of others but, rather, ignore points of conflict and continue to establish their own argument. Conversely, many people do not recognize when others have rationally rebutted their arguments. These issues highlight an important educational need that cuts across many fields of endeavor, such as the law, science, business, public policy, and, in fact, many aspects of ordinary daily life: the need to teach people to become better at arguing.

Since the time of Aristotle, and even earlier, in Chinese culture, for instance, there have been many frameworks and "rules" of argumentation proposed and identified by mathematicians, philosophers, and researchers. Formal logic, a branch of mathematics, was long the primary province of argumentation (e.g., Whitehead and Russell 1910), while, more recently, naturally occurring human argumentation, which has both an inductive and deductive character, has also been carefully evaluated, especially from a practical perspective (e.g., Toulmin 1958; Kuhn 1991). While the many theoretical and practical approaches to argumentation vary in level of detail, perspective, and specific context of applicability, they can be viewed as sharing at least some basic principles, such as all relevant facts should be considered, claims should be well-grounded, and both supporting and conflicting claims should be considered. Many agree, at least in a general sense, on such "principles of argumentation," as well as the need for more people to understand and employ a principled approach in making their arguments across a wide variety of fields.

So how can more people learn to argue, and argue well? While some students learn to argue simply in the course of their studies, through interactions with their classmates and teachers, support of argumentation learning is missing from most formal courses. Furthermore, even in subject areas where argumentation is an explicit part of the curriculum, such as the law and logic, a teacher's ability to teach argumentation is naturally limited by constraints on time and availability. As a result, many educational technology and learning science researchers have investigated how computer technology can fill this gap and make a difference to students' learning of argumentation, across a wide variety of fields, including the law (Aleven and Ashley 1997; Pinkwart et al. 2006a), science (Suthers et al. 2001; Ranney and Schank 1998; Linn et al. 1998), diagnostic reasoning (Woolf et al. 2005), and conversational argumentation (McAlister et al. 2004; de Groot et al. 2007). Leveraging the inherent advantages of computers (i.e., consistency, widespread use, superior availability as compared to human teachers, never tiring, etc.), these researchers and developers have created *argumentation systems* to relieve some of the burden on teachers to teach argumentation, by supporting learners in creating, editing, interpreting, and/or reviewing arguments.

The field of computer-supported collaborative learning (CSCL) has, in particular, been interested in argumentation and how students can benefit from it (cf. de Vries et al. 2002; Baker 2003; Schwarz and Glassner 2003; Andriessen 2006; Stegmann et al. 2007; Muller Mirza and Perret-Clermont 2009). So-called "collaborative argumentation" (Andriessen 2006) is viewed as a key way in which students can learn critical thinking, elaboration, and reasoning (Andriessen 2006; Bransford et al. 1999). Such a model of argumentation is aimed not at having students learn how to prevail over an opponent, which is perhaps more an emphasis in, for instance, the legal domain, but rather as a means of arriving at an agreed-upon position between members of a group. This type of argumentation is practiced when, for instance, scientists build upon—and sometimes refute—one another's theories and empirical research to arrive at scientific conclusions. However, even though the tenor and goals of collaborative argumentation are different than debate-style argumentation, as in the law, there is still a need, as discussed above, for the participants to understand the "ground rules" of argumentation and to discuss and argue with one another in a rational fashion.

In this paper, we thoroughly review the way in which argumentation has been supported and taught to students using computer-based systems. In particular, we focus on the practice of teaching argumentation through the use of software. By scaffolding good argumentation practices, the systems reviewed in this article not only support students in "learning to argue" but also, in many cases, help students learn about specific domain topics through argumentation ("arguing to learn"). In fact, these different aspects of argumentation are mutually dependent and often not clearly divisible (Koschmann 2003). In other words, the acquisition of argumentation and domain competencies goes hand in hand. Our intent in writing this article was to explore this practice over a range of domains (e.g., the law, science, formal reasoning, conversational argumentation), seeking to identify both commonly used and unique software techniques. We also present empirical results that demonstrate the effect of argumentation tools on argumentation process and learning outcomes. That is, we do not limit our review to systems that are thought of as "educational" (e.g., *Belvedere*: Suthers et al. 2001; *LARGO*: Pinkwart et al. 2006a; *ARGUNAUT*: de Groot et al. 2007) by including systems that are designed less to support learning argumentation and more to support *the actual practice* of argumentation (e.g., *Carneades*: Gordon et al. 2007; *Araucaria*: Reed and Rowe 2004). Finally, while we have a particular interest in collaborative argumentation, this paper also reviews approaches in which students learn argumentation or use argumentation systems in one-to-one (human-to-computer) fashion. Our rationale for widening the scope outside of education and collaboration is twofold: First, many single-user systems are educationally focused and are used collaboratively (i.e., with two or more students sitting in front of one computer) and, second, non-CSCL systems have important lessons to teach about how to build systems that can help people learn to argue.

Our review work has uncovered general themes in past argumentation systems that provides a road map for this paper. First, in the section titled "Argumentation systems," we give the reader a taste of existing argumentation systems, by briefly reviewing some of the most prominent systems. We particularly focus on the systems that are the most sophisticated and elegant in their approach, from the total of 50 systems that we analyzed, along with whether they have led to good results (either learning or performance), and their prominence within the research community. We also have chosen the example software tools as a representative sampling across as wide a continuum of argumentation systems as possible.

In section "Argument representations," we review the form in which argumentation systems have *presented* arguments to human users. Most (but certainly not all) past systems employ visual representations of arguments, in which the students or users contribute to, and/or interact with, graphical representations of arguments (cf. *Belvedere*: Suthers et al. 2001; *Digalo*: Schwarz and Glassner 2007; *Reason!Able*: van Gelder 2002). The nodes of such graph representations are conceptual objects, such as “facts” and “claims,” while the links represent relations between the constructs, such as “supports” or “opposes.” A comparison of different types of representations, complete with the pros and cons of each, is presented in this section.

In “Interaction design,” we discuss various aspects of how students can manipulate and/or create arguments. For instance, some systems allow users to freely create their own arguments (e.g., *Digalo*: Schwarz and Glassner 2007), while others prompt users to analyze arguments extracted from a transcript (e.g., *Araucaria*: Reed and Rowe 2004). Whether a system is collaborative (e.g., *DebateGraph*: http://www.debategraph.org; *AcademicTalk*: McAlister et al. 2004) or single user (e.g., *Athena*: Rolf and Magnusson 2002) is another characteristic discussed in this section of the paper.

Next, in “Ontologies,” we look at how and why different systems define the conceptual primitives used to construct arguments within the underlying target domain. For instance, the legal educational system *LARGO* (Pinkwart et al. 2006a), uses an ontology containing the concepts “test,” “hypothetical,” and “fact situation” with links such as “distinction” and “modification,” while the system *Convince Me* (Ranney and Schank 1998) employs more scientific-focused primitives such as “hypothesis” and “data” elements with “explain” and “contradict” links. Other systems, such as *Rationale* (van Gelder 2007), provide more expansive primitive sets that allow users to construct arguments in different domains.

In “Automated analysis,” we review the means by which past systems have analyzed student-generated arguments and activities, in many cases using artificial intelligence (AI) techniques. For instance, the *ARGUNAUT* system (de Groot et al. 2007) uses both machine learning and AI graph-matching techniques (McLaren et al. in press) to identify important patterns in collaborative argumentation for the purpose of alerting a teacher (or discussion moderator). We characterize these analysis techniques by whether they focus on core argumentation characteristics (e.g., Does this argument correctly follow the Toulmin approach (1958)?) and, second, by whether they focus on communicative characteristics of an argument (e.g., Do the students acknowledge one another’s contributions?).

In “Tutorial feedback,” we discuss past efforts to provide pedagogical support to students in the form of messages or visual highlighting. Feedback issues include *when* to provide feedback (e.g., immediate or delayed), *how* to provide feedback (e.g., highlighting or text messages), and *who* receives the feedback (e.g., teacher or student). For example, the feedback of the *ARGUNAUT* system (de Groot et al. 2007) was provided to teachers for the purpose of helping them guide several ongoing collaborative discussions. Other systems, such as *Belvedere* (Suthers et al. 2001), provide feedback directly to students.

The section “Architecture and technology” focuses on the underlying software architecture of past systems. We discuss how most of the past efforts have, unfortunately, not built upon the software architecture and ideas of earlier systems. In particular, many argumentation system designers and developers have used a “from scratch” approach that has ignored ongoing developments in computer science and software engineering, such as the use of design patterns and reusable components. On the other hand, we cite recent efforts in educational technology research that promote such software practices (e.g., Suthers 2001; Harrer and Devedzic 2002; Devedzic and Harrer 2005; Goodman et al. 2005; Israel and Aiken 2007; Bouyias et al. 2008).

Next, in “Empirical studies,” we review and discuss a variety of empirical studies that have been done with argumentation systems, including, among other aspects, studies that have evaluated the effect on learning argumentation of argument diagrams (e.g., How do diagrams help learning? (Easterday et al. 2007)), different representations (e.g., matrix versus graph versus text (Suthers and Hundhausen 2003)), and adaptive feedback (e.g., Schank 1995). The general finding across these studies is that argument representation and guidance *does* make a difference in helping students learn to argue.

Finally, in the "Conclusions" section, we summarize the lessons learned from this large and impressive body of work on argumentation systems, particularly focusing on lessons for the CSCL research community and its ongoing efforts to develop computer-mediated collaborative argumentation systems.

Argumentation systems

In this section, we introduce some of the argumentation systems that will be discussed in more detail throughout this article. As discussed above, we consider *both* single-user and collaborative argumentation systems because (a) single-user systems are often used collaboratively (i.e., multiple students in front of one computer) and (b) the features and services offered by single-user systems are also potentially useful in collaborative settings and are thus worthy of consideration by designers of collaborative systems. Of the systems introduced in this section, *Belvedere*, *gIBIS*, *QuestMap*, *Compendium*, *Digalo*, *AcademicTalk*, *InterLoc*, *DebateGraph*, and *Collaboratorium* offer built-in support for collaboration, while *Reason!Able*, *Rationale*, *Athena*, *Carneades*, *ArguMed*, *LARGO*, *SenseMaker*, and *Convince Me* are single-user systems.

Perhaps the best known of all argumentation systems is *Belvedere* (Suthers et al. 2001; see Fig. 1), a multiuser, graph-based diagramming tool for scientific argumentation. *Belvedere* exemplifies many of the themes in this review, including argument representations, ontologies, visualization, analysis, and feedback. Initial versions of the tool were designed to engage secondary school children in complex scientific argumentation and provided advisory guidance using Intelligent Tutoring System (ITS) mechanisms to support students in their argumentation and to encourage self-reflection. In later versions, the focus moved from advisory to representational guidance, that is, using specially targeted argument interfaces to bias and guide students' discourse. Another change in focus in a later version was a move from more complex scientific reasoning to simpler evidential argumentation involving evidential relations between data and hypotheses. Figure 1 shows a visualization of *Belvedere* in which argument contributions (e.g., a data element "Linux is free," a hypothesis "For common people Windows is preferable") are represented as nodes, and links between the nodes represent the relationship between the contributions (e.g., "Linux is free" is opposed to "For common people Windows is preferable"). This style of argument visualization is the predominant one found among the argumentation systems discussed in this paper.

Argumentation is a central concern of the teaching of philosophy and, thus, a number of key educational argumentation systems come from this tradition, including *Reason!Able* (van Gelder 2002, 2003), its commercial successor *Rationale* (van Gelder 2007), and *Athena* (Rolf and Magnusson 2002). *Reason!Able*, for instance, is an argument diagramming system that supports "rapid and easy construction, modification and evaluation of argument visualizations" (van Gelder 2003, p. 106). It is a key part of the *Reason!* method aimed at teaching reasoning and argumentation skills.

While *Reason!Able, Rationale*, and *Athena* can be used for both argument analysis and production, *Araucaria* (Reed and Rowe 2004) aims at the analysis of arguments, provided as textual transcripts. Primarily focused on research contexts, *Araucaria* can also be used in

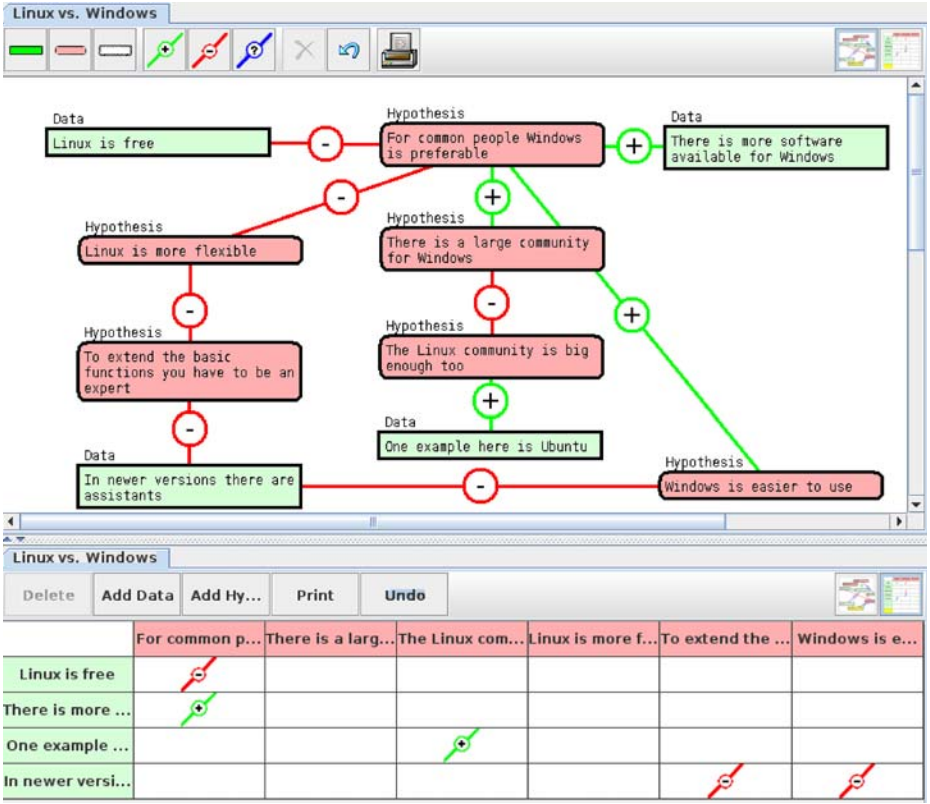

Fig. 1 Belvedere with a graph-and-matrix interface

educational settings, as an aid to argumentation and critical thinking courses. System features include the support of three different diagrammatic notations for laying out arguments (Standard, Toulmin (1958) and Wigmore (1931) notation), translations between the notations, and the support of different argumentation schemes (Walton et al. 2008).

Some argumentation systems can process the constructed diagrams to automatically derive acceptability values for argument elements. *Carneades*, for instance, "supports a range of argumentation tasks, including argument reconstruction, evaluation and visualization" (Gordon et al. 2007, p. 875). Although it is conceptualized as a domain-independent tool, it is primarily aimed at legal argumentation. It makes use of a formal, mathematical model to compute and assign acceptability values to propositions and supports multiple proof standards, that is, different procedures to derive the acceptability of a claim and associated arguments such as "preponderance of evidence." Similar decision procedures are implemented in *ArguMed* (Verheij 2003), an argument assistant system that also focuses on the legal domain, and *Hermes* (Karacapilidis and Papadias 2001), a system that supports collaborative decision making in threaded discussions.

In contrast to *Carneades*, *ArguMed*, and *Hermes*, the Intelligent Tutoring System *LARGO* (Pinkwart et al. 2006a) focuses on *teaching* law students legal argumentation skills. The student's task is to analyze existing U.S. Supreme Court Oral Arguments by "translating" provided transcripts into a graph-based visual representation, supported by an advice-upon-request function. *LARGO* uses a special-purpose argumentation model and representation that focuses on *hypothetical reasoning*, that is, legal parties propose tests that interpret laws, legal principles, or precedence cases in a specific way to decide the current situation at hand; these tests are challenged by suggesting competing hypothetical situations (*hypotheticals*) in which the proposed test leads to an undesired outcome. This then, in turn, may lead to abandoning or modifying the test.

Another well-known tool is *SenseMaker*, which supports the construction of scientific arguments, either individually, or in small groups of students in front of the same computer. It uses a specific representation consisting of hierarchically nested frames (or “containers”) to make “student thinking visible during individual and collaborative activities in the classroom” (Bell 1997, p. 10). Figure 2 shows a screenshot of *SenseMaker*. Hyperlinks (underlined texts in Fig. 2) represent evidence that students collect from World Wide Web resources (e.g., “Bicyclists at night” in Fig. 2). Frames represent claims (e.g., “THEORY 1: Light Goes Forever Until Absorbed (LGF)” in Fig. 2). Elements (i.e., hyperlinks and frames) are placed into frames to indicate support for the corresponding claim. *SenseMaker* is part of the *Knowledge Integration Environment (KIE;* Bell and Linn 2000), which was developed for middle and high school science instruction, and its successor, the Web-based Inquiry Science Environment (*WISE*; Linn et al. 2003). Both environments offer a range of scientific inquiry activities and tools, one of which is argument diagramming.

Convince Me (Ranney and Schank 1998) is another tool that supports scientific reasoning in educational contexts. Students use *Convince Me* to lay out and evaluate arguments based on scientific principles by defining the argument structure and specifying believability ratings for individual argument elements. To receive system feedback on the quality of their arguments, students run a simulation to check whether their ratings are in line with those of a computational model of reasoning called *ECHO*. In initial system versions, students typed arguments in a form-based interface; later versions were enhanced with a graph-based argument representation.

Another strand of argumentation tools have their origin in the *IBIS methodology* (Issue-Based Information System; Kunz and Rittel 1970), which aims at supporting and recording decision-finding processes for “wicked problems” (Rittel and Webber 1973) which occur in

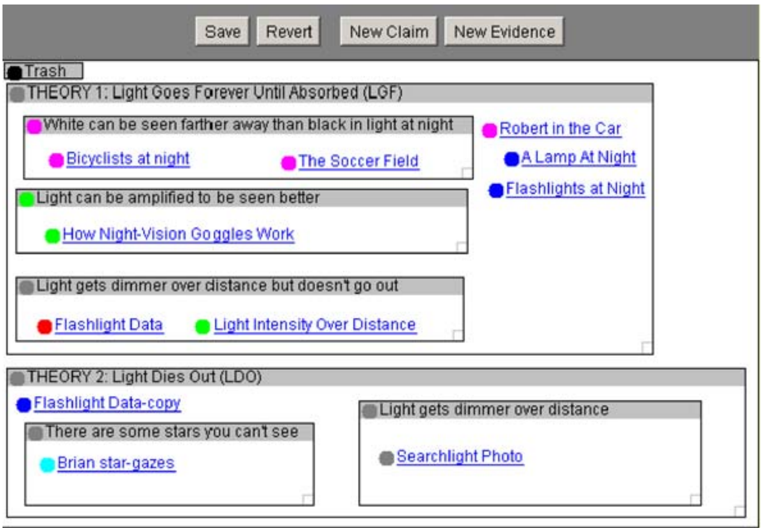

Fig. 2 SenseMaker with a container visualization. The shown argument is based on Fig. 1 in Bell (1997)

design and planning. *gIBIS* (Conklin and Begeman 1988) is an early computer-based implementation of IBIS in which small groups of users collaboratively construct a graph to solve design problems, while simultaneously capturing the design rationale. More recent systems in this tradition are *QuestMap* and *Compendium* (Buckingham Shum et al. 2006). The use of these tools is neither restricted to design problems nor to cooperative work scenarios. *QuestMap*, for instance, has also been used to teach legal argumentation (Carr 2003). More recently, *Compendium* has been used in schools for scientific argumentation (Okada and Buckingham Shum 2008).

Digalo (Schwarz and Glassner 2007) is an argumentation system that was designed for classroom use in which groups of three to seven students discuss a controversial topic, for example, “What do you think about doing experiments on animals?” *Digalo* has configuration options to allow flexible usage in different scenarios. While it has been used primarily for informal discussions, it can be configured to support more formal and domain-specific discussions. *Digalo* is one of the e-discussion tools supported in the *ARGUNAUT* system, which provides “a unified mechanism of awareness and feedback to support moderators in multiple e-discussion environments” (de Groot et al. 2007, p. 165). *ARGUNAUT* is aimed at supporting a teacher while he or she moderates multiple, simultaneous group discussions.

AcademicTalk (McAlister et al. 2004) takes a somewhat different approach to supporting small-group discussions, by promoting peer learning via a user interface with threaded discussions and predefined sentence openers. The system is based on *dialogue game theory*, which is concerned with models of desired dialogue practices with participant roles, valid discussion moves, corresponding sentence openers, and rules of interaction. The specific dialogue game implemented in *AcademicTalk* is a critical reasoning game aimed at promoting critical discussions and reasoning. Its successor is *InterLoc* (Ravenscroft et al. 2008), which has a more sophisticated and usable user interface, supports a wider range of dialogue games (e.g., a creative thinking game), and allows the user to configure his or her own dialogue games.

Large-scale argumentation, involving many participants, perhaps as many as hundreds or even thousands, is another model supported by recent systems. This type of system is distinct from any of the above, which are all used by either a single user or a small group. One example of a large-scale argumentation system is *DebateGraph* (http://www.debategraph.org), a wiki debate visualization tool for public debates about various topics. It has support for asynchronous argumentation and large community use. It offers an open, Web-based forum for public deliberation and provides multiple argument representations, such as graphs and threaded text. Another example is *Collaboratorium* (Klein and Iandoli 2008) a Web-based system that supports users in debating subjects of global importance, such as climate change (Malone and Klein 2007), through the creation of argument maps. To guide the argumentation, moderators evaluate posts for correct structure and validity. The *Collaboratorium* also supports the notion of “collective intelligence” by allowing participants to rate contributions; the highest-rated contributions are considered the community’s decisions.

We now turn to the specific features and affordances of these systems. In the discussion that follows, all of the above systems will be discussed again, in further detail, along with many other argumentation systems from the literature.

Argument representations

A key goal of argumentation technology is to provide an *external* argument representation to allow users to create, manipulate, and review arguments. Argument representation formats are crucial to the interface between user and system, making different pieces of information more or less salient. System designers can exploit these properties to create user interface affordances that guide the user toward productive activity, as described in Suthers' (2003) theory of representational guidance.

A variety of different representational formats have been employed in existing systems. Some formats aim primarily at supporting communication between users while others aim to represent the conceptual structure of debate; some are used for education while others are used for collaborative decision making and deliberation; some aim at argument production, while others are used for argument analysis and evaluation. In this section, we discuss the five major representational types found in the literature: (1) linear, (2) threaded, (3) graph-based, (4) container, and (5) matrix.

The simplest form of argument representation is a *linear*, usually textual, form. Simple computer-mediated communication (CMC) tools like chats (e.g., IRC) are used for this form of argumentation. Chat can be thought of as a written analog to (sequential) spoken communication. The main advantage of such tools is their ease of use and familiarity. A problem with chat, however, is *sequential incoherence*, especially in chats of more than two or three participants (McAlister et al. 2004). This problem occurs when it is not clear which comments and responses refer to which other comments. An example illustrates this:

(10:01:12)—Alice: I like Windows, because I like to play games
(10:02:22)—Bob: There are at least three important systems: Windows, Linux and MacOS.
(10:02:23)—John: Which one do you like most?

Here, John intends to ask Alice which game she likes most. However, because of Bob's intervening statement, this is not immediately obvious without a careful review of the time stamps. *Threaded* discussions, on the other hand, explicitly capture message-reply sequences, avoiding sequential incoherence, and better support users in managing large argumentation strands. While threaded discussion tools do not provide special support for argumentation, there have been attempts to support argumentation by tagging according to type, that is, annotating the role of contributions in an argument. An example for this is *Hermes* (Karacapilidis and Papadias 2001), which uses a forum-like style in which contributions are marked as, for instance, issues, alternatives, and pro and con positions. The contribution categories largely differ between argumentation domains and tool purposes; this is discussed in more detail in the section "Ontologies."

Most argumentation tools, educational as well as general purpose, do not use purely textual argument representations, such as linear or threaded texts. The most frequently employed representation is a *graph* style (see Figs. 1, 2, 3, and 4). In this approach, contributions are displayed as boxes or nodes that represent argument components, such as claims or facts. The edges (or arrows) of these graphs represent relations between the argument components (e.g., supports or refutes). There are notable differences between the kinds of argument graphs that can be created with the existing systems: Some systems use proper graphs, while others use hyper-graphs, that is, graphs with links between links. For instance, *Belvedere* allows a user to construct conjunctions of propositions with an "and" link, which can, yet again, be related to other nodes or links. Some systems impose no restrictions on the linking of parts, while others permit only tree structures, that is, special cycle-free graphs with a single root element (e.g., the main hypothesis or question). Graph representation systems are numerous, including *Belvedere*, *Athena*, and *Digalo* in the unrestricted variant and *Reason!Able*, *Araucaria*, and *Carneades* in the more restricted tree-based variant.

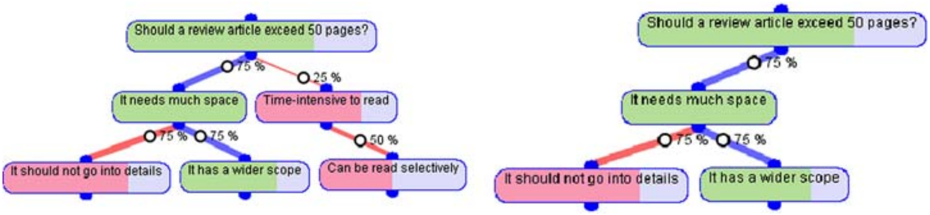

Fig. 3 Acceptability filter in *Athena* (0% vs. 50%)

A key asset of graphs is their explicitness and clarity, that is, each elementary knowledge unit is encapsulated as an individual node and relations between knowledge units are explicated with edges. The different types of knowledge units and relations can be easily distinguished via their visual appearance. This explicitness makes graph-style representations an excellent and intuitive knowledge modeling approach, for instance, for analyzing argument transcripts (e.g., *LARGO* and *Araucaria*) or in keeping a record of the current state of a debate to support decision making (cf. van Gelder 2003; Buckingham Shum et al. 2006). Others use graphs as a structured medium of debate (e.g., Schwarz and Glassner 2007) or as representational tools that provide an external and persistent stimulus to encourage and scaffold discussions, that is, students do not discuss *within* a graph but *about* a graph (Suthers 2003). The suitability of graphs as a medium for argumentation interacts with other factors, such as the complexity of the underlying ontology and the number of discussants and synchronicity of communication. Graphs may become unwieldy when a relatively large number of participants engage in synchronous discussion using a complex ontology. Also, the user group and domain may have an impact on the suitability of graph notations: Hair (1991) reports changing the graph representation in the *Legalese* argumentation tool (designed to assist lawyers in constructing arguments) because lawyers strongly preferred threaded text representations.

Another approach to structuring discussions and argumentation visually is used in *SenseMaker* (Bell 1997), which visualizes argumentation strands belonging together graphically via frames (windows) which serve as *containers*. Here, each frame represents a claim that is supported by the evidence elements and other claims contained in that frame. Elements (claims and evidence) can be annotated with “usefulness scores,” which are represented as colored dots (e.g., red for high; see Fig. 2). Other examples of the container

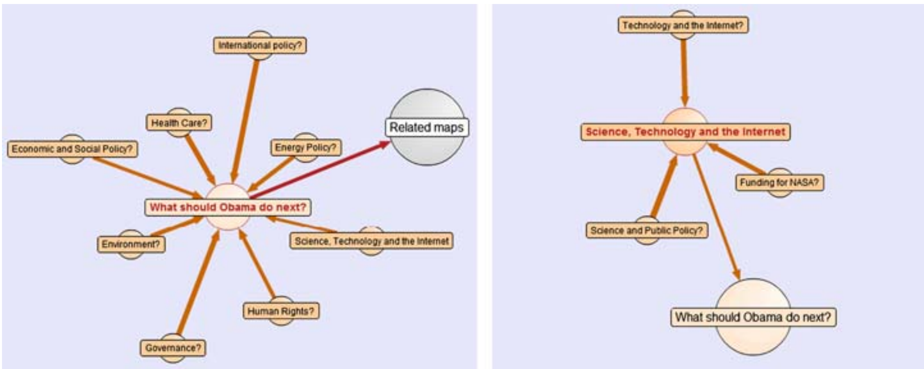

Fig. 4 Local views in *DebateGraph* (*left*: upper level, *right*: lower level)

visualization technique are the wiki-based *Debatepedia* (http://wiki.idebate.org) which provides one frame per question, containing arguments pro or con the topic, and *Room 5*, a system for legal argumentation (Loui et al. 1997).

The key advantage of the containers style is the possibility of recognizing, at a glance, argument components that belong together. Yet, at the same time, this technique makes it difficult to express types of relations, because relations are expressed implicitly by one element visually residing within another element. Also, one is not able to recognize whether two components in the same frame are related to each other. Additionally, with this type of visualization, it is hard for the user to get an overview of what is happening in a large argumentation map (a similar problem to the graph style).

An attempt to visualize implicit or missing relations between argument elements is the use of a *matrix* argument representation in which argument components (e.g., hypotheses, facts) are the rows and columns of the matrix while the cells represent the relations between the components. This visualization method has, to our knowledge, only been implemented in the *Belvedere* (v3 and v4) system (see Fig. 1). The primary strength of a matrix in representing arguments is highlighting the missing relations between important aspects of arguments (Suthers 2003). However, the creation of arguments via a matrix style is more abstract (and perhaps less intuitive) than, for instance, constructing a graph. Furthermore, it is not possible to represent links between links in the matrix style.

Some systems support multiple visualizations, that is, different views of the same argument, each of which (possibly) focuses on different aspects with different affordances. These views are sometimes provided simultaneously: In the last published version of *Belvedere* (4.1), for instance, it is possible to use both a graph view and a matrix view at the same time (see Fig. 1) and hence, to (potentially) benefit from their different strengths (see Table 1). Other systems also support multiple visualizations but not simultaneously, that is, the user has to switch between the different views. Examples include *DebateGraph* (http://www.debategraph.org), which offers a graph style and a threaded textual representation, *Araucaria*, which allows the user to switch between three representational notations that differ in terms of layout and ontology, and *CoPe_it!* (Karacapilidis et al. 2009), which offers views that differ in terms of formality and available actions to support different argumentation phases (e.g., brainstorming versus decision making).

Whereas the above approaches offer different views of the same argument, other systems provide different representational tools to support different argumentation-related activities. Some systems offer a more structured visualization style (e.g., graphs) for knowledge modeling together with more lightweight CMC facilities (e.g., chat) for communication purposes. For instance, Munneke et al. (2003) combined the use of chat and the collaborative writing of an argumentative text using the tool *TC3 (Text Composer, Computer-supported, and Collaborative)*; Lund et al. (2007) provided a chat and graph-based tool in parallel during a discussion phase using the CSCL environment *DREW* (*Dialogical Reasoning Educational Webtool*); Suthers et al. (2008) have investigated how to integrate graphs as a conceptual knowledge representation with a chat and threaded discussion as the actual communication medium. Multiple representations can also be used in different phases of a longer argumentation task. For example, the *CoChemEx* system (Tsovaltzi et al. 2010) uses a more structured graphical representation for planning problem-solving activities and the interpretation of results while using chat to coordinate and deliberate during the problem-solving activity itself. A second example is the work of Lund et al. (2007), in which students first debate in a chat (as medium) and then represent the debate as a graph. Some systems offer configuration facilities to define when tools are available during the learning process. An example is *CoFFEE*

Table 1 Comparison between argument visualization styles

Representation style	Typical uses	Pros	Cons
Linear (e.g., chat)	- Discussions (especially synchronous)	- Familiar and intuitive to most users, easiest to use - Best to see temporal sequence and most recent contributions	- Risk of sequential incoherence (McAlister et al. 2004) - Not suited to represent the conceptual structure of arguments - Lack of overview
Threaded (e.g., forums, *Academic Talk*)	- Discussions (especially asynchronous) - Modeling	- Familiar and intuitive to most users, easy to use - Easy to manage large discussions - Addresses issue of sequential incoherence	- Moderately hard to see temporal sequence (because of multiple threads) as compared to Linear - Limited expressiveness (only tree-like structures)
Graph (e.g., *Belvedere, Digalo*)	- Discussions - Modeling	- Intuitive form of knowledge modeling (Suthers et al. 1995) - Highly expressive (e.g., explicit relations) - Many graph-based modeling languages exist	- Hard to see temporal sequence - Lack of overview in large argumentation maps (need a lot of space, can lead to "spaghetti" images (Hair 1991; Loui et al. 1997)
Container (e.g., *SenseMaker, Room 5*)	- Modeling	- Easy to see which argument components belong together and are related	- Limited expressiveness (e.g., only implicit relations, only tree-like structures) - Lack of overview in large argumentation maps because of missing relations
Matrix (e.g., *Belvedere*)	- Modeling	- Easy systematic investigation of relations - Missing relations between elements are easily seen (Suthers 2003)	- Limited expressiveness (e.g., supports only two element types (row, column), no relations between relations) - Uncommon (Non-intuitive) way of making arguments

(*Collaborative Face to Face Educational Environment*; Belgiorno et al. 2008). *CoFFEE* provides a chat to send quick messages, a threaded forum for more structured discussions, a graphical discussion tool for collaborative brainstorming, and a collaborative writing tool.

The use of multiple representations is a challenge for data analysis in empirical studies as well as for automated, system-triggered analyses especially when students do not use the representations in expected ways. For instance, students might distribute their activities across multiple, available channels without considering the intended use of each channel. In such cases, it is necessary to integrate data from the different sources to evaluate the overall argument. In section "Empirical studies," we review a number of studies concerned with the topic of using multiple representations.

Interaction design

In this section, we discuss different interaction design techniques used in argumentation systems. We describe how users can interact with argument representations and other interface widgets of argumentation systems, emphasizing the important aspects of computer-mediated human-human interaction and visual representations. Depending on intended purpose, different systems offer different modes of argument creation and manipulation, support individual use or collaboration, allow for synchronous or asynchronous student-student interaction, and provide specific features to help users.

First, we address the creation of arguments. Arguments can be thought of as sets of interrelated components: basic, self-contained "argument moves" made by arguers (e.g., a claim, a hypothesis). There is no general consensus in demarcating elementary units, that is, where one unit ends and another begins. This decision often depends on the specific objectives of system and users (Reed and Rowe 2004). One way to scaffold users is to provide a domain ontology, that is, categories or sentence openers that define elements of interest (see section "Ontologies"). Here, we review another dimension, namely the *amount of autonomy* that the user has in deciding on the content of her argument components. This point is critical for learning applications, because different approaches may promote different learning goals such as argument production, analysis, and evaluation. We identified the following five classes of argument construction:

1. *Free-form arguments*: The students (or users) are free to create elementary argument components on their own without restrictions, except for a possibly predefined topic. (Example systems: *Digalo, Athena*).
2. *Argumentation based on background materials*: Background materials are given to promote the argument. Based on the given materials, students create and interrelate elementary argument components. For instance, one version of *Belvedere* provides a collection of hypertext pages about scientific controversies. Similarly, students in *SenseMaker* use World Wide Web resources as pieces of evidence in their arguments.
3. *Arguments rephrased from a transcript*: Students review an already existing argument and, based on this, reconstruct the argument in a more structured form. For instance, *LARGO* provides a written protocol from an oral legal argument, which is analyzed by students and encoded as an "argument map." In contrast to a corpus of background materials, a transcript is a much more focused resource that already contains the complete argument—yet, in a complex form that is not easily understandable.
4. *Arguments extracted from a transcript*: Here, students are given a transcript to analyze, but instead of paraphrasing the existing argument, they are prompted to copy passages directly from this transcript to elementary argument elements (Example: *Araucaria*).
5. *System-provided knowledge units*: Elementary components are predefined. The student's task is to choose from the set of predefined components and connect them in an appropriate way to define how they relate to one another. (Example: *Belvedere* v2, Suthers et al. 2001).

Approach (1) clearly aims at supporting argument *generation* skills. Approaches (3) and (4) are typically used to train argument *analysis* skills. Approaches (2) and (5) support both *generation* and *analysis* skills. The different approaches to argumentation support vary in the degree of user and system control: In approach (1), users have complete control of the contribution content. In approaches (2) and (3), users are in control of the contribution content, but their work relies on given background materials or a transcript. In approach (4), users only decide on the segmentation of information; the actual textual content is taken from the transcript. Finally, in approach (5), elementary knowledge units are completely predefined for the user.

These argument approaches are not mutually exclusive: Some systems integrate several approaches. One example is *Rationale*, which supports freely created arguments as well as arguments extracted from a transcript (in *essay planning mode*). *Belvedere v2* combines approaches (2) and (5), that is, students can make use of predefined knowledge units but can also write their own contributions, possibly based on given background materials. *Araucaria* mainly uses approach (4), but also supports the user in freely entering elements to add reconstructed information that is only implicitly contained in the transcript.

Approaches (4) and (5) have the advantage that the system "knows" the meaning of elementary knowledge units used by the student because they are predefined (approach 5) or can be easily traced back to a transcript passage that is "known" in advance (approach 4). *LARGO* (approach 3) achieves the same effect by letting the student explicitly define which transcript passage is encoded in a diagram element using markup techniques. Making such information accessible to the system makes it much easier to automatically analyze the produced arguments. (We will return to this point in section "Automated analysis.") Graphical argumentation systems sometimes restrict not only the creation but also the visual arrangement and structural relationships of contributions in the workspace. Conversely, many systems allow the user to create contributions and relate them to one another freely. Thus, the layout and structure depend on the users' preferences and choices and can carry a user-intended meaning (e.g., arguments in close proximity to one another belong together, even if they are not formally connected). This method is used, for example, in *Athena*, *Digalo*, *Convince Me*, *LARGO*, and *Belvedere*.

Another approach is to have the system control the layout. For instance, *Reason!Able* forces a Premise-Conclusion tree structure with "objecting" and "supporting" arguments as the children of the referred claim. *Araucaria* enforces argument layout according to Walton, Wigmore, and Toulmin diagrams. Such approaches lower the risk of cluttered arguments and ensure adherence to syntactic constraints, which is required by some automated analyses (e.g., in *Carneades*, see section "Automated analysis"). Of course, restrictions on layout may not be appropriate in all cases. In particular, the more restrictive, system-controlled approaches are more typically seen in formal modeling scenarios. Advantages and disadvantages of the user versus system-controlled approaches, together with characteristics of the approaches (that can be either pro or con) are summarized in Table 2.

Table 2 Comparison: Layout control in graph style

Layout control	Characteristics	Pros	Cons
System-controlled	- System "guides" user through the construction - Some diagram aspects cannot be manipulated directly by the user	- Clear and consistent layout - Easy to read - Avoids unwanted syntactic constructs (e.g., cyclic arguments)	- Too restrictive in some pedagogical scenarios (errors and weaknesses as opportunities to teach)
User-controlled	- Diagram clarity and consistency depend on user's proficiency and discipline - User can directly manipulate all diagram aspects	- More flexible	- Danger of cluttered diagrams - Unwanted syntactic constructs possible

One of the critical differences between systems is whether collaboration is supported and, if it is, what types of student-student interactions are supported. In particular, while single-user argumentation tools support knowledge construction and learning of argumentation constructs, collaborative argumentation systems typically provide learners with insights into the opinions, viewpoints, and knowledge of other students. The range of collaboration options that are available are:

1. *Single-user argumentation systems* are software modeling tools that help individuals structure their thoughts and/or prepare argument representations. Some systems provide only modeling facilities, while others actively provide feedback (Examples: *Convince Me, LARGO [with feedback]; Athena, Araucaria, Carneades* (*without feedback*))
2. *Small group argumentation systems* serve as a software mediator between a relatively small number of learners (typically 2 to 5) and offer (typically) synchronous communication and/or collaborative modeling tools. Users may profit from both interaction with the system and with other users, developing general argumentation skills, discussing different points of view, and/or learning skills of persuasion (Nussbaum et al. 2007; Keefer et al. 2000). System-generated feedback can support both argument aspects and communication aspects. (Examples: *Digalo, QuestMap, Belvedere, AcademicTalk*)
3. *Community argumentation systems* are, in many respects, similar to small group argumentation systems but with support for a larger number of participants (and, typically, a larger number of contributions). The larger number of users puts additional constraints on the system: Communication is typically asynchronous to avoid coordination problems. The representational format enforces a more rigorous organization of contributions, that is, discussion/argument threads are preferable over graphs. (Examples: *DebateGraph, Collaboratorium:* Klein and Iandoli 2008)

Of course, argumentation systems that focus on scaffolding individual work can also be used by small groups that share a single computer (cf. van Gelder 2003), and multiuser systems can also be used by single users (e.g., *Digalo*). Single-user and collaborative activities are sometimes part of the same activity sequence: In some phases, individual work may be beneficial (e.g., the initial representation of one's point of view), while in others, collaboration with other students may be more fruitful (e.g., discussion of different points of view). In section "Empirical studies," we look deeper into the issue of phased activity sequences and their impact on learning and system use. When users try to collaborate in synchronous mode while spatially separated, they might run into problems due to a lack of coordination. One way to address this problem is to use *floor control* to guide turn taking. For instance, in one configured use of *Digalo* (Schwarz and Glassner 2007), students must explicitly request control over the shared workspace, then perform their actions (e.g., add a new contribution), and finally release control to let the next student take a turn (a study by Schwarz and Glassner concerning the effects of floor control is discussed in section "Empirical studies"). Some systems allow "indirect" user-to-user interactions. In *LARGO*, for instance, student-student interactions take place only by students rating contributions of their peers; in *Araucaria*, users can share the diagrams they produced when analyzing a transcript via a central server.

Another important aspect of interaction is the management of large argument maps. Over time, an argument can grow quite a bit, thus making it difficult for students to follow and review. For instance, Karacapilidis and Papadias (2001) reported that in the *Hermes* system, most of the discussions comprised 20 to 40 contributions. Because graph representations typically use a lot of screen space for their nodes and links, it is important for an argumentation system to offer interaction techniques that assist users in maintaining the overview of a larger ongoing argumentation, while at the same time allowing users to focus on specific (and smaller) parts of the argument. The most common techniques for summarizing large, graph-style maps are zooming and scrolling (e.g., *ArguNet*: Schneider et al. 2007; *Rationale*, and *Athena*). However, dependent on screen size, users of zooming and/or scrolling displays are unable to read or edit the content of an argument map if, for instance, the zoom-in level is too low. On the other hand, scrolling through large maps can cause the user to lose the "big picture."

One of the more sophisticated methods is a mini-*overview map*; a smaller version of the entire argument, typically put in one corner of the screen. One example of a system that uses this interaction technique is *Rationale*. This approach has the advantage that it can be used in combination with other techniques such as zooming. However, while it is helpful for getting a map overview, a mini-overview map does not differentiate between important and less important argument parts, which users may want to treat differently. *SEAS (Structured Evidential Argumentation System*; Lowrance 2007) uses "starburst" and "constellation" presentations to summarize arguments; unfortunately, these representations likewise do not distinguish between what is central in a large map and what is not.

Filtering—Fading selected parts of an argument based on a specific rule is another interaction technique for managing large arguments. It is used in *Athena, Convince Me*, and *Digalo*, for instance. In *Athena*, it works as follows: After building an argumentation map, the user can assign scores regarding the acceptability of a statement and relevance of a relation between two elements. In Fig. 3, the filling level of nodes (i.e., the degree to which they are colored) shows the acceptability of statements, and the relevance level for relations is shown through the percentages on the edges. Based on these values, *Athena* calculates an overall score per element by aggregating the acceptability values of child elements, weighted according to their relevance. Elements with scores below a user-defined threshold are filtered out. For example, Fig. 3 shows a map without filtering (0%) as compared to the same map with 50% filter. Especially for large graphs, such a mechanism can be useful in helping users focus on the main points of an argument. Other systems, such as *Digalo*, allow filtering by other criteria such as time stamps, contribution types, or author of nodes.

Another technique for reducing the visual complexity of large arguments is the use of *local views* that hide portions of arguments based on a "distance" to an explicitly set focus point. A prominent example that employs this technique is *DebateGraph*. In this system, which makes use of a tree-based visualization, users see only the child nodes of their current main focus. The child nodes can also be used for navigation: When a user clicks on a node, he or she will see one level deeper in the tree, plus the parent node for reverse navigation (see Fig. 4, which shows navigation to a "Science, Technology and the Internet" sub-argument). A similar interaction technique is frequently used in structured discussions and forums, in which users can expand different threads.

Ontologies

Ontologies—explicit specifications of a conceptualization (Gruber 1993)—provide the foundation for most argumentation systems. Such representational systems describe the components of arguments, together with relations between components and modifiers of the components and relations (such as scores or relation relevance values, as shown as percentages on the links in Fig. 3). While ontologies have most typically been used for computational reasons in Artificial Intelligence systems, for instance, as tools for automated logical inference, argumentation systems also use ontologies, with the aim of making users aware of the conceptual components in the task domain (Suthers 2003).

Ontologies may combine theoretical perspectives (e.g., Toulmin model, 1958) with pragmatic considerations (e.g., understandability by a specific target group). For instance, Stegmann et al. (2007) simplified the original Toulmin model, perhaps the most well-known and popular argument ontology, to improve the usability of their argumentation interface. Suthers (2003) changed the "perspective" of Belvedere to a simpler version of evidential reasoning; this was done based on the observation that students had problems using the more expressive ontology. Schwarz and Glassner (2007) say that such ontologies that result from an evolutionary and reflected process can be classified as *educated ontologies*; they are learned in schools and universities in the form of definitions and rules. This contrasts with *informal ontologies*, which are based on reasoning that typically occurs in natural conversations. While educated ontologies seem especially appropriate for argument modeling, their informal counterpart may be more suited to support structured—and typical less formal—communication. One variation is sentence-opener interfaces, which do not explicitly expose categories but which scaffold new contributions through predefined sentence-starting phrases. Typically, these interfaces are based on an underlying model of desired communication acts and processes, for instance, dialogue games (McAlister et al. 2004). One general problem that communication ontologies and sentence openers strive to address is to help students stay on topic by limiting user options.

There are a number of argumentation systems that aim at supporting a broad range of various argumentation domains with a "one-ontology-fits-all" approach. Interestingly, the ontological approaches between these systems differ considerably. One extreme case is *Rationale*, which employs a fine-grain ontology with approximately 30 different contribution and relationship types. While such a general approach has appeal, some studies show that such wide-ranging ontologies may confuse users with a "plethora of choices" (Suthers 2003, p. 34). Other researchers have found that users of argumentation software tools are able to use effectively fairly large ontologies—of more than, say, ten elements—if "labeling" of the contributions is done through titles of contributions, and not by selecting specific contribution types from a palette (Jeong 2003). The other extreme is coarse-grain ontologies with unclassified nodes and a limited number of link types (e.g., pro and con), as in *Athena*. Such an approach is, of course, quite general, but also limited in expressiveness. A compromise is a basic set of elements that is extensible, as in, for instance, *Aquanet* (Marshall et al. 1991). This approach has the advantage of flexibility without overwhelming users. Yet, while quite flexible, this approach may make automated analysis techniques more difficult to implement. Table 3 compares the ontologies of *Athena* and *Rationale* with respect to their contribution and relationship types.

In general, the absolute number of ontological elements is not the only important factor. The elements' understandability, their differentiability (i.e., Is the difference in meaning between elements clear?), their organization in the user interface (i.e., Are they organized in a clear and consistent way, for instance, by grouping related elements together?), and synchronicity of interaction (i.e., How much time does the user have to choose?) are also important. The development of a suitable ontology is a critical aspect in the design of an argumentation system and might involve iterative refinement based on observed problems and weaknesses (Suthers et al. 2001; Buckingham Shum et al. 2002).

While the ontologies discussed above are intended to be quite generic, there are also argumentation tools that provide *domain-specific ontologies*, that is, representations that provide support for the particular requirements of a specific domain. For instance, argumentation in the legal domain is quite structured and specific, with well-defined types, roles and "outcomes" (the decision of the case). These domain characteristics are then also reflected in some argumentation ontologies (e.g., Wigmore (1931) diagrams in *Araucaria*).

Table 3 Ontologies—different approaches to achieve generality

System	Contribution types	Relationship types
Athena	Node	Unclassified: Link
Rationale[a]	*Grouping*: box; *Reasoning*: contention, reason, objection; *Advanced Reasoning*: contention, reason, objection, co-premise; *Basic Boxes*: basis, assertion, by definition, case study, common belief, data, event, example, expert opinion, law, media, personal experience, publication, quote, statistic, web; *Teacher tools*: Feedback notes (general, claim, structure, evidence, evaluation, blank); *Extras*: note, topic, question, option, pro, con, idea, consequence, information required	General links that are classified automatically depending on the types of nodes they connect

[a] Rationale organizes contribution types in its user interface according to a common theme. The table shows these themes in *italic* letters.

Table 4 shows the domain-specific ontologies of the systems *LARGO*, *Convince Me*, and *AVERs* (Bex et al. 2007).

Differences between ontologies in argumentation systems derive not only from domain differences, but also from different conceptualizations and perspectives of system designers. Even systems targeted at the same domain can, and do, have different ontologies. For instance, although *Carneades* and *LARGO* are used to model legal arguments, they have different ontologies: *Carneades* models arguments as Premise-Conclusion trees with proof standards assigned to propositions, while *LARGO* takes the perspective of hypothetical reasoning. Thus, ontologies are not unique even within a single domain.

Independent of an ontology's granularity and generality, there are different property types used in ontologies. Together, these properties define the aspects of an argument that are represented in a system. The properties can be divided into the following categories: (1) *Content data*, which contains basic argument data such as title, label, and argument text; (2) *Evaluation data*, numeric or categorical values to assess the believability or relevance of an element. Such data can be used, for instance, for automated filtering of content. Other types of evaluation data are, for example, the assignment of *proof standards* (Gordon and Karacapilidis 1997) as used in *Carneades* or argumentation schemes (Walton et al. 2008), as used in *Araucaria*, to mention just a few; (3) *Awareness/collaboration information*, which is important in multiuser systems to assure that each participant is able to see, for instance, who has added each contribution; (4) *Technical metadata*, such as unique identifiers for argument elements.

Table 4 Ontologies—domain dependent elements

System	Domain	Contribution types	Relationship types
LARGO	Law/hypothetical reasoning	Test, hypothetical, current fact situation	*Model-specific relations:* Test modification, Distinguish hypothetical, Hypothetical leading to test change *Unclassified:* General relation
Convince Me	Science	Hypothesis, Evidence	Explanation, contradiction
AVERs	Crime investigation	Claims about legal cases (Pro/Con), Quotes from source documents, Inferences, Schemes	Directed links

Furthermore, there are also tools that support *multiple ontologies*. To support this approach, there must be a shared underlying format, flexible enough to support different ontologies and the connections (or transformations) between them. One such example is *Araucaria*, which allows switching between notations based on the Walton, Toulmin, and Wigmore models (Bex et al. 2003; Wigmore 1931). The main advantage of this approach is that one can choose the ontology that best fits a topic, but there is a risk that students might be confused by different notations and choose an inappropriate notation for a task.

Automated analysis

As we have seen, argumentation tools can guide students' activities in a fruitful direction via specially designed user interfaces and interaction models. But what about more *active* forms of support? In this section, we discuss automated analysis and diagnostic techniques that have been applied in argumentation systems.

Argumentation systems can be roughly grouped into two categories of analysis approaches: *argument analyses* and *discussion analyses*. Argument analysis systems are concerned with the construction of sound and syntactically valid arguments and the reconstruction of existing arguments (see section "Interaction design"). With these systems, users typically create an argument diagram as a semiformal model of an argument, for instance, as a network representing the inferential relations between propositions. These diagrams can then be analyzed to identify weaknesses and errors, to find opportunities to encourage reflection, to suggest further steps, or to simulate reasoning processes within the represented model. Discussion analysis systems, on the other hand, are mainly concerned with social and interaction aspects of discussions. For instance, successful knowledge sharing, responsiveness of the participants, and resolution of differences of opinion are often important foci of these types of systems. The consideration of social and interactional aspects is in line with pragmatic theories of argumentation that emphasize the dialogical context of arguments (e.g., Walton 2008, pp. 3–8, describes six types of dialogue that sometimes involve argumentation) and evaluate a broader range of communication beyond Toulmin's claims, warrants, and so forth (e.g., van Eemeren and Grootendorst 2004, pp. 62–68). Table 5 outlines the different types of analyses that we identified in the literature.

Analysis of core argumentation aspects

Domain-specific patterns Belvedere (Suthers et al. 2001) and *LARGO* (Pinkwart et al. 2006a) use a model of valid argument structures to check *domain-specific constraints* at a *syntactic level*. Typically, these are argument patterns that violate the system's internal model of valid argumentation (e.g., circular arguments, invalidly connected contribution types), or correct but incomplete patterns (e.g., a required contribution type is still missing in the argument diagram). In *Belvedere*, patterns of interest are, for instance, a single data element supporting conflicting hypotheses. *LARGO* uses a model of hypothetical reasoning in the legal domain to identify, for instance, two hypothetical elements linked by a general relation type, where a more specific relation type would be preferable. Patterns are typically specified by expert rules; often a single rule per pattern. Such rule systems can be formalized, for instance, as a graph grammar (Pinkwart et al. 2008a).

In general, such rule-based approaches have several favorable properties: Typically, rules represent local conditions and/or patterns and can be applied more or less independently of one another. The modular nature of rules allows easy modification of existing rules,

Table 5 Overview of discussed analysis approaches

		Description
Argument analyses	Domain-specific patterns	System checks for specific syntactical patterns in diagrams, e.g., circular arguments [*Belvedere*, *LARGO*]
	Problem-specific patterns	System analyzes differences between the students' diagrams and problem-specific expert models [*Belvedere*, *LARGO*, *Rashi*]
	Simulated reasoning	System simulates reasoning processes based on the modeled argument to determine the acceptability of elements in the current constellation. [*Zeno*, *Hermes, ArguMed, Carneades*, *Convince Me*]
	Content quality assessment	System evaluates the quality of individual elements at the content level. [*LARGO*]
	Problem-solving phases	System classifies the current phase of student problem solving when their task is to create an argument diagram. [*Belvedere*, *LARGO*]
Discussion analyses	Discussion process	System analyzes the discussion process and automatically codes contributions (and more complex interaction patterns), for instance, with dialogue acts. [*Epsilon*, *ARGUNAUT*, *Group Leader Tutor*, Rosé et al. (2008)]
	Discussion topics	System identifies specific contents (i.e., topics) discussed. [*Epsilon, Pedabot*, Kumar et al. (2007)]
	Problems while discussing	System detects possible problems in the students' interaction. [*Epsilon*, *ARGUNAUT*, Ravi and Kim (2007)]
	Student and group models	System aggregates and classifies student behavior over a period of time in student and group models. [*Epsilon*, *Group Leader Tutor*]
	Discussion phases	System classifies the current process state and phase according to a hypothesized theoretical model. [*Group Leader Tutor*]

addition of new rules, or deletion of rules. Another advantage is that rules can combine fine-grained heuristics from different pedagogical and domain-specific theories within a single analysis framework. For instance, the collaborative problem-solving environment *COLLECT-UML* (Baghaei et al. 2007) uses the same rule-based analysis framework to identify errors in UML diagrams and problems in the students' collaboration.

Problem-specific patterns Belvedere (Suthers et al. 2001) and *LARGO* (Pinkwart et al. 2006a) provide a second mechanism to identify characteristics of concrete problem instances. The analysis is based on the identification of discrepancies between student and expert diagrams. *LARGO* checks whether irrelevant transcript passages are referenced in an argument diagram, whether important passages have been omitted, and whether an incorrect element type has been associated with a transcript passage (e.g., modeling a transcript passage as "hypothetical" instead of "test"). *Belvedere*'s expert coach searches semantic units in an expert solution that potentially conflicts with the students' solution.

The two systems differ in how expert models are specified, how they are linked to the student solution, and how the actual analysis is carried out. In *LARGO*, experts provide annotations that identify both important and irrelevant transcript passages and classify the role of passages (e.g., "test," "hypothetical"). While constructing an argument diagram, students establish links between diagram elements and corresponding transcript passages. The analysis algorithm then checks for diagram elements that refer to irrelevant passages, missing references to important passages, and classifications of passages that differ from the annotated ones. In *Belvedere*, experts provide a complete solution for each problem by constructing prototypical diagrams. The mapping between elements of student and expert solutions is established by a set of predefined knowledge units that students and experts choose from when constructing their diagrams. The algorithm then uses a constrained search to find knowledge units in the expert solution that are likely to conflict with the students' solution.

Yet another way of matching student and expert solutions can be found in inquiry-learning systems where argumentation is preceded by other activities in which structured data is generated. In the *Rashi Human Biology Inquiry Tutor* (Woolf et al. 2005), students collect data using a set of tools that simulate interviews and examinations of medical patients. The students then use this data to argumentatively decide between alternative diagnoses. Because the simulation tools generate data in structured and machine-interpretable form, the student-created arguments can be compared to expert arguments from a knowledge base.

Problem-specific expert models have the advantage that errors, weaknesses, and/or opportunities to encourage reflection can be detected in the *current problem instance*, a feature that cannot be achieved using the syntactic approaches. On the downside, considerable additional effort is needed to provide expert representations for every problem instance. Some systems, therefore, provide authoring tools to reduce time and effort in developing expert models (e.g., *Rashi*, see Murray et al. 2004). The use of predefined knowledge units, as required in *Belvedere*, restricts the students' task because they do not have to decide the textual content of diagram elements on their own. This contrasts with *LARGO*, where students have to phrase the textual content of diagram elements by themselves. Argumentation can be characterized as an ill-defined domain, that is, an expert model may not even be enough to check student solutions, because these could differ from the expert model, but still be correct. The ill-definedness of argumentation is readily apparent in the legal domain, for instance, where appeal courts frequently overrule prior decisions, that is, higher courts do not always follow the line of argumentation of lower courts. Overall, whether or not to provide problem-specific support is a cost-benefit trade-off, that is, the question is whether learning benefits exceed authoring costs.

Simulated reasoning Whereas the domain- and problem-specific analyses conceive of argument graphs as static representations, some approaches employ argument graphs as "executable" models to *simulate reasoning and decision-making processes*. For instance, *Zeno* (Gordon and Karacapilidis 1997), *Hermes* (Karacapilidis and Papadias 2001), *ArguMed* (Verheij 2003), and *Carneades* (Gordon et al. 2007) use formal-logical models of argumentation (Chesñevar et al. 2000) to determine the acceptability of sentences. With these systems, users model an argument by outlining the argument structure (e.g., proposition elements connected via inferential relationships like "support" and "opposes") and then specifying operational parameters (e.g., indicating whether propositions are accepted/rejected/open at the current stage of a discussion or the inferential strength of relations). The arguments can then be analyzed using formal-logical models of validity that simulate decision procedures, for instance, proof standards from the legal domain such as "preponderance of evidence" and "beyond reasonable doubt" (Karacapilidis and Papadias 2001). The resulting classifications (e.g., a proposition is acceptable or not) can then be displayed to help users draft and generate arguments (Verheij 2003) or make decisions (Karacapilidis and Papadias 2001).

Convince Me (Ranney and Schank 1998) uses a different, but related, approach. It bases its acceptability decisions on the mutual strengthening and weakening of propositions in an undirected network. The network represents a connectionist model of *explanatory coherence* (called *ECHO*) that mimics (theoretical) human reasoning (Theory of Explanatory Coherence (TEC); Thagard 2006). Essentially, TEC models how people evaluate competing explanations. The user input to *Convince Me* is an argument consisting of propositions (evidence and hypotheses) and explanatory and contradictory relations between the propositions. These structures are translated into a neural network consisting of units (propositions), excitatory links (coherence relations), and inhibitory links (incoherence relations). Propositions increase or decrease the activation of their neighbors (as metaphor for believability) in neural net fashion and, after a sufficient number of iterations, the activity values in the network stabilize. The system then displays the model's evaluation together with the students' believability assessments for the same propositions to help students restructure their arguments and/or revise believability ratings. A limitation of quantitative models like *ECHO* is that they cannot easily provide human-understandable explanations, for example, when results appear to be counterintuitive.

The formal-logical and connectionist approaches are similar in that they rely heavily on the argument structure and element properties specified by the user. Therefore, the resulting evaluations depend strongly on how skillfully users model an argument. The simulations are carried out on an argument network that may contain errors and may lack important elements. Hence, they do not provide an external criterion to assess the validity of the arguments or individual propositions per se. Instead, they evaluate the acceptability of elements relative to a *specific network configuration*. Students' argumentation skills can (theoretically) benefit in two ways: First, the simulations might help students understand the mechanics of reasoning by inspecting the acceptability values that emerge from a given argument and by observing how these values change as the argument structure changes. Second, the simulations allow testing the effect of possible counterarguments and determining whether more evidence for one's own position is necessary. Although these approaches provide classifications on the contribution level, they do not assess the actual content of contributions. Their reasoning is based solely on the network structure and user-defined element metadata.

Content quality assessment All of the above approaches rely on well-structured information, amenable to automated processing. However, when students provide natural language contributions to an argument, computational evaluation is much harder and costlier. Shallow text processing is one approach, to be discussed below, but *LARGO* (Pinkwart et al. 2006a) takes a different approach by using peers as an external resource to assess the quality of contributions on the content level. To implement this approach, a sufficient number of peers must work on the same problem, a realistic expectation in most educational scenarios. After finishing a diagram element, students are prompted to provide quality ratings for the diagram elements of their peers that refer to the same transcript passage. These ratings are collected and numerically combined using collaborative filtering (Goldberg et al. 1992).

An asset of collaborative filtering is its relatively low development and online processing cost, especially in contrast to natural language machine learning approaches, which typically require significant data to learn from. Furthermore, prompting students to rate their peers' contributions may have a learning effect because students reflect on and assess their own and their peers' contributions. The feedback of peers can lead to reliable and accurate assessments (Loll and Pinkwart 2009; Cho and Schunn 2007). On the other hand, this is also one of the main limitations of the approach; peers who are willing and capable of providing high quality assessments are a prerequisite. Furthermore, interrupting a student to request such feedback might interfere with the student's own learning activities. The challenge is to find ways to elicit feedback without disturbing or annoying students unduly.

Problem-solving phases Automated analysis may be applied not only to the argument diagrams themselves but also to the *process* and *phases* of creating the diagrams. A simple approach might be to use marked time periods to classify process phases; *LARGO* and *Belvedere* determine the current phase based on dynamic aspects of the momentary solution state, allowing students to work at their own pace. In *LARGO* (Pinkwart et al. 2006b), the analysis of argument transcripts is perceived as a multiphase process involving "orientation," "transcript analysis," "relating elements," "error correction," and "reflection phase." *LARGO* determines the current phase through a *meta-analysis* of domain-specific patterns in the current version of the diagram. *Belvedere* (Suthers et al. 2001) distinguishes between an "early," "middle," and "late" phase. The classification is based on static conditions of each phase, for instance, the diagram is in the "late phase" if there are at least two hypotheses, four data elements, four evidential relations, and the number of data elements and evidential relations exceeds the number of hypotheses.

Knowing the current process phase allows a system to provide more appropriate feedback to students. *LARGO*, for instance, uses feedback messages that encourage reflection on a solution at the later stages of diagram construction, when diagrams have reached a sufficient degree of maturity.

Analysis of more general discussion aspects

Discussion process We next discuss approaches that analyze discussions according to how students interact. The approaches range from classifying single contributions according to their *communicative intention* (e.g., arguing for/against a position, maintaining the dialogue, etc.), to the classification of *adjacency pairs* (e.g., contribution-counterargument pairs, etc.), to the classification of larger *interaction patterns* (e.g., chains of opposition of arbitrary length). The key (potential) advantage of such classifications is that they abstract from the actual textual content and, thus, may make it possible to analyze student interaction.

A number of systems use *sentence-opener interfaces*, which restrict students' communication but also facilitate automated analysis of the communication processes. With such interfaces, students are required to compose messages by choosing from a predefined set of sentence beginnings. *Group Leader Tutor* (McManus and Aiken 1995) was one of the first such systems and uses a handcrafted analysis model that, based on the selected sentence opener, tags each student message with a collaboration skill, a sub-skill, and an attribute. For instance, the sentence opener "The advantages of this idea are …" is tagged with the triple (Creative Conflict, Structuring Controversy, Preparing a Pro Position). Soller et al. (1998) adopted and revised the approach in the context of the collaborative problem-solving system *Epsilon*. A more recent version of the *Group Leader Tutor* (Israel and Aiken 2007) extends the approach by also considering keywords in the students' free text input to resolve possible ambiguities and avoid misclassifications.

On the downside, requiring students to select from a predefined list of sentence openers might tempt them to adjust the content of their contributions to fit an available opener, rather than freely and honestly expressing their view (Soller 2001). Past researchers have tried to "enable the widest possible range of communication with respect to the learning task," (Soller 2001, p. 50) also taking into account differences between face-to-face and computer-mediated interaction. Nevertheless, sentence-opener interfaces still restrict the way that students communicate with one another.

In contrast to sentence-opener approaches, which combine interaction design and knowledge engineering to *manually* build classifiers, supervised machine learning (ML) techniques have the potential to *automatically* derive classifiers from coded data (Witten and Frank 2005). Some approaches employ authentic discussion data coded along several dimensions of interest. The resulting classifiers analyze student contributions in terms of textual, categorical, and contextual properties. Rosé et al. (2008) developed classifiers for message segments in a threaded discussion according to a coding scheme for argumentative knowledge construction (Weinberger and Fischer 2006). Their approach aims at supporting analysts in the task of coding collaborative interactions. They developed successful classifiers to analyze, among others, the "micro-level of argumentation" (e.g., claims, warrants), the "macro-level of argumentation" (e.g., arguments, counterarguments, integrations), and "social modes of co-construction" (e.g., externalization, elicitation, quick consensus building, conflict-oriented consensus building). *ARGUNAUT*'s Deep Loop (McLaren et al. in press) uses a similar approach to support a moderator's awareness in graphical e-Discussions. Six classifiers were successfully learned, including the single contributions off topic and reasoned claim, and the contribution pairs question-answer and contribution-counterargument.

ML has the potential to develop effective classifiers that would, due to their complexity, be hard or even impossible to build in knowledge engineering fashion. On the other hand, a (potential) limitation is that these approaches may not be suitable for directly providing feedback to students. For instance, the *ARGUNAUT* classifiers were tentatively accepted as accurate enough for supporting human moderators, who are well aware of possible misclassification and who have the option of ignoring the classifications. For automated feedback to students, the bar should be set higher because there is no "reject option"; incorrect classifications could lead to inappropriate (and possibly harmful) feedback. Whether the current classifiers are accurate enough remains an open issue. A second issue concerns how one determines the *epistemic unit* of a contribution, which do not always correspond to "natural" and easily detectable message boundaries. The data used in the Rosé et al. (2008) approach were, therefore, manually pre-segmented. Extending the approach to a fully automated online system would require automated segmentation. A third issue, the restrictiveness of the *ARGUNAUT* classifiers to a predetermined structure (single contributions and pairs of contributions), was addressed by another *ARGUNAUT* approach, called *DOCE* (Detection of Clusters by Example; McLaren et al. in press), which is more flexible with regard to the size of patterns (e.g., "chains of opposition" of arbitrary and varying length). Results on this approach are still somewhat preliminary.

Discussion topics Not only *how* students interact with one another but also the *concrete content* of their interaction might be relevant. There are three approaches from the literature that are concerned with the identification of topics in narrow technical domains (thermodynamics, object modeling techniques, operating systems) using knowledge engineering, ML, and Information Retrieval (IR; Baeza-Yates and Ribeiro-Neto 1999) techniques. Although these approaches are not directly targeted at improving argumentation, the identification of the content provides important contextual information that could, potentially, be used to model and support argumentation. For instance, important subtopics that have not been covered yet could be identified to generate context-sensitive feedback. Also, the chronological sequence of topics might be relevant; for instance, we might expect a discussion to progress from basic to advanced topics.

Goodman et al. (2005) developed a topic tracker that was capable of identifying six different topics in student dialogues relevant to the task of object-oriented modeling (e.g., "defining classes"). The topic information is used, among other purposes, to ensure coherent discussion and a sufficient coverage of relevant topics. The topic identification is based on a knowledge engineering approach. Initially, log data was analyzed to determine the accuracy of predictions based on a small set of domain-specific keywords. An error analysis revealed many errors of omission (i.e., students talking about a topic not detected by the classifier) and identified several causes of this, for instance, a too limited vocabulary (e.g., problem-specific terms have not been considered), referential statements (e.g., pronominal references), and misspellings. Based on this analysis, a topic detector was implemented that extends the keyword approach by also considering topic trends and transitions (for instance, referential statements as indicators of the continuation of a prior topic). Kumar et al. (2007) developed an approach to detect thermodynamics concepts in chat discussions. In a first step, a classifier decides whether any of the relevant topics have been raised. This classifier was developed using a ML-based text categorization approach (cf. Sebastiani 2002). In a second step, one of 16 possible topics (e.g., "reheat cycles") is assigned according to the highest *term frequency-inverse document frequency* score (TF-IDF). TF-IDF is an IR measure used here to determine how strongly terms are associated with specific topics. An overall score with respect to a given topic is then computed by averaging over the weights of all terms. The identified topics are used to trigger a tutorial dialogue about the topic. *Pedabot* (Kim et al. 2008) uses similar IR techniques to retrieve messages from past discussions to scaffold current discussions in the context of undergraduate computer science courses on operating systems. When a new discussion thread starts, the system automatically retrieves a list of relevant messages from past semesters.

The identification of topics is different from the more general discourse categories discussed above because *topic-specific* aspects are captured. This leads to classifiers whose scope is necessarily restricted to a limited number of a priori determined topics whereas the process-oriented classifiers capture aspects of human communication that are not tied to a specific topic, leading to a (hopefully) broader scope of applicability. Promising in this respect are approaches such as the one of *Pedabot*. Here, relevant topics and indicative terms for these topics were extracted fully automatically from two text books using a method described by Feng et al. (2006).

Problems during discussion The approaches discussed thus far have been concerned with *how* students communicate, and about what. Some of these characteristics can be interpreted as indicators of interaction quality (e.g., chains of reasoning and conflict-oriented consensus building) or a lack thereof (e.g., off-topic contributions). We now review approaches that focus on the identification of interaction problems, in particular (possibly) problematic patterns in discussions, failed attempts to share knowledge, and lack of responsiveness.

Collaborative argumentation is a social activity, hence, interaction aspects are of key importance. *ARGUNAUT*'s Shallow Loop (Hoppe et al. 2008) allows moderators to search for (possibly) deficient student-student interactions in discussion maps (i.e., graphical representations of e-Discussions) using a "Moderator's Interface" that allows configuring and running differently targeted analysis procedures. Moderators can choose from a number of predefined "shallow alerts" that can, for instance, point to inactivity (e.g., a student has not contributed to the discussion map for x minutes), a lack of interaction (e.g., contributions unconnected for x minutes, users who only link their own contributions with one another), undesired social behavior (e.g., use of profanity, ignored users, indicated by a lack of links to this user's contributions for x minutes), or dominant/overly passive users. The patterns found by *ARGUNAUT*'s Shallow Loop are simple but effective in identifying possible problems in student-student communication. They rely on common sense heuristics and are, thus, easily comprehended by humans, which is an advantage over machine-learned classifiers that are typically based on non-human readable models.

In contrast to the *ARGUNAUT* Shallow Loop (Hoppe et al. 2008), which covers a range of relatively straightforward and easy to detect interaction problems, the approach of Soller (2004) focuses on one specific but harder to analyze problem: exchange of relevant knowledge between the participants in a discussion. Especially when collaborators come from different backgrounds with different expertise, it is essential, for fruitful discussion, that group members share knowledge between themselves (cf. van Eemeren and Grootendorst 2004, p. 60). To address this issue, Soller (2004) developed a computational model to classify episodes of knowledge sharing according to whether they are successful or not. The pieces of shared knowledge correspond to key object modeling (OMT) concepts. The model consists of two Hidden Markov Models (HMM; Rabiner 1989) that were trained with human-annotated sequences of dialogue acts and workspace actions. Soller's approach is also of interest from a methodological point of view because, in contrast to the ML approaches discussed earlier, HMMs capture *sequential dependencies* between observations. Communication processes are by nature sequential, hence, natively sequential modeling approaches like HMMs seem to be a good fit to such modeling problems. This assumption is supported by the results of initial experiments by Soller (2004), which indicate that non-sequential flat representation models are not flexible enough to capture effectively the sequential nature of communication processes. However, her results must be taken with a grain of salt: First, her experiments involved only a very limited number of instances (29), raising the question of reliability and generality of the results. Second, the achieved accuracy of 74% is clearly better than chance, that is, the classifier successfully captures *some* regularities in the students' behavior but is this enough to build a feedback model on? Third, experiments were conducted on *manually pre-segmented* data. Full automation of analysis will require combining an HMM-based classifier with another component to pre-segment the data.

Finally, we discuss an interaction problem that is, in some respects, related to knowledge sharing. Productive discussions require participants to be responsive, that is, to answer questions that have been raised and to acknowledge the statements of their peers. To address this, Goodman et al. (2005) developed classifiers to detect questions and corresponding answers. This information is used to prompt the group to answer questions that have been ignored and is aggregated in student and group models, as will be discussed below. The classifier consisted of two Artificial Neural Networks (ANN), one for question detection, the other for answer detection. After the question detector finds a question, the answer detector tests every contribution that is made within 2 min of the question to check whether there is an answer in reply to the question. The ANNs analyze students' contributions in terms of dialogue acts and surface features (e.g., occurrence of a question mark). Similarly, Ravi and Kim (2007) analyzed threads in a discussion board to classify messages as questions and answers. Two ML classifiers were learned for detecting questions and answers, respectively, which are based solely on text features. The idea was for threads with unanswered questions to be brought to the attention of an instructor.

Student and group models Some systems aggregate student and group behavior in models in a manner similar to student models in intelligent tutoring systems (VanLehn

2006). The collected data might be based directly on student actions (e.g., counting the number of activities) or on results from prior analyses (e.g., counting the number of questions to which the student has replied). These aggregated scores represent the current state of the students' interaction. These *indicators* can be further processed and compared to a model of desired interaction yielding *diagnoses* of student interaction (Soller et al. 2005).

Goodman et al. (2005) developed group and student models to keep statistics on the students' behavior and to determine the general "health" of group collaboration. In particular, the group model contains the number of answered and unanswered questions, showing how (un-)responsive group members are, and values for group agreement and dialogue speed, which are based on action counts and used dialogue acts. Analogously, student models are maintained with values for the student's certainty/confusion and activity level. These indicators are mainly used for the visualization of group health in meters to support student self-reflection but can also be used for immediate feedback, for instance, to prompt the least active student in a collaborating group.

Similarly, the *Group Leader Tutor* (Israel and Aiken 2007) maintains student and group models with counters for off-topic contributions, initiated new ideas, inappropriate use of sentence openers, and the application of the different collaboration skills, all derived from sentence openers/keywords as described above. When certain threshold values are reached, *Group Leader Tutor* intervenes by sending canned chat messages to the participants. For instance, the *Group Leader Tutor* may act when an imbalance of the number of contributions per participant is detected, or when the frequency of off-topic contributions surpasses a certain threshold. *Group Leader Tutor's* low-level counts are further processed to yield higher-level indicators of the group's collaborative effort. For instance, the indicator "initiating ideas and assuming personal responsibility" is computed as the ratio between initiated ideas and the total number of responses of that student. A very low value indicates that the student rarely initiates new ideas (classification: "low"). Values in the midrange show a good balance between initiating and responding (classification: "high"). Other values indicate suboptimal but perhaps not alarming behavior (classification: "medium"). The concrete thresholds are set based on initial trials and research by Robertson et al. (1998). After a session is finished, group and student models are "opened" to the participants to stimulate self-reflection processes.

Discussion phases Some researchers view collaboration and discussion as following regular patterns that can, at least theoretically, be represented and tracked with process models. Knowing the current state of discussion can help a system interpret user information in a situation-specific way or to generate feedback and prompts that are appropriate in a given situation. Not only the current state but also the chronological sequence of state transitions may be of interest and point to possible problems (e.g., in a debate, we expect to see a progression from the detection of a conflict to its resolution).

A part of the pedagogical expertise of the *Group Leader Tutor* (Israel and Aiken 2007) is encoded in a model that tracks the current discussion state. The model is based on Roschelle's process-oriented theory of collaboration through convergence (Roschelle 1992) and represents the discussion as a transition through a state space that can be subdivided into the phases "Display," "Confirm/Disconfirm," and "Repair until Convergence." Technically, the model is realized as a set of finite state machines (FSM); transitions are triggered based on the selected sentence openers. The FSMs might trigger tutorial interventions when dedicated states are entered or discussion loops are detected, which indicate a lack of progress.

Tutorial feedback

Automated argument/discussion analysis is not an end in itself; it typically serves the purpose of providing *feedback*, that is, displaying messages to the user or student to assist them in the task at hand. In this section, we discuss feedback strategies, addressing, in particular, feedback timing, feedback mode, and content and feedback selection strategies.

Feedback control and timing

Feedback control and timing are crucial design decisions that can strongly affect whether learning is successful or not. In particular, in this section, we discuss *who* decides when feedback should be provided (student, system, or moderator) and, *when* the feedback should be provided (immediate or delayed).

On-demand feedback Some systems provide feedback upon a student's request. In *Belvedere* and *LARGO*, for instance, students request feedback to check for possible weaknesses in their argument diagrams and to receive hints on how to proceed. There are several reasons why such a strategy might be beneficial: First, the feedback is provided when the student really wants it, not interrupting ongoing activities. Second, the student is not flooded with unnecessary messages because he or she decides the feedback frequency. Third, the construction of an argument diagram is a continuous process, with no clear conclusion, hence, it makes sense to let the user decide when the process is ready for a check (Pinkwart et al. 2006b). Fourth, on-demand feedback allows the tutoring component to appear less authoritative, possibly leading to less student discouragement (Suthers et al. 2001). On the downside, some students take minimal or even no advantage of on-demand feedback, even when they are stuck, as observed by Suthers et al. (2001) and Pinkwart et al. (2008b).

Immediate system feedback Some systems provide feedback without a student's explicitly requesting it *during* the course of a discussion. The *Group Leader Tutor* (Israel and Aiken 2007) uses this kind of feedback to "repair" communication problems (e.g., when an off-topic contribution has been detected), to mediate phases of creative conflict, to refocus the group when the discussion gets stuck, and to react to changes in the group model (e.g., the relative amount of participation of a student falls below a threshold). Kumar et al. (2007) launch a tutorial dialogue whenever their topic profiler identifies a relevant domain topic in the students' contributions. Goodman et al. (2005) describe the peer agent *Pierce* who provides feedback on unanswered requests/questions, when students appear confused, or when out-of-sequence or missing topics are detected in the *Epsilon* system. Some research has demonstrated the effectiveness of immediate feedback (cf. Shute 2008). Especially when feedback is intended to scaffold and improve the *current* student activity, it should be provided immediately. Second, as mentioned above, many students do not make use of on-demand feedback and, thus, miss learning opportunities. On the other hand, the amount of feedback can become excessive and unnecessarily distract the student (see section "Feedback selection and priority").

Summative system feedback Some systems provide feedback after a session has finished. This kind of feedback is provided by the *Group Leader Tutor* (Israel and Aiken 2007), which displays the content of the student and group model to students at session end in order to encourage reflection and self-assessment of the collaboration. On the one hand, such delayed feedback does not interfere with ongoing students' activities. On the other hand, the feedback does not scaffold student activities in the actual situation in which a problem occurs. In this respect, intermediate and summative feedback can be seen as complementary approaches, one to provide immediate scaffolding and the other to foster reflection when the actual activity is over.

Moderator-driven feedback Some systems let a human moderator decide when to provide feedback to students. For instance, in *ARGUNAUT* (Hoppe et al. 2008), a software tool is provided (the "Moderator's Interface") aimed at increasing the moderator's awareness and helping him/her decide when to intervene. Moderators can easily select alerting rules, at the push of a button, that provide information about, for instance, off-topic contributions and imbalanced participation of group members. If, and when, interventions are triggered is completely under the control of the moderator.

Feedback mode and content

In the following, we discuss the concrete forms of feedback that have been provided in argumentation systems. By this, we mean the mode of feedback (i.e., textual, highlighting of argument elements, visualizations of behavioral/interaction aspects) and specific strategies to phrase the textual content of feedback messages.

Textual The most common form of feedback is textual messages presented to the student. *Belvedere* (Suthers et al. 2001) presents pre-canned text messages, which take the form of suggestions and questions when syntactic patterns are identified, and challenging feedback when differences between the students' diagram and an expert solution have been found. *LARGO* (Pinkwart et al. 2006b) presents short versions of the five most relevant feedback messages to the student. (We will discuss how the most relevant messages are chosen in section "Feedback selection and priority.") Both systems use suggestions/prompts for self-reflection rather than imperative/corrective formulations to avoid confusion when a diagnosis is a "false alarm" (Pinkwart et al. 2006a) and to foster the development of the students' skills of self and peer critiquing, that is, the feedback should encourage the student *himself* to think about the diagram and possible weaknesses (Suthers et al. 2001). For instance, to prompt students to look for evidence against a hypothesis (in order to counteract a possible confirmation bias), *Belvedere* might provide the following feedback: "Don't forget to look for evidence against this hypothesis!" *ARGUNAUT* (Hoppe et al. 2008) allows moderators to provide textual feedback in two ways, first as annotations that are embedded in the argument diagrams and, second, as messages that are displayed in pop-up windows. In the approach of Kumar et al. (2007), a relevant domain concept (or concepts) is automatically identified in the students' conversation. The tutorial agent then takes up the identified concept(s) and tries to stimulate the student to reflect on the concept(s) in a tutorial dialogue. The peer agent *Pierce* (Goodman et al. 2005) appears as a learning companion to students and contributes messages to their chat conversation. *Pierce* can consider multiple indicators when generating a message, for example, he might ask the student with the lowest activity (indicator 1) to comment on a statement that has not been acknowledged/answered (indicator 2). *Pierce*'s interventions are formulated as suggestions and questions (e.g., "Sarah said '…'. What do you think about that, Jeremy?").

Highlighting A second form of feedback is highlighting of relevant portions of an argument diagram. Such an approach is used in *Belvedere* (Suthers et al. 2001) and *LARGO* (Pinkwart et al. 2006a) when the respective systems automatically find syntactic patterns in the students' diagrams. *ARGUNAUT* (Hoppe et al. 2008) allows moderators to highlight contributions in discussion maps to draw students' attention to salient features of the discussion.

Meters Meters are sometimes used to display group indicators (e.g., dialogue speed, relative amount of statements needing a reply, etc.) and student indicators (e.g., certainty level, activity level, etc.). Soller et al. (2005) discusses how meters can be used as *mirroring tools*, which reflect back to students' their actions and behaviors (e.g., student *X* provided *10%* of all contributions), and as *metacognitive tools*, which go beyond this by evaluating students' actions and behaviors and indicate a desired state (e.g., student *X* has a *low* activity level). The peer agent *Pierce* (Goodman et al. 2005) supports students' activity using meters that visualize aspects of the student and group model. The meters are colored green, yellow, or red to indicate whether the current value is in the normal range, in borderline range, or out-of-range. The design of *Pierce's* meters was inspired by research on open student models (e.g., Bull et al. 1995). *ARGUNAUT* also provides meters, but to support a moderator rather than students.

Feedback selection and priority

It is often helpful to control the frequency and selection of feedback, in order to provide the right amount of feedback without flooding students with messages. Goodman et al. (2005) report on tests that simulate interventions of the peer agent *Pierce* with existing usage protocols (recorded without the peer agent). If feedback messages had been sent out, *Pierce* would have interrupted students 75 times simply to react to unacknowledged contributions during a dialogue of 338 utterances, and 38 times to hint on out-of-sequence or missing topics during a 328 utterance dialogue. Pinkwart et al. (2008a) report similar numbers: The number of identified characteristics sometimes exceeds 100 for a single *LARGO* diagram. Clearly, this amount of intervention by both *Pierce* and *LARGO* is likely to overload students.

Belvedere (Suthers et al. 2001) and *LARGO* (Pinkwart et al. 2006a) address this issue by providing the most important and short versions of the five most important feedback messages, respectively, when students request help. Belvedere uses a preference-based quick-sort algorithm (Suthers et al. 2001). The prioritization algorithm iterates through a list of criteria, which are ordered from most to the least important. After applying the first criterion, the second one is used to prioritize feedback that received the same priority value in the first iteration and so on (i.e., consecutive criteria are used as "tie breakers" for preceding ones). Some of *Belvedere*'s criteria are: priority of new advice, priority of expert advice over syntactic advice, priority of advice that binds to diagram elements that have been created by the advice-requesting student, priority of certain pattern types over other types, and so forth. *LARGO* (Pinkwart et al. 2006a) uses, among others, the diagnosed usage phases to determine appropriate feedback. Each pattern is associated with one out of five different usage phases. When the student requests a hint message, those that correspond to the current usage phases are preferred. In *ARGUNAUT* (Hoppe et al. 2008), the control and regulation of feedback is left to a moderator who is assumed to be knowledgeable enough to select the most important feedback, based on the support they are given by the AI-based classifiers (McLaren et al. in press). Goodman et al. (2005) propose to tune the activation threshold of *Pierce*'s intervention rules to reduce the number of interventions. They further suggest the application of more in-depth natural language analyses to improve the accuracy of the indicators and hence, the accuracy of feedback (less "false alarms").

Architecture and technology

Another key aspect of argumentation systems is their underlying software *architecture*. Building upon a solid software architecture is beneficial in reducing development time and in achieving stability, extensibility, and performance. Even more importantly, a suitable software foundation is critical to implementing technologically sophisticated CSCL settings which use collaboration scripts and/or floor control, both of which have been shown to be promising for learning in the context of argumentation (see section "Empirical studies"). However, most argumentation systems are based completely on their own, unique design and code, without even reuse of past successful designs. Our review, which included email contacts with the developers of 12 systems asking them about system architectures, also revealed a huge lack of software documentation. As such, reusable design patterns and guidelines have thus far not emerged from the field of argumentation systems. By way of comparison, in the more general field of software engineering, design patterns have been commonly used in system development for about 15 years (Gamma et al. 1995).

The few publications about software architectures for educational technology systems include early work by Wenger (1987), who proposed an ITS architecture based on four software modules (expert, student, tutor, and communication model). This approach recognizes the significant advantages of clearly separating functionality and designing systems based on a well-specified and modular architecture. However, this early and general proposal, while followed since then in the field of Intelligent Tutoring Systems, is not enough for modern, distributed, and collaborative educational technology systems. We summarize the main contributions of more recent work in the field in the following.

Suthers (2001) discussed different generic argumentation architectures, based on experiences from different versions of *Belvedere*. He differentiated systems based on their coupling model: (1) strict "What you see is what I see" (WYSIWIS), (2) relaxed WYSIWIS, where different users can have different viewports on a shared view, and (3) model-level coupling in which users see the same semantic state of a shared model, but the views may be totally different. Comparing a centralized architecture (used in *Belvedere v1*) and a mixed replicated/distributed architecture (*Belvedere v2*), Suthers finally proposed a hybrid architecture that combines the advantages of the different architectures: The model is available on a central server as well as, in form of a copy, on the client machines. Furthermore, it is possible to have different views on the same underlying data. Therefore, users are able to choose the view that best fits their needs at any time without losing the possibility for collaboration with others that use a different view.

Harrer and Devedzic (Harrer and Devedzic 2002; Devedzic and Harrer 2005) have identified some design patterns for ITS systems, based on detailed reviews of existing systems. Examples are the *KnowledgeModel-View* pattern that manages multiple models and views (analog to the MVC pattern for one model and view) and the *ICSCL* pattern, which allows adapting learning materials separately for individuals and groups at the same time. Even though these patterns are described in the context of intelligent tutoring systems, some (including the two mentioned above) can be applied to designing collaborative argumentation systems (e.g., user-dependent feedback).

While Harrer and Devedzic report on general design patterns, some other publications describe the design of specific systems. Goodman et al. (2005) show how *Pierce*, an agent for supporting collaborative learning, is designed, and Israel and Aiken (2007) present the architecture of their "Intelligent Collaborative Support System." Bouyias et al. (2008) report on ideas for an architecture to support the fading of collaboration scripts. Also, some authors propose component-based architectures for their ITS systems (Kumar et al. 2007;

Israel and Aiken 2007; Tedesco 2003) to facilitate the exchange of modules, argue for specific client-server architectures underlying their implementations (Baghaei et al. 2007; Tedesco 2003; Vizcaino et al. 2000), or describe architectures comprising collaborative learning tools (Belgiorno et al. 2008). Yet, these software architecture descriptions are not adapted to the specific requirements of educational argumentation systems, and it is not clear how they can be used more generally. In Loll et al. (2009), we have presented a preliminary sketch of a flexible architecture specialized for educational argumentation applications—however, this work is currently still in its early stages.

In addition to architecture, another important technological aspect is the *format* used to save and exchange argumentation data in different systems. The choice of the data format is important: A standardized and agreed-upon format, for instance, would facilitate conducting meta-analyses of study data even if the studies were done with different tools. Also, common formats would allow for interoperability between applications, enabling a data exchange (e.g., in order to load data gathered with one tool into another for analysis purposes). In addition, different formats have different affordances. Two primary approaches for argumentation data formats have thus far emerged: state-based (e.g., *GraphXML*: Herman and Marshall 2000; *Graph Exchange Format* (GXL): Taentzer 2001; *Argument Interchange Format* (AIF): Chesñevar et al. 2007) and action-based (e.g., the *Common Format* (CF), which was used in the *ARGUNAUT* project). While the former approach only saves the current state of an argument, the latter stores every action, such as adding, removing, or editing parts of the graphical argument. The action-based approach uses less bandwidth (because only small updates are sent, not the whole map), and is more intuitive for collaborative systems where actions of users must be broadcast to other users. The action-based approach, however, requires more time to compute a map state at a given time, which is required whenever a new client joins an argumentation, because all actions from the beginning to the given time have to be provided to the new client. The choice of format also holds implications on the options for automated argument analysis and feedback: Some analyzers use actions as inputs and would, thus, benefit from an action-based data format (e.g., *ARGUNAUT*), while others are based on the state of an argument (e.g., *LARGO*) and, thus, work better with a state-based data format.

In summary, even though some proposals for data formats have been made, none has been established as a standard for argumentation systems yet. One reason for this may be that they are all limited in different ways: While the mentioned graph-based formats can be used for graph-based representations, they do not work as well for other representations. Furthermore, they do not provide support for argumentation-specific needs (e.g., transcripts or links to external resources). The existing argumentation-specific formats support these needs, but are not flexible enough to support the variety of argument styles used in the different systems.

Empirical studies

In this section, we discuss empirical studies that investigate whether, and under what conditions, argumentation support systems succeed in real use. Many systems use *argument diagrams* (see section "Argument representations") with the implicit assumption that this kind of representation is beneficial for learning. In the first subsection below, we test this assumption by presenting the results of two studies. We then turn to more specific aspects of this assumption. The first two of these are motivated by Bell's (1997) distinction between knowledge representation and discussion tools. In the second subsection, we review studies in which external *knowledge representations* are used to provide discussion guidance for students before turning, in the third subsection, to studies that investigate how the *communication design* of interfaces affect students. Argumentation tools are used in specific *pedagogical contexts*, hence, success is determined not only by the software but also by the overall setting in which it is employed. Empirical studies that explore this issue are discussed in the fourth subsection. Finally, in subsection five, we conclude with a discussion of studies concerned with *intelligent support*. Table 6 provides a brief overview of all of the studies.

Can learning be improved with diagramming tools?

A key question in the design of an argumentation system is how arguments should be presented to the user. As discussed previously, a considerable number of systems use a visual representation; in particular, box-and-arrow diagrams (also called argument graphs). There are intuitive and theoretical reasons for this, for instance, van Gelder (2003) argues

Table 6 Overview: Empirical studies involving argumentation systems

		General description
Can learning be improved with diagrams?	Easterday et al. (2007)	Using diagrams to teach causal reasoning on public policy problems
	Carr (2003)	Using diagrams to teach legal reasoning
The effect of knowledge representations on discourse	Suthers and Hundhausen (2003)	Comparing the effect of different representational notations
	Suthers et al. (2008)	Comparing the effect of different ways to integrate conceptual representations with discourse representations
	Nussbaum et al. (2007)	Using a special-purpose diagramming format (argumentation *vee* diagrams) to support consideration and integration of different viewpoints
The effect of communication design on discourse	Schwarz and Glassner (2007)	Using informal ontologies and floor control in graphical e-Discussions
	McAlister et al. (2004)	Using sentence openers to support critical discussions
	Stegmann et al. (2007)	Using micro-level scaffolding to improve the quality of argumentation
The effect of the overall pedagogical setup	Lund et al. (2007)	Comparing the effect of instructing students to use diagrams for debating versus representing debate
	Munneke et al. (2003)	Comparing the effect of constructing diagrams (a) individually before a debate versus (b) collaboratively during a debate
	Schellens et al. (2007)	Using role assignments to improve argumentation quality
The effect of adaptive support	Pinkwart et al. (2007, 2008b)	Using *LARGO* (and its feedback) to teach hypothetical reasoning
	Schank (1995)	Using *Convince Me* (and its feedback) to improve students' reasoning skills

that diagrams are more *readable and comprehensible* than prose. In particular, they usually require less interpretation; they can be understood more easily via colors, shapes, position in space, and other visual clues; and they are well suited to the non-sequential structure of most arguments. The 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). In a collaborative learning context, graphical representations may be beneficial because they force students to express their ideas to one another in an explicit and complete form, helping to organize and maintain coherence and serving as "conversational resources" (Andriessen 2006). However, structure can also be a burden leading to problems such as "cognitive overhead" and "premature commitment to structure" (Buckingham Shum et al. 1997). When used as a medium for debates, graphical representations can feel unnatural and unintuitive compared to less structured alternatives such as chat. Also, depending on topic, the number of participants, and available discussion time, the number of boxes and arrows can be substantial, leading to cluttered and hard-to-read argument maps. In other words, cognitive, social, and other factors, some of which are supportive, and others of which are detrimental, impact the intended effect of visual representations. This complex interplay of (potential) factors makes it hard to assess a priori whether visualization is appropriate in most situations, or even in a given situation. Thus, it is important that theory-based design be empirically substantiated.

To gain insight into *how* diagrams and diagramming software tools might help learning and facilitate cognitive processes, Easterday et al. (2007) conducted a study in which diagrams were used to teach causal reasoning on public policy problems. The study compared the effects of three interventions on performance and learning: Students in a *Text* condition had to analyze a problem presented as text only; students in a *Diagram* condition were provided with an additional pre-made causal diagram; students in a *Tool* condition were provided with a software diagramming tool they could use to actively construct a diagram from the text. The overall experimental setup included a pretest (showing equivalent groups), training, a performance test, and a learning test. Each of the three tests consisted of a textual argument to be analyzed and ten multiple-choice causal reasoning questions.

The results showed that performance test scores were significantly better for the *Diagram* condition compared to *Text*. Learning test scores were significantly better for the *Tool* condition compared to *Text*. In sum, most effective for scaffolding on-task *performance* was use of the pre-made diagram but for active *learning* constructing a diagram with a software tool was even more effective. The authors posit the following explanation: The performance test is easiest for *Diagram* students because they can directly interpret an external problem representation in the form of a diagram, whereas *Text* and *Tool* students have to perform an additional comprehension step, that is, reading the text, extracting important information, and forming a mental representation. *Text* students have no additional help. They have to solve the problem directly in their minds, which may involve considerable working memory load. *Tool* students spend additional effort constructing a (possibly imperfect) diagram but interpretation of a diagram is easier than doing all the work mentally as in the *Text* condition. In sum, the task leads to similar efforts for *Tool* and *Text* students resulting in similar on-task performance. Pre-made diagrams make it easier to analyze a problem. On the other hand, the learning test scores show that the students that actively constructed diagrams learned most about causal reasoning. Note, however, that this study was restricted to individual problem solving, that is, it does not account for collaborative situations, which are much more complex in nature.

Carr (2003) conducted a study that showed that argument diagramming is *not* necessarily better than traditional learning methods. The study investigated whether second-year law students benefit from using the argument diagramming system *QuestMap* (the experimental group) as a replacement for paper and pencil assignments (the control group). Over the course of a semester, both groups were assigned five identical problems in the legal domain. Their task was to determine whether or not certain types of evidence should be admitted before the court by laying out relevant arguments, counterarguments, rebuttals, and so forth. Students in the treatment group worked in small groups on the problems with access to *QuestMap* while students in the control group worked without *QuestMap*, either alone or in small groups. At the end of the semester, students completed a practice final exam.

The groups were compared in terms of the results of the practice final exam with a task similar to the ones they worked on during the semester. Responses were scored by the professor and coded using the Toulmin model to identify argument components present in the responses. Neither the comparison of overall scores nor the comparison of codes yielded significant differences between the groups. Another analysis was done to determine whether arguments of the *QuestMap* users become more elaborate over time, again without positive results.

This study shows that, contrary to expectation, and the Easterday study, the use of diagramming tools does not *always* lead to measurably better results than traditional (i.e., not visual) instruction. The general question whether diagramming tools, or more generally computer-supported argumentation, can be beneficial for students and superior to other forms of instruction and learning is clearly influenced by a vast number of factors, including students, domain, tool implementation, and overall instructional setup, but also depends on the purpose of the discussion and, ultimately, on how the benefit is measured (there is no universally agreed-upon model or theory on how to measure argumentation quality). It appears that whether and how argumentation tools should be used depends strongly on *context*; that is, a setup that is beneficial in one context may not be so in another context.

The effect of knowledge representations on discourse: Representational guidance and artifact-centered discourse

Knowledge representations can be used to express the *conceptual* structure of a debate while avoiding aspects of human communication (socially oriented talk, interaction management, etc.). By avoiding information that possibly obscures or distracts from the essence of a problem, it may be easier for students to ultimately come up with more informed decisions and solutions. Although a knowledge representation by itself may not be suitable as a medium of communication (Suthers 2003), it might be employed as a *supplemental* resource/tool to structure and guide students in discussion. We discuss three studies that explore this issue: The first study (Suthers and Hundhausen 2003) investigates the effect of *different* knowledge representations in face-to-face discussions. The second (Suthers et al. 2008) investigates how knowledge representations can best be integrated with computer-based communication facilities. The third (Nussbaum et al. 2007) is concerned with argumentation *vee* diagrams, special-purpose representations aimed at encouraging the consideration and integration of different viewpoints.

Suthers and Hundhausen (2003) analyzed the *Matrix*, *Graph*, and *Text* notations in terms of salience of knowledge units and constraints on expressiveness and posited the following four predictions: First, users of *Matrix* and *Graph* notations will classify ideas more often according to conceptual categories because these categories are explicitly represented in these notations (i.e., categories are more salient). Second, *Matrix* users will elaborate the most on evidential relationships because each empty matrix cell prompts them to think about a possible relation (i.e., missing relations are salient); *Graph* users will elaborate the second most because relations are explicitly specified but, on the other hand, contributions lose their salience when the first relation is specified (i.e., they are no longer unconnected). Third, *Matrix* users will also elaborate more on previous ideas because each empty cell in a matrix row/column prompts the students to think about possible relations to previous elements; *Graph* users will elaborate second most, again because of the salience of explicit representations. Fourth, these process differences will, in sum, lead to differences in the learning outcomes and subsequent products because more elaborated information will also be better memorized.

These four hypotheses were tested in a lab study in which a *Matrix*, a *Graph*, and a *Text* group worked on an identical task. Student dyads sitting in front of the same computer had to record data elements, hypotheses, and evidential relations from given background materials in their respective notation using *Belvedere*. Subjects then individually completed a multiple-choice test for domain knowledge and wrote collaborative essays.

The data analysis revealed significant effects of the different notations on process and outcomes, and at least partially confirmed three of the four hypotheses. The use of matrices led to an increased amount of discussion and representation activity with respect to evidential relations, and a more extensive revisiting of previously formulated relations. *Graph* and *Matrix* users visited previously discussed ideas more often than *Text* users. With respect to learning outcomes and subsequent products, an increased impact of students' collaborative learning activity on the written essays could be identified at least for the *Graph* condition in terms of carryover items from sessions to essays. However, none of the two experimental conditions profited in terms of domain knowledge (posttest).

Overall, this study showed that knowledge representation tools influence argumentation, and that the right choice of representational notation increases the chances of favorable outcomes. But how can such conceptual representations be best integrated with verbal discourse to support collaborative knowledge construction? Suthers et al. (2008) explored this question in another study. In this study, verbal discourse took place in an asynchronous fashion via standard CMC technology (threaded discussions and linear chat instead of face-to-face communication); representational artifacts were always of graphical nature (i.e., *evidence maps*). By integrating verbal discourse with visual artifacts, it was hoped to remedy some of the problems that are typically encountered in standard CMC tools, in particular *incoherence* and *lack of convergence*. An evidence map might help to increase coherence because the conceptual relevance of verbal contributions in the discourse becomes more obvious when the participants refer to components of the evidence map; convergence may be improved when verbal contributions that refer to the same topic are collected together within the evidence map.

Three conditions were compared in a lab study: The *Text* group used a standard threaded discussion tool (control condition). The *Graph* group used a graph notation (i.e., nodes and arrows) in which additional chat boxes could be added and linked to any other object. This setup corresponds to a *tight integration* of conceptual representation (nodes and arrows in graph) and discourse representation (chat boxes in graph). The *Mixed* group used a graph notation and a separate threaded discussion tool. In order to contextualize contributions of the threaded discussions, students could insert references to graph elements by clicking on the respective graph object while writing their contribution. This setup corresponds to a *loose integration* of conceptual representation (nodes and arrows of graph) and discourse representation (separate threaded discussion tool with references to graph elements).

Again, this study used the argument mapping system *Belvedere*. Students collaborated in (spatially separated) pairs on a science challenge problem. The experiment was designed to simulate an asynchronous collaboration scenario. Task-relevant information was distributed across students and sessions in order to necessitate the integration of information from different sessions and participants. After the experimental intervention, each student individually wrote an essay. One week later, a posttest was conducted consisting of memory and integration questions, which required the integration of information from different sessions and participants.

The prior results of Suthers and Hundhausen (2003) were essentially confirmed: Users of evidence maps (*Graph* and *Mixed* conditions) stated hypotheses earlier and elaborated more on them compared to the *Text* condition. Furthermore, users in the *Graph* condition also considered more hypotheses than those in the *Text* condition and previously collaborating partners were more likely to converge to the same conclusion. On the other hand, the observed positive process characteristics did not result in better outcomes in terms of essay quality and posttest performance. Concerning the question of whether a tighter or looser integration of knowledge and discourse representation is more effective, slight advantages for the more tightly integrated version were observed: The *Graph* condition performed significantly better on integration questions in the posttest than the *Mixed* condition suggesting that these students were more successful in integrating different pieces of information. Furthermore, some of the significantly positive effects of evidence maps were only observed in the *Graph* condition. A possible explanation is that the distribution of information across two separated tools makes it more difficult for students to integrate the provided knowledge.

These studies suggest that visual representations can foster students' elaboration and integration of information and can guide them to converge to the same conclusion. In a similar vein, Nussbaum et al. (2007) tested whether *argumentation vee diagrams* (AVD; see Fig. 5) can support the integration of different views. AVDs are graphical organizers aimed at scaffolding discussions of controversial questions. They are visually depicted as a "V" with a left column to enter pro arguments, a right column to enter counterarguments, and a text box at the bottom to enter an integrated conclusion. The generation of the conclusion is further supported by critical questions, presented at the base of the "V," to stimulate further reflection. (Note: Newer AVD versions use additional, more specific questions than the ones displayed in Fig. 5.) Overall, AVDs are more direct scaffolds for integration and convergence than the visualizations used in the work of Suthers and colleagues. Their use is motivated by considerations related to the *argument-counterargument integration framework*, a model concerned with psychological and discursive strategies to integrate arguments and counterarguments into a final conclusion. Another aspect of argumentation that AVDs support is the visual presentation of counterarguments, which has the potential to help arguers change their views when appropriate.

The effect of AVDs was tested in a study with undergraduate, pre-service teacher students. Students in the experimental condition used AVDs, students in the control condition did not. In both conditions, small groups discussed over the course of a semester three different questions (e.g., "Should ability grouping be used to teach reading?") using a standard discussion forum. The experimental group also used AVDs: Each student created an individual AVD before the discussion; at the conclusion of their discussion, each group collaboratively created another AVD to summarize their discussion with students assigned specific roles (composer, elaborator, and integrator). After the first discussion round, the instructor reviewed summary notes, constructed AVDs, and provided feedback to the students. Furthermore, additional critical questions were added to the AVDs to provide more guidance.

The results showed that AVD users were significantly more often able to reach a compromise. In addition, their opinions changed more significantly over time. There was

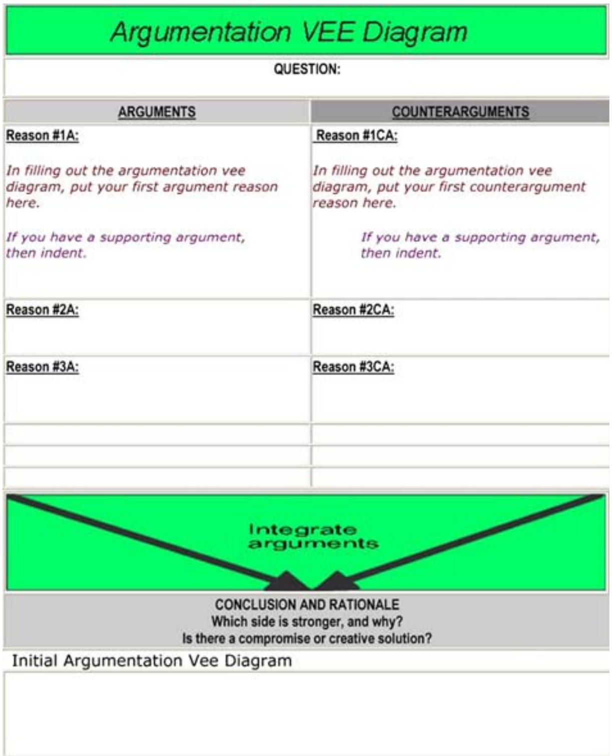

Fig. 5 Argument vee diagram. (Reproduced from Nussbaum et al. (2007), Fig. 1. With kind permission from Michael E. Nussbaum and Springer Science + Business Media)

no opinion change in the control group. These results essentially confirmed the hypothesis that the AVD approach positively contributes to argument-counterargument integration and opinion change.

In summary, these three studies show that students can be guided with external knowledge representations toward more extensive elaboration of information, consideration of counterarguments, and integration of information. However, there are also limitations: The question arises whether the prompted changes in behavior are always favorable. Having *quantitatively* more elaborations, as observed when using matrices, does not necessarily mean that the *quality* of discourse has improved. As noted by Suthers (2003), the prompting character of matrices might have triggered an *overly* extensive consideration of relations, when many of the relations were irrelevant. One must also be careful in attributing the positive effects of the Nussbaum et al. (2007) study exclusively to the use of the AVDs because AVD users also received additional instruction/feedback after the first discussion round. Some representations are only suitable for certain kinds of debates and support only specific aspects of argumentation. For instance, the simplistic form of AVDs seem less appropriate to support more complex debates that involve three or more alternative viewpoints; two-dimensional matrices can only represent relations between two types of objects (e.g., data elements and hypotheses). A final issue is the complexity induced by having multiple tools and notations, for example, one for knowledge representation and another for communication. Students have to learn how to operate each tool, coordinate their usage in a live situation, and cope with distributed information across multiple representations (Suthers et al. 2008).

The effect of communication design on discourse: Predefined categories and turn-taking control

As opposed to guiding collaborating students via supplementary knowledge representations, some approaches try to bias and scaffold students' communication with special-purpose communication interfaces. Three approaches discussed in this section differ in the degree of scaffolding they provide: In the study presented by Schwarz and Glassner (2007), students are biased toward a desired way of interacting in graphical discussions via typed text boxes and links which form an informal ontology for communication. For similar reasons, the approach by McAlister et al. (2004) uses sentence openers in threaded discussions but with the addition that the system recommends appropriate openers for replying to previous messages, following a prescriptive communication model. Stegmann et al. (2007) provide scaffolding through a user interface that supports the creation of single arguments with specific components and argument sequences.

Schwarz and Glassner (2007) investigated the effects of *informal ontologies* and *floor control* in the context of *Digalo*. Informal ontologies are based on reasoning as it typically occurs in conversations in natural settings; floor control describes a technological setup to control turn taking. This study addressed the questions of how floor control and informal ontologies affect communication and the co-construction of knowledge. Floor control allows for an almost synchronous style of communication while also introducing advantages observed in the context of asynchronous CMC. For instance, deeper reflection and less socially oriented talk (which can distract from the learning objectives) were observed in this context, possibly because delays give the user more time to reflect and inhibit impulsive responses. Similar effects can be expected from floor control because users have more time to process their peers' contributions and prepare their own contributions. Schwarz and Glassner (2007) hypothesized that the use of an informal ontology and floor control would lead to more relevant claims and arguments and greater reference to the contributions of peers, while discussions without floor control would contain more chat-style communication, especially when no ontology is used.

The study followed a 2×2 design with the four conditions floor control/ontology (*FO*), floor control/no ontology (*FN*), no floor control/ontology (*NO*) and no floor control/no ontology (*NN*). Participants were seventh graders in an Israeli school that discussed in small groups the topic "whether or not the wearing of school uniforms at school is binding."

The resultant discussion maps yielded results largely in accordance with the hypotheses discussed above, that is, groups using a setup without ontology and floor control (*NN*) produced significantly more chat-like contributions than the other three groups with the largest difference between the two extremes (*FO* versus *NN*). Similarly, the number of "other" references (as opposed to "productive" references) was significantly larger in the most unstructured condition (*NN*) compared to the most structured one (*FO*). Conversely, the number of relevant claims and arguments in the floor control/ontology condition (*FO*) was significantly higher than in the no floor control/no ontology condition (*NN*). In sum, the results confirmed the hypotheses, that is, structuring students' activities via informal ontologies and controlled turn taking positively contributed to favorable behavior (relevant claims and arguments) and suppressed harmful behavior (chat-style communication). In the words of Schwarz and Glassner (2007, p. 474), "the FC [floor control] function gave participants *time* to reflect and react and the ontology function provided a *tool* for reflection and reaction."

McAlister et al. (2004) developed the tool *AcademicTalk*, which is intended to structure academic argumentation in multiple ways: First, the user interface allows organizing discussions in threads, similar to asynchronous CMC but for use in near-synchronous fashion. The rationale is to avoid problems like *sequential incoherence* (see section "Argument representations"), which is common for unstructured alternatives such as chats. A set of sentence openers is provided to scaffold students' communication with a special emphasis on argumentation moves. Finally, a model of well-formed dialogues, based on dialogue game theory, is used to visually highlight preferred replies to bias students toward fruitful exchange. The use of *AcademicTalk* was embedded into a four-phased activity: preparation, exploratory discussion, controversial discussion, and summary. *AcademicTalk* was compared with standard chat in a study with volunteering students of an online delivered University course.

The analysis of computer logs showed clear benefits of *AcademicTalk*: significantly less off-topic contributions and significantly more reasoned claims, rebuttals, and utterances of direct disagreements. As noted by the authors, the more critical character of the *AcademicTalk* discussions might be caused by the sentence openers, which gave the students the "permission" (McAlister et al. 2004, p. 200) to critically react to their peers' contributions. The study design did not allow for teasing out the effect of each of the three interface design principles individually, but there is at least preliminary evidence that the combination of threaded structure, sentence openers, and best recommendation guides students toward a more critical way of discussing.

Stegmann et al. (2007) investigated the effect of a micro-scripting approach on both argumentative and domain knowledge. Their "scaffolding approach to scripting" aspires to have the student internalize the script. They enhanced an asynchronous discussion board with a structured interface to scaffold the construction of single arguments and argument sequences, in order to increase the formal quality of argumentation. *Single arguments* are supported by a set of input text fields implementing a simplified version of the Toulmin model. Students fill in three text fields: "Claim," "Grounds" (data, warrants, and backings in the original Toulmin model), and "Qualifications" (qualifiers and rebuttals in the original Toulmin model). Students are not strictly required to use the three text fields as they can also freely type into a standard input box. The construction of *argumentation sequences* is supported by the system pre-entering the subject lines of new messages. Texts are chosen based on an ideal pattern defined by Leitão (2000) consisting of the argument-counterargument-integration sequence. The first message of a thread is titled "Argument," replies to arguments as "Counterargument," and replies to counterarguments as "Integration." Students were allowed to change the pre-entered subject line.

How might these scaffolds push students toward a more qualitative way of argumentation and learning? A high degree of formal structure in single arguments might be connected with deeper cognitive elaboration: There may be a connection between providing grounds to support a claim and self-explanation, and between providing qualifications (which anticipate exceptional situations and a restricted scope) and the awareness/consideration of alternative explanations and viewpoints. Similarly, argument sequences of the form argument-counterargument-integration may be associated with the creation and solution of socio-cognitive conflicts. Hence, a higher formal quality of argumentation may lead to real learning outcomes in terms of argumentation and domain knowledge.

To investigate the effect of scripting on the formal quality of argumentation and acquisition of knowledge, a study with Educational Science students was conducted, using a 2×2 design with the two factors "single argument scripting" and "argumentation sequence scripting." Small groups of students collaboratively analyzed three problem cases and arrived at a joint solution. The experimental design also involved pre- and posttests on domain and argumentation knowledge.

Log data was analyzed using the method of Weinberger and Fischer (2006) to assess the *formal quality of argumentation*. Single argument scripting significantly increased the formal quality of single arguments indicated by a higher proportion of grounded claims and qualified claims, and less bare claims. Similarly, argumentation sequence scripting significantly increased the formal quality of argumentation sequences indicated by a higher proportion of counterarguments and a higher transition probability from arguments to counterarguments. Corresponding effects have also been observed in the analysis of pre-to-posttest gains in argumentation knowledge. The analysis yielded significantly higher scores for tasks related to single argument and argumentation sequence when a respective script was used during the experimental intervention. However, no significant effects were observed for domain knowledge acquisition.

These three studies show that system designers can bias student interaction by specific communication interfaces. Appropriate interface and interaction designs caused students to use more relevant claims and arguments and less chat-like expressions (Schwarz and Glassner 2007), disagree and rebut more frequently (McAlister et al. 2004), and, overall, engage in argumentation of a higher formal quality (Stegmann et al. 2007). Furthermore, these results were realized with different discourse representations (graphical and threaded discussion) and interaction styles (drag and drop of typed boxes and arrows, sentence openers, and forms). A crucial but still unanswered question is whether these process improvements persist over time, that is, whether these approaches lead to script internalization and a durable change in behavior. A second question is whether these scaffolds should be faded out/removed over time to avoid *over-scripting*, that is, the structural support turns into a hindrance when students have already internalized the model (Dillenbourg 2002; Stegmann et al. 2007). A third question is whether formally improved argumentation is also more effective in terms of acquisition of domain knowledge. Stegmann et al. (2007) found here no significant effects.

The effect of the overall pedagogical setup: Macro-scripting approaches to argumentation

System designers can positively influence how students interact, and possibly what they learn during discussion, through user interfaces. In addition, other contextual factors can contribute to productive computer-mediated exchange. Fruitful collaboration and argumentation typically does not occur spontaneously, that is, it is not sufficient to provide students with basic communication and collaboration tools even if these tools are well designed (Dillenbourg et al. 1996). Based on this insight, researchers and practitioners have tried a number of measures to make argumentation more successful, including the provision of relevant background information, tool familiarization, and procedural instructions. To make argumentation a more situated activity, some have contextualized the use of argumentation tools by designing wider curricula: For instance, *SenseMaker* is part of the *Web-based Inquiry Science Environment* (*WISE)*, for which a number of curriculum projects have been designed (Linn et al. 2003); Suthers et al. (1997) developed activity plans, problems with accompanying Web-based materials, and assessment instruments that could be used, together with *Belvedere*, to implement scientific inquiry in the classroom. Also other more specific setups have been used to foster argumentation. For instance, students possibly collaborate and learn better when they first prepare themselves individually before joining a group discussion (Baker 2003; Schwarz and Glassner 2007; Rummel and Spada 2005), when they receive different background materials to make collaboration necessary for an optimal solution (e.g., Suthers et al. 2008), and when they have been assigned different roles to distribute tasks and emphasize the particular responsibilities of the individual (e.g., Nussbaum et al. 2007; Schellens et al. 2007). Some approaches acknowledge the dialectical character of argumentation, that is, the origin and motivation for argumentation should be a conflict of opinion that provides a reason for argumentation, otherwise argumentation might become aimless (van Eemeren and Grootendorst 2004); learning happens then by resolving this *socio-cognitive conflict* (Doise and Mugny 1984). There are different strategies that can create and/or maximize such conflicts artificially, for instance, by grouping students with different a priori opinions (Baker 2003) or assigning roles that represent different, opposite opinions in a role play scenario, which can be further amplified by preparing students with different background materials according to their roles (Muller Mirza et al. 2007).

These approaches are typically realized as a prescribed series of individual and collaborative activities, known as a *collaboration script*, or more specifically, a *macro-script*. Macro-scripts are pedagogical models, or instructional plans that define phases, roles, and sequences of activities; they stand in contrast to micro-scripts, which are dialogue models intended to be internalized by the student (Dillenbourg and Hong 2008), such as the one by Stegmann et al. (2007) discussed above. In the following, we present three exemplary studies concerning macro-script designs. The first study compares a script in which diagrams are used for *holding* a debate with a script in which diagrams are used for *representing* a debate (Lund et al. 2007). The second study compares a script in which students constructed a diagram *individually before a debate* with one in which a diagram was constructed *collaboratively during a debate* (Munneke et al. 2003). The final study compares a condition in which students were *scripted by roles* with one *without role assignment* (Schellens et al. 2007).

Lund et al. (2007) compared two ways of combining chat and diagrams (i.e., multiple representations; cf. Ainsworth 1999) using different activity sequences. In particular, they compared diagrams used together with a chat as the actual *medium* of debate and using diagrams for representing a preceding chat debate. When using diagrams and chat together, communication is distributed across two different media, and *coordination* of the two representations is required. When using the chat tool as the medium of debate and afterward a diagram to synthesize the debate, a *translation* from chat to diagram representation is required. It was expected that translation involves active reflection processes, which might aid exploring the space of debate and consequently lead to the construction of deeper knowledge.

The study used the CSCL environment *DREW* (Corbel et al. 2002), which incorporates a chat tool, a text editor, and *JigaDREW*, a graphical discussion tool with the typical box-and-arrow notation. An additional feature is that boxes and arrows can be annotated with comments and personal opinions (in favor, against). Elements for which participants express conflicting opinions are visually marked in the diagram to highlight issues that possibly need attention.

The study was conducted with French secondary school children. The discussion topic was "genetically modified organisms" (GMO). In one condition, student dyads were instructed to use chat and diagrams to debate, and to synthesize the debate afterward using chat. In the second condition, student dyads were instructed to use only the chat tool to debate, and to synthesize the debate afterward using the diagramming tool. In other ways the conditions were equal, including the creation of individual diagrams before the experimental intervention and revision of the individual diagrams after the experimental intervention, except for the order of actions.

To compare the two conditions, the collaborative diagrams created during the intervention as well as the changes applied to the individual diagrams after the intervention were analyzed. The analysis yielded two significant effects: First, dyads that used diagrams to represent a chat discussion expressed significantly less often their opinion regarding the *same* diagram element. The collaborative diagram had more of a character of a unique voice of, or consensus between both students. Conversely, dyads that used the diagram for debating more often stated opinions on the same elements, possibly indicating an increased awareness of the distinction between arguments and opinions. Second, dyads that used diagrams for representing added significantly more "non-argumentative" semantic relations to their individual diagrams after the intervention. These links express that one proposition causes, follows from, or provides an example for another proposition, that is, they add explanations and elaborations rather than supporting or opposing arguments. A possible explanation given by the authors is that students might have deepened their understanding of the debate topic, indicated by a tendency to add more semantic relations, which is in line with the assumption that the translation process from a chat to a diagrammatic representation leads to more reflection and the construction of deeper knowledge. Although this last interpretation appears to be a little speculative, the study clearly indicates that different instructions on how to use a notation leads to significant differences, that is, the design of an appropriate context of tool use is a crucial ingredient of successful pedagogy.

Munneke et al. (2003) compared how constructing diagrams individually before a debate and constructing diagrams collaboratively during the debate affect the depth and breadth of students' exploration of the space of debate. The expectation was that preparing an individual diagram before the debate would make the individual viewpoints clearer and consequently trigger opinion statements and reflection on viewpoints. On the other hand, it was hypothesized that constructing a diagram collaboratively during the debate would increase students' elaboration on arguments and relations, that is, students will explore the space of debate in more breadth and depth, in line with the observations by Suthers and colleagues (see section "The effect of knowledge representations on discourse").

A study was conducted with 16 and 17-year old students, in which, like the Lund et al. study, they were asked to discuss GMOs. The main task for both conditions was to collaboratively write an argumentation text. Students used the CSCL tool *TC3* (*Text Composer, Computer Supported, and Collaborative*), which offers a chat, an information panel, a collaborative text editor, and an argument-diagramming tool. In the "Diagram before debate" condition, students started with reading background material and constructed a diagram individually; then they discussed in pairs and wrote the argumentative text while the individual diagrams could still be accessed; finally they were able to modify their individual diagrams once again. In the "Diagram during debate" condition, students started by reading background material individually, then they constructed in pairs a collaborative diagram while the background materials were still available. Finally, student pairs wrote an argumentation text while both diagram and background materials were still available.

The data analysis revealed that students in both conditions were task-focused with only few socially oriented contributions; the diagrams may have helped them maintain the focus in their discussions. Students who prepared an individual diagram before the discussion wrote and talked significantly more about opinions, as predicted. On the other hand, there were no significant differences between conditions in terms of depth of the debate (e.g., relating arguments or arguing on arguments), that is, the collaborative construction of diagrams did *not* lead to more elaboration. Overall, both conditions suffered from a lack of high-value argumentation interaction in the form of rebuttals or relations between arguments, which suggests that additional measures might be necessary to increase interaction quality. Interestingly, the depth of the argumentation texts was in both conditions significantly higher than in the constructed diagrams. It might be the case that students were too focused on the text production task, thus interfering with their discussions in chat and diagrams.

Schellens et al. (2007) approach scripting argumentation with role assignments instead of activity sequences. They investigated how the assignment of roles affects students' argumentation in asynchronous group discussions with a special focus on knowledge construction through social negotiation. Their rationale was that having been assigned a role, a student will take more responsibility toward the group and the content they contribute to the discussion. The research was interested in the effect on both the process and product of collaboration.

A study was conducted with freshman students of an Instructional Design course. During the semester, they discussed four course-relevant topics, which required the application of theoretical concepts to real situations, in groups of ten using an asynchronous discussion board. The course was given in two consecutive years in almost identical fashion, except that in the first year no roles were assigned whereas in the second year the role concept was introduced. Four randomly selected students in each group were assigned one of the roles "moderator," "theoretician," "summarizer," and "source searcher," and were instructed accordingly. The study investigated the impact of the two conditions (role assignment, no role assignment) on exam scores and the level of knowledge construction, which was measured via a content analysis of the students' contributions to the discussion.

The log data analysis showed a significant effect of the specific discussion theme on the average level of knowledge construction possibly due to differences in complexity, that is, discussion themes should be chosen according to the ability level of the students. There were positive effects of student's activity level and attitude toward learning environments (measured via a questionnaire) on both exam score and level of knowledge construction. Students in the "scripted by assigning roles" condition had a significant higher level of knowledge construction and also attained better exam scores. With respect to the effect of specific roles on the level of knowledge construction, summarizers were positively, source searchers negatively affected. No significant effects were found for theoreticians and moderators. In sum, depending on their specific role (and associated tasks), students profit more or less, or might even be negatively affected from role assignments.

These studies show that different scripts lead to different behavior, underscoring the importance of the context in which tools are used. Lund et al. (2007) found some indication that students who summarize a chat discussion in a diagram deepen their conceptual understanding while students using a diagram as a medium are more inclined to express conflicting opinions directly in the diagram. Because the data analysis was based only on the produced diagrams, we cannot know whether the students who used diagrams for representing the debate showed the same ability in their chat discussions; perhaps they just distributed specific activities differently across the different tools. To investigate such questions in more detail, it is necessary to capture and analyze the communication of *all* available channels. Munneke et al. (2003) found that students who construct individual diagrams before a debate talked more about opinions than those who constructed diagrams collaboratively during the debate. Schellens et al. (2007) found overall positive effects of assigning roles to students. In summary, these three examples demonstrate that the pedagogical setup *does* have an influence on process and learning outcome. Another important insight is that there is no general superiority of one script design over others. A particular script might be especially suitable to foster a specific aspect of argumentation or knowledge construction while being neutral or even detrimental with respect to others. Therefore, it is important to understand the impact of certain factors and their interplay in order to design scripts that are appropriate with respect to the pedagogical objectives and the situation at hand.

The effect of adaptive support: Simulated reasoning and tutorial advice

In this section, we discuss evaluation results of educational argumentation systems that *adaptively* support students in their activities. To the best of our knowledge, there are relatively few studies that have been conducted to investigate the benefits of system feedback in argumentation systems. As previously discussed, *AcademicTalk* recommends the next best sentence opener to be used in reply to previous messages. Here we discuss two other studies of adaptive support.

Pinkwart et al. (2007) tested *LARGO* in a lab experiment with paid volunteers recruited from a first-year Legal Process course. An experimental group that used *LARGO* was compared to a control group that used a notepad tool. The study followed a pre-posttest design; the experimental intervention involved students analyzing two legal transcripts over two sessions.

Pretest scores showed no significant differences between both groups. There were also no significant differences in the posttest scores. On the other hand, students at different aptitude levels (*High*, *Medium*, and *Low*) yielded some interesting results: *LARGO* users in the *Low* group did significantly better than the *Low* subjects in the control condition on near-transfer questions and on questions concerned with the evaluation of hypotheticals, a primary learning objective of *LARGO*. Thus, low aptitude students benefited more from *LARGO* than more skilled students. Whether *LARGO's* feedback contributed to the observed benefits cannot be answered definitively because the availability of this function was not part of the experimental design (i.e., there was no condition in which *LARGO* was used without the feedback function). However, a post hoc analysis suggested some positive effects: Students requested feedback quite often and, perhaps more important, the use of feedback increased over time indicating that students found it helpful.

Pinkwart et al. (2008b) performed a second study in which they tested whether the results from the first experiment could be replicated in a more realistic setting. In this study, *LARGO* was used as a mandatory part of a first-year law school course, not by volunteers in a lab as in the first study. Again, the experimental design randomly assigned students to either a *LARGO* group or a text group (the control condition).

However, the prior results were *not* replicated. In fact, some measured significant effects pointed in the opposite direction, that is, the control group performed better on the posttest. To investigate why the effect of *LARGO* was so different between the studies, log data was analyzed. The results showed that students in the second study used the advice function less often (10.1 versus 1.8) with frequency of use decreasing over time, suggesting a possible reason for the reduced benefits of *LARGO* in this experiment. Pinkwart and colleagues argue that the original cause for differences in both studies might, thus, be of motivational nature: Students in study 1 were paid volunteers. Having deliberately decided to take part in the study might indicate some basic interest in the use of such a learning system, which possibly resulted in an increased willingness to explore the system and its features. On the other hand, in study 2, the use of *LARGO* was mandatory but the students did not receive any benefit (performance in the study did not count for course grade) other than learning—which is often not an obvious benefit perceived by students.

Schank (1995) describes a study that compared an experimental group using *Convince Me* with a control group using paper and pencil. The goal was to test whether *Convince Me* is effective in making students better reasoners.

The students were University undergraduates with varying backgrounds, and could be considered novices with respect to the task. The experiment followed a pre-posttest design; during the experimental intervention, four exercises were solved either with *Convince Me* or with paper and pencil. The exercises covered a variety of domains (scientific reasoning in biology, medicine, and physics; an ethical controversy about abortion). Each exercise consisted of instructions and a text passage introducing two competing theories. Students then created a corresponding argument as a set of interrelated hypotheses and pieces of evidence, and rated these elements according to their believability (or acceptability). After having completed an initial version, students in both conditions were able to modify the argument either by changing its structure or their believability judgments.

As a measure of solution quality, the correlation between student-produced argument structures and students' belief ratings was determined. Good reasoners are expected to produce argument structures that reflect their beliefs, or provide belief ratings that correspond to a given argument structure. To determine how well structure and beliefs corresponded, *ECHO* was applied to the arguments to compute how strongly a coherent reasoner believes in the provided propositions, given the particular argument structure. These ratings were then correlated with the actual student beliefs to see how close the student comes to the ideal coherent reasoner. These correlations were computed for the arguments constructed in the pretest, posttest, and during the exercises, and aggregated per condition.

In the pretest, correlations were low for both conditions. Both conditions improved significantly during the exercises, but the *Convince Me* subjects improved to a significantly higher value. These results demonstrate that *Convince Me* was effective in scaffolding students during the task but do positive effects also transfer to situations without tool support? In the posttest, subjects that had used *Convince Me* achieved significantly improved scores compared to the pretest, while the control group had nonsignificant improvements.

Thus, the three studies led to mixed results with respect to adaptive and intelligent support for argumentation. All three studies did not isolate the adaptive support as an experimental factor; rather, each research team compared their system as a whole with a control condition. Hence, it is not clear whether the observed effects stem from the feedback, from the system's user interface, or from some other tool-specific factor. A first *LARGO* study (Pinkwart et al. 2007) shows encouraging results in that low-aptitude students benefit from using *LARGO*; however, their second study (Pinkwart et al. 2008b) did not confirm these results. This result illustrates the importance of contextual factors. Even if one could design the ideal tool, one that provides a maximum of effectiveness and efficiency, it might still be misused and lead to suboptimal results. To overcome the (hypothesized) problem of lacking motivation, the tool could be more tightly integrated into the curriculum, into a collaborative macro-script, or be given more weight with respect to the course grade. Another possibility would be to try to increase students' motivation through *collaborative* use in dyads or small groups. Would different feedback be needed for a collaborative setting? The Schank (1995) study successfully used less direct feedback by showing students how far their beliefs corresponded with a model of coherent reasoning. The use of *Convince Me* was beneficial in terms of both on-task scaffolding and learning. An open question is whether the correspondence between the student's and *ECHO*'s belief ratings accurately measures human reasoning skills. On the one hand, *ECHO* demonstrated that it could provide reliable estimates in a number of different domains and applications. However, *ECHO* is an abstract computational model, with an evaluation mechanism that is difficult for humans to understand. In this respect, evaluations based in human judgment have clear advantages over *ECHO*.

In this section, we have discussed empirical studies investigating the educational effects of argumentation system designs. What is, in a nutshell, the overall result? Do (specifically designed) argumentation systems improve learning compared to a control condition? It is important to distinguish three different aspects: First, do systems successfully scaffold on-task performance (*scaffolding effect*)? This aspect is addressed in almost all studies, either by assessing the quality of argumentative/discourse processes or produced artifacts. Results achieved on this dimension were quite promising, indicating changes in the "right" direction (Easterday et al. 2007; Suthers and Hundhausen 2003; Suthers et al. 2008; Nussbaum et al. 2007; Schwarz and Glassner 2007; McAlister et al. 2004; Stegmann et al. 2007; Schank 1995). Second, do systems, by promoting good argumentation practice, also help students better acquire knowledge about domain topics (*arguing to learn*)? A few studies explicitly addressed this aspect via posttests (Suthers and Hundhausen 2003; Suthers et al. 2008; Stegmann et al. 2007), essentially without significant results. Third, do systems help students acquire argumentation and reasoning skills (*learning to argue*)? Results are mixed here: The studies of Easterday et al. (2007), Stegmann et al. (2007), Pinkwart et al. (2007), and Schank (1995) showed significant positive effects for their systems while Carr (2003) and Pinkwart et al. (2008b) did not. Another important factor is the overall pedagogical setup, that is, *how* argumentation systems are used. The three discussed studies that varied the overall setup showed that not only student on-task behavior changes (Lund et al. 2007; Munneke et al. 2003; Schellens et al. 2007) but also that the acquisition of domain knowledge can be positively influenced (Schellens et al. 2007).

To conclude, it is quite evident that argumentation systems, if well designed, have potential to improve students' argumentative discourse. Improved performance has also been demonstrated in some studies for posttest transfer tasks in which students applied argumentation and reasoning skills without tool support, that is, students seemingly acquired such skills. How durable such changes are and whether acquired argumentation skills impact and transfer into the real-world argumentation practice are still open questions to be answered by future research.

Conclusion

Assisting students in their acquisition of argumentation skills is an important educational goal, one that has clearly been recognized by the research and educational communities. For instance, within the CSCL community, argumentation has been named as one of the critical "flash themes" (Stahl 2007). As this survey article demonstrates, a variety of educational and general-purpose systems for different types of argumentation have been developed over the last (roughly) 15–20 years. There have also been a number of empirical studies that have investigated the pros and cons of different variants of software for learning and practicing argumentation.

Our survey shows that, indeed, some important achievements have been made. When attempting to design an educational argumentation system today, a look at the spectrum of existing systems is both informative and inspiring. As this article shows, the designers of existing systems have chosen different ways to

- represent arguments visually,
- design the interaction between the student, the argument, and (potentially) other students,
- represent arguments in the form of ontologies, and
- automatically analyze arguments and provide students with intelligent feedback.

The variety of existing systems not only illustrates the different design options that are possible, but also, at the same time, shows that the key enabling technologies needed to support students in their learning of argumentation are available today and have been actively developed and tested in a number of projects.

In terms of empirical results, we see at least two general findings that have emerged. A first result is that the form of external argument representation (and accompanying interaction) *does* matter and, thus, should be seriously considered by system designers. The studies by Suthers and Hundhausen (2003), Suthers et al. (2008), Nussbaum et al. (2007), McAlister et al. (2004), Schwarz and Glassner (2007), and Stegmann et al. (2007) show that the way in which a system lays out an argument visually and allows students to use it has an impact on the behavior and learning gains of students. Visual representations that provide more guidance and structure, such as matrices and graphs (Suthers and Hundhausen and Suthers et al. studies) and AVD diagrams (Nussbaum study), have the effect that students make use of that structure, which then leads to more elaborated arguments and argumentative discourse. Micro-scripts, which provide process structure, for example, instructing students how to make an argument and how to respond to them, also have been shown to have the desired effects of students following the underlying rules—such as using more relevant claims and arguments (Schwarz and Glassner study), disagreeing and rebutting other positions more frequently (McAlister study), and engaging in argumentation of a higher formal quality (Stegmann study).

A second general result, supported—among others—by the work of Lund et al. (2007), Munneke et al. (2003), and Schellens et al. (2007), is that scripts on the macro level can be an effective technique for supporting learning with educational argumentation tools. These studies show that the design of a tool is not the only factor that matters when it comes to the question of whether an educational argumentation system is effective or not: The overall pedagogical setup, including sequencing of activities, distributions of roles, instruction on how to use diagramming tools, usage of additional external communication tools, and collaboration design, has an influence on learning outcomes.

In summary, the technology to build educational argumentation systems is available, studies have indicated that argumentation systems can be beneficial for students, and there are research results that can guide system designers and teachers as they implement and use argumentation systems. So has the "CSCL flash theme" mentioned earlier been fully explored? Clearly, the answer is "no." In fact, both on the technology side and on the educational psychology side, there are a number of remaining research challenges that need to be addressed in order to make real progress in understanding how to design, implement, and use educational argumentation software.

First, an overarching theory about how computers can support the acquisition of argumentation skills is largely absent. While some generalizable results (such as the aforementioned findings on visual representations and scripting) exist, our literature review also revealed conflicting empirical results (e.g., about the value of diagrams for educational argumentation) that can currently only be explained speculatively. These inconsistencies reveal the need for a more general theoretical framework about learning argumentation and its support with technology. However, in order to build and refine such a robust theoretical model of learning argumentation, more—and more systematic—empirical studies are required.

The second research challenge is connected to the first. As stated above, the variability of existing educational argumentation systems along multiple dimensions is impressive. Yet, these variations between system designs and approaches have often not been a subject of empirical research, so we do not really know which design option is educationally more beneficial than another. An example is the question about how to design an argument ontology for an educational system. The variety of argumentation ontologies is vast—and surprising—particularly for systems that are ostensibly for "general argumentation." Although many elements are available in a variety of tools, there are no two systems that have the same ontology (though there is probably more overlap in the ontologies than is apparent at the surface: Often, different names are used for the same basic concept, e.g., "supports," "pro," "agree"). Is there a "best" ontology for teaching a specific form of argumentation? That notion clearly merits further empirical research. Even for the restricted research question of whether the size of the ontology is important (too few elements may provide too little guidance, too many elements may confuse students), the literature shows that there are conflicting viewpoints: Suthers (2003) found that too many ontological elements about scientific arguments made student diagrams worse due to student's incorrect usage of the elements, whereas Jeong (2003) and Soller (2001, 2004) reported that their students were able to deal with a wider variety of ontology elements. A simple explanation for the lack of studies that systematically compare different argumentation system designs is that it is quite difficult to practically do such studies—varying factors in a controlled manner would require eliminating confounds, which is quite difficult when two existing software systems are compared as a whole. The systems we have reviewed typically differ in much more than one factor and are often not as flexible in terms of configuration options as is desirable (see also challenge six below).

Third, our review has shown that the vast majority of the existing argumentation tools make use of graph-based argument representations. As stated above, research findings suggest that the way in which arguments are represented visually matters. But are *graphs* really the best way to visually represent arguments in educationally targeted systems? While some theoretical considerations (making structure visible to students) and some convincing technical reasons (making structure accessible for automated analysis) for using graphs exist, there is still no strong empirical evidence about the effect of using graph-based representations for argumentation learning. The lack of empirical support calls for further studies—especially in light of other design choices (such as scripts) having shown some positive effects on student learning. Connected to this third challenge is the finding that all of the approaches that we reviewed have explicitly focused on providing students with visual (i.e., graphical or textual) representations. Yet, factors like gestures, tone of speech, and mimicry are highly important for face-to-face argumentation (Roth 2000; Allwood 2002; Lund 2007a) and, thus, should also be considered for the design of argumentation software. Unfortunately, these aspects of human argumentation do not translate easily into visual representations that a computer system can use.

As a fourth point, a surprising result of our survey is that a considerable number of the tools that we evaluated are indeed for single users, even though collaboration is viewed as critical for learning argumentation. A simple explanation for this is, of course, that collaborative software tools have simply not been around as long as non-collaborative ones. Yet this issue raises research questions about what forms of collaboration would be appropriate with the existing single-user tools, and how collaborative tools for argumentation can and should be designed to support the learning of argumentation. As mentioned, some promising research has already been done in this direction—showing, for instance, that collaboration scripts can be effective for teaching argumentation—but further research on the design of synchronous and asynchronous collaborative argumentation software is clearly required.

Our fifth point is related to the fourth: Our survey has found that the existing educational technology systems for argumentation are associated with two more-or- less completely separated research fields: Intelligent Tutoring Systems (typically single user, with system feedback) on the one hand, and CSCL systems (multiple users, usually with no system feedback) on the other hand. It is surprising to see that these two fields, even though they both address the problem of supporting students as they learn to argue, do not have significant overlapping research or connections (with the notable exceptions of systems like *Belvedere* and *Group Leader Tutor*), as is also evidenced by the absence of articles in *ijCSCL* that address the topic of tutorial feedback. Technically, the design of collaborative ITS systems for argumentation is feasible today. Yet, there are a number of related open research questions that must first be addressed, including how to adapt the support that ITS systems provide to groups of learners, and how to integrate intelligent support on the collaboration process (i.e., adaptive scripting) with intelligent support on a constructed argument (e.g., how to handle conflicting feedback messages that could be presented based on different analyses). Despite these considerable challenges, the integration of ITS and CSCL elements for educational argumentation systems is promising. Argumentation can be characterized as ill-defined (Reitman 1964; Voss 2006; Lynch et al. 2006), and the development of an ITS system that could perform a full and comprehensive automated assessment of student actions and performance in an argumentation task may, thus, be very hard if not impossible. Yet, as we have seen in our review, a partial analysis and helpful feedback on certain aspects of argumentation is often possible with ITS technology. In cases in which this system support is infeasible, collaboration is an obvious choice to facilitate learning for several actors by letting them help one another in their learning processes. Partially, the separation between CSCL and ITS research on argumentation systems can be explained by the different foci of the respective research fields: While ITS research is concerned with domain learning, and providing feedback to help students learn domain content, CSCL research is more interested in interaction and how students collaborate. The two fields also stem from often contentious educational viewpoints (i.e., cognitivist ITS; constructivist CSCL). However, some integration of the positions can probably benefit both fields, because the process of learning group argumentation can hardly be separated from the domain principles of arguments—good arguers need to be strong in both.

The sixth research challenge that our survey revealed is related to the technology side of educational argumentation systems. Our review has revealed (with some notable exceptions as described in section "Architecture and technology") an almost total lack of system documentation and research publications about generic, flexible, and reusable software design patterns for building educational collaborative argumentation systems. Not only do, apparently, few people conduct research on educational argumentation systems from a computer science perspective, but also the existing tools are not well described from a technical viewpoint. This unfortunate situation imposes severe implications on the research community: Researchers who want to design educational argumentation systems often have to reinvent the wheel again and again, expending considerable effort in building new systems. This is because reusable system components and/or source code is generally not available, design rationales are often not well documented, and there is no technology available that offers the degrees of flexibility and configuration options that are required for studying most of the open research issues that we have identified. Such a flexible platform would also have great practical application in schools if teachers were able to configure their systems—guided by research results—in a manner that fits their particular needs.

The seventh and final issue is of a more pragmatic nature. It is the question of how argumentation technology can make inroads into everyday learning settings in schools and universities. To get a better intuition of past and current usages of argumentation systems, we conducted an (informal) email-based survey with the research and development teams of some of the systems listed in "Appendix" (23 requests for 29 systems; 18 responses covering 23 systems). To summarize, at least some of the systems accomplished the transition from research prototypes to systems for real usage, either as free tools or as commercial products. For instance, *SenseMaker* as part of a number of *WISE* projects is used by hundreds of teachers around the world. *Reason!Able* was used in hundreds of schools (mainly in Australia) and dozens of universities. Its commercial successor *Rationale* is now adopted by at least dozens of schools and universities. Regular usages are reported for *Athena*, *iLogos*, and *ArguNet* in higher education philosophy. *InterLoc* has been adopted by tutors in several educational institutions in U.K. The overall breadth of usage is oftentimes not known because many of the systems can be downloaded without registration. For instance, there is quite an active community using the freely available *Compendium* system, and *Araucaria* has been downloaded to over 10,000 different IP addresses, of which a fair amount presumably resulted in educational usages.

However, on a larger scale, the proliferation of argumentation systems in education still lags behind its potential. We want to highlight three directions that hold promise for a wider adoption of argumentation systems in the future: First, Web-based (and Web 2.0) technologies will lower the technical hurdle considerably as compared to most of the existing systems. To use future (and some of today's) argumentation systems, one will only require a Web browser and still have access to an appealing and highly usable multiuser interface. Today's teachers often fail to employ systems in the classroom because of highly complicated installation procedures and/or problems with the IT infrastructure (too restrictive computer/network settings with permissions, open network ports, and firewalls, wrong Java versions installed, etc.). Second, systems should be developed with realistic usage scenarios in mind, also considering the needs and preferences of teachers—without their buy-in, argumentation technology will not succeed in classrooms. A recent example is the *ARGUNAUT* system (Hoppe et al. 2008), which aims at supporting teachers to moderate multiple group discussions in parallel, an important requirement for classroom scenarios. Our third point is connected to the second. The development process of educational argumentation systems should follow a participatory design approach, that is, stakeholders like students, teacher, and educational decision makers should be included from the start and on a regular basis. This inclusion contrasts with many existing approaches, which are mainly driven by theoretical considerations with little or no participation of system users. Having important stakeholders "in the loop," continuously, right from the beginning of a project, will ensure early corrective feedback and tailoring of the software to the users' needs.

Acknowledgment We would like to thank the German Research Foundation (DFG) for providing support for this research under the grant "LASAD - Learning to Argue, Support Across Domains." Also, we want to thank the researchers and developers of the argumentation systems that we reviewed for their feedback in email surveys, and the reviewers for their useful comments.

Appendix: Overview of reviewed tools

Our review covered the systems, methods, and studies shown in the table below. In the rightmost column, in brackets, we provide the number of citations to the main paper of each system, based on a Google Scholar (http://scholar.google.com) as an indicator of the influence of each system. This Google search was done in October and November 2009. All URLs were last visited on 2009-10-27.

No	Tool	Feature description	Reference [#]
1	AcademicTalk	collaborative, educational, sentence openers, based on dialogue game theory	McAlister et al. 2004 [56] http://www.londonmet.ac.uk/ltri/research/projects/at.htm
2	Aquanet	collaborative, configurable ontology	Marshall et al. 1991 [216]
3	Araucaria	transcript, argument schemes, central database for argument exchange	Reed and Rowe 2004 [96] http://Araucaria.computing.dundee.ac.uk
4	Argue/ArguMed	argument assistance, legal domain	Verheij 2003 [66] http://www.ai.rug.nl/~verheij/aaa/argumed3.htm
5	ArguNet	collaborative, Web-based	Schneider et al. 2007 [-] http://www.argunet.org
6	ARGUNAUT	educational, support system for human moderators, used with *Digalo*	De Groot et al. 2007 [12], McLaren et al. in press [-] http://www.argunaut.org
7	Athena	educational, report generator	Rolf and Magnusson 2002 [30] http://www.athenasoft.org
8	AVER	criminal investigations	van den Braak and Vreeswijk 2006 [5]
9	AVERs	criminal investigations	Bex et al. 2007 [13]
10	Belvedere v1 and v2	educational, collaborative, ITS, scientific/evidential reasoning	Suthers et al. 1995 [163], Suthers et al. 2001 [43]
11	Belvedere v3 and v4	educational, collaborative, multiple views, scientific /evidential reasoning	Suthers 2003 [39] http://lilt.ics.hawaii.edu/lilt/software/belvedere
12	BetterBlether	educational, collaborative, sentence openers	Robertson et al. 1998 [67]
13	Carneades	support of multiple proof-standards, IBIS	Gordon et al. 2007 [58] http://carneades.berlios.de
14	CoChemEx	educational, collaborative, inquiry learning, chemistry, scripted	Tsovaltzi et al. 2010 [-]
15	CoFFEE	educational, collaborative, multiple tools, configurable	Belgiorno et al. 2008 [1] http://www.coffee-soft.org/
16	Collaboratorium	collaborative, IBIS	Klein and Iandoli 2008 [1], Malone and Klein 2007 [6] http://cci.mit.edu/research/climate.html
17	Collect-UML	Educational, collaborative, problem solving, UML diagrams, ITS	Baghaei et al. 2007 [6]
18	Compendium	successor of Questmap, collaborative, IBIS	Buckingham Shum et al. 2006 [54], Okada and Buckingham Shum 2008 [1] http://compendium.open.ac.uk
19	Convince Me	educational, model of coherent reasoning	Ranney and Schank 1998 [26] http://www.soe.berkeley.edu/~schank/convinceme
20	CoPe_it!	successor of Hermes, (also) educational, collaborative, multiple views, support of multiple proof-standards, decision support, IBIS	Karacapilidis 2009 [-]
21	CycleTalk Chat Environment	educational, collaborative, problem solving, thermodynamics, tutorial dialogues	Kumar et al. 2007 [22]

No	Tool	Feature description	Reference [#]
22	DebateGraph	collaborative, local views	http://www.debategraph.org
23	Debatepedia	collaborative, wiki-based	http://wiki.idebate.org
24	Digalo	educational, collaborative, configurable ontology	Schwarz and Glassner 2007 [4] http://www.dunes.gr
25	DREW	educational, collaborative, multiple tools	Corbel et al. 2002 [18]
26	Epsilon (with tutorial agent Pierce)	educational, collaborative, problem solving, OMT diagrams, sentence openers, interaction analysis, tutorial feedback, group and student model	Goodman et al. 2005 [28]
27	Epsilon (interaction analysis)	educational, collaborative, problem solving, OMT diagrams, sentence openers, interaction analysis	Soller 2001 [214]; Soller 2004 [38]
28	Group Leader Tutor	educational, collaborative, sentence openers, Group Leader agent to facilitate interaction	McManus and Aiken 1995 [100]; Israel and Aiken 2007 [3]
29	Hermes	collaborative, support of multiple proof-standards, decision support, IBIS	Karacapilidis and Papadias 2001 [128] http://www-sop.inria.fr/aid/hermes
30	IBIS/gIBIS	collaborative, notational support to solve wicked problems	Conklin and Begeman 1988 [1310]
31	iLogos	educational, causal diagrams	Easterday et al. 2007 [4] http://www.phil.cmu.edu/projects/argument_mapping
32	Interloc	successor of *AcademicTalk*, educational, collaborative, sentence openers, configurable dialogue games	Ravenscroft et al. 2008 [1] http://www.interloc.org
33	KIE/SenseMaker, WISE	educational; container visualization, inquiry learning, science learning	Bell 1997 [142]; Bell and Linn 2000 [222], Linn et al. 2003 [89] http://tels.sourceforge.net/sensemaker
34	LARGO	educational; legal argumentation, ITS	Pinkwart et al. 2006a [23]
35	LASAD	educational, collaborative, flexible/configurable architecture, intelligent support	Loll et al. 2009 [-] http://cscwlab.in.tu-clausthal.de/lasad/
36	Legalese	legal argumentation	Hair 1991 [10]
37	Pedabot	educational, support for technical discussion boards by IR	Kim et al. 2008 [4]
38	Questmap	collaborative, IBIS	Carr 2003 [32]
39	Rashi/Human Biology Inquiry Tutor	educational, ITS, inquiry learning, multiple tools	Woolf et al. 2005 [4]
40	Rationale	educational, multiple argument modes	Van Gelder 2007 [8] http://rationale.austhink.com
41	Reason!Able	educational	Van Gelder 2002 [22], Van Gelder 2003 [35]
42	Room 5	collaborative, legal argumentation, implements dialogue game	Loui et al. 1997 [54]
43	SEAS	decision support, argument templates; table, starburst and constellation depictions of multidimensional arguments	Lowrance 2007 [4], Lowrance et al. 2008 [4] http://www.ai.sri.com/~seas

No	Tool	Feature description	Reference [#]
44	TC3	educational, collaborative, tool suite to support collaborative writing of argumentative texts	Munneke et al. 2003 [14]
45	Zeno	Predecessor of Carneades and Hermes, support of multiple proof-standards, decision support, IBIS	Gordon and Karacapilidis 1997 [213]
46	–	educational, collaborative	Jeong 2003 [115]
47	–	educational, collaborative, argumentation vee diagrams	Nussbaum et al. 2007 [2]
48	–	educational, collaborative, scripting by roles approach	Schellens et al. 2007 [3]
49	–	educational, collaborative, micro-scripting, Toulmin-based	Stegmann et al. 2007 [16]
50	–	educational, collaborative, integration of conceptual and discourse representations, uses Belvedere	Suthers et al. 2008 [37]

References

Ainsworth, S. E. (1999). Designing effective multi-representational learning environments, *Technical Report number 58. ESRC centre for research in development, instruction and training*, University of Nottingham, UK.

Aleven, V., & Ashley, K. D. (1997). Teaching case-based argumentation through a model and examples: Empirical evaluation of an intelligent learning environment. In B. du Boulay & R. Mizoguchi (Eds.), *Proceedings of the 8th International Conference on Artificial Intelligence in Education (AI-ED 1997)* (pp. 87–94). Amsterdam: IOS.

Allwood, J. (2002). Bodily communication dimensions of expression and content. In B. Granström, D. House, & I. Karlsson (Eds.), *Multimodality in language and speech systems* (pp. 7–26). Dordrecht: Kluwer Academic.

Andriessen, J. (2006). Arguing to learn. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 443–460). New York: Cambridge University Press.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison Wesley.

Baghaei, N., Mitrovic, A., & Irwin, W. (2007). Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 2*(2–3), 159–190.

Baker, M. J. (2003). Computer-mediated argumentative interactions for the co-elaboration of scientific notions. In J. Andriessen, M. J. Baker, & D. D. Suthers (Eds.), *Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments* (pp. 47–78). Dordrecht: Kluwer Academic.

Belgiorno, F., De Chiara, R., Manno, I., Overdijk, M., Scarano, V., & van Diggelen, W. (2008). Face to face cooperation with CoFFEE. In P. Dillenbourg & M. Specht (Eds.), *Proceedings of the 3rd European Conference on Technology Enhanced Learning (EC-TEL 2008)* (pp. 49–57). Berlin: Springer.

Bell, P. (1997). Using argument representations to make thinking visible for individuals and groups. In R. Hall, N. Miyake, & N. Enyedy (Eds.), *Proceedings of the 2nd International Conference on Computer Support for Collaborative Learning (CSCL 1997)* (pp. 10–19). Toronto: University of Toronto Press.

Bell, P., & Linn, M. C. (2000). Scientific arguments as learning artifacts: Designing for learning from the web with KIE. *International Journal of Science Education, 22*, 797–817.

Bex, F., Prakken, H., Reed, C., & Walton, D. (2003). Towards a formal account of reasoning about evidence: Argumentation schemes and generalizations. *Artificial Intelligence and Law, 11*(2–3), 125–165.

Bex, F., van den Braak, S., Oostendorp, H., Prakken, H., Verheij, B., & Vreeswijk, G. A. W. (2007). Sense-making software for crime investigation: How to combine stories and arguments? *Law, Probability and Risk, 6*(1–4), 145–168.

Bouyias, Y. N., Demetriadis, S. N., & Tsoukalas, I. A. (2008). iArgue: A web-based argumentation system supporting collaboration scripts with adaptable fading. In *Proceedings of the 8th*

International Conference on Advanced Learning Technologies (ICALT 2008) (pp. 477–479). IEEE Computer Society Press.
Bransford, J. D., Brown, A. L., & Cocking, R. R. (1999). *How people learn: Brain, mind, experience, and school*. Washington: National Academy Press.
Buckingham Shum, S. J., MacLean, A., Bellotti, V. M. E., & Hammond, N. V. (1997). Graphical argumentation and design cognition. *Human-Computer Interaction, 12*(3), 267–300.
Buckingham Shum, S. J., Uren, V., Li, G., Domingue, J., Motta, E., & Mancini, C. (2002). Designing representational coherence into an infrastructure for collective sense-making. *Invited discussion paper presented at the 2nd International Workshop on Infrastructures for Distributed Collective Practices*.
Buckingham Shum, S. J., Selvin, A. M., Sierhuis, M., Conklin, J., Haley, C. B., & Nuseibeh, B. (2006). Hypermedia support for argumentation-based rationale: 15 years on from gIBIS and QOC. In A. H. Dutoit, R. McCall, I. Mistrik, & B. Paech (Eds.), *Rationale management in software engineering* (pp. 111–132). Berlin: Springer.
Bull, S., Brna, P., & Pain, H. (1995). Extending the scope of the student model. *User Modeling and User-Adapted Interaction, 5*(1), 45–65.
Carr, C. S. (2003). Using computer supported argument visualization to teach legal argumentation. In P. A. Kirschner, S. J. Buckingham Shum, & C. S. Carr (Eds.), *Visualizing argumentation: Software tools for collaborative and educational sense-making* (pp. 75–96). London: Springer.
Chesñevar, C., Maguitman, A., & Loui, R. (2000). Logical models of argument. *ACM Computing Surveys, 32*(4), 337–383.
Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., et al. (2007). Towards an argument interchange format. *The Knowledge Engineering Review, 21*(4), 293–316.
Cho, K., & Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education, 48*(3), 409–426.
Conklin, J., & Begeman, M. L. (1988). gIBIS: A hypertext tool for exploratory policy discussion. *Proceedings of the ACM Conference on Computer-supported Cooperative Work (CSCW '88)* (pp. 140–152). New York: ACM.
Corbel, A., Girardot, J. J., & Jaillon, P. (2002). DREW: A dialogical reasoning web tool. In A. Méndez-Vilas & J. A. Mesa González (Eds.), *Information Society and Education: Monitoring a Revolution, Proceedings of International Conference on Information and Communication Technologies in Education (ICTE 2002)* (pp. 516–521).
de Groot, R., Drachman, R., Hever, R., Schwarz, B., Hoppe, U., Harrer, A., et al. (2007). Computer supported moderation of e-discussions: The ARGUNAUT approach. In C. Chinn, G. Erkens & S. Puntambekar (Eds.), *Mice, minds, and society—The computer supported collaborative learning (CSCL) Conference 2007*, (pp. 165–167). International Society of the Learning Sciences.
Devedzic, V., & Harrer, A. (2005). Software patterns in ITS architectures. *International Journal of Artificial Intelligence in Education, 15*(2), 63–94.
de Vries, E., Lund, K., & Baker, M. (2002). Computer-mediated epistemic dialogue: Explanation and argumentation as *vehicles for understanding scientific notions*. *The Journal of the Learning Sciences, 11*(1), 63–103.
Dillenbourg, P. (2002). Over-scripting CSCL. In P. A. Kirschner (Ed.), *Three worlds of CSCL: Can we support CSCL* (pp. 61–91). Heerlen: Open University of the Netherlands.
Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macro scripts. *International Journal of Computer Supported Collaborative Learning (ijCSCL), 3*(1), 5–23.
Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1996). The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds.), *Learning in humans and machine: Towards an interdisciplinary learning science* (pp. 189–211). Oxford: Elsevier.
Doise, W., & Mugny, W. (1984). *The social development of the intellect*. Oxford: Pergamon.
Easterday, M. W., Aleven, V., & Scheines, R. (2007). 'Tis better to construct or to receive? Effect of diagrams on analysis of social policy. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AI-ED 2007)* (pp. 93–100). Amsterdam: IOS.
Feng, D., Kim, J., Shaw, E., & Hovy, E. (2006). Towards modeling threaded discussions through ontology-based analysis. In *Proceedings of 21st National Conference on Artificial Intelligence (AAAI 2006)* (pp. 1289–1294).
Gamma, E., Helm, R., Johnson, R. E., & Vlissides, J. (1995). *Design patterns. Elements of reusable object-oriented software*. Amsterdam: Addison-Wesley Longman.
Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61–70.

Goodman, B. A., Linton, F. N., Gaimari, R. D., Hitzeman, J. M., Ross, H. J., & Zarrella, G. (2005). Using dialogue features to predict trouble during collaborative learning. *User Modeling and User-Adapted Interaction, 16*(1–2), 85–134.

Gordon, T. F., & Karacapilidis, N. (1997). The Zeno argumentation framework. In *Proceedings of the 6th International Conference on AI and Law (ICAIL 1997)* (pp. 10–18), New York: ACM.

Gordon, T. F., Prakken, H., & Walton, D. (2007). The Carneades model of argument and burden of proof. *Artificial Intelligence, 171*(10–15), 875–896.

Gruber, T. R. (1993). A translation approach to portable ontology specification. *Knowledge Acquisition, 5*(2), 199–220.

Hair, D. C. (1991). Legalese: A legal argumentation tool. *SIGCHI Bulletin, 23*(1), 71–74.

Harrer, A., & Devedzic, V. (2002). Design and analysis patterns in its architectures. In *Proceedings of the International Conference on Computers in Education (ICCE'02)* (pp. 523–527). Washington: IEEE Computer Society Press.

Herman, I., & Marshall, M. S. (2000). *GraphXML—an xml based graph interchange format*. Available online: http://www.cwi.nl/ftp/CWIreports/INS/INS-R0009.pdf, last visited: 2009-10-29.

Hoppe, H. U., Giemza, A., Wichmann, A., Krauß, M., Baurens, B., Rigolleau, B., et al. (2008). *Combined deliverable D3.2b—moderator's interface and D4.2b—The off-line tracer. Part A (documentation)*. Argunaut project deliverable. Available online: http://www.argunaut.org/argunaut-d3-2b-d4-2b-PartA.pdf, last visited: 2009-10-29.

Israel, J., & Aiken, R. (2007). Supporting collaborative learning with an intelligent web-based system. *International Journal of Artificial Intelligence in Education, 17*, 3–40.

Jeong, A. C. (2003). The sequential analysis of group interaction and critical thinking in online threaded discussions. *The American Journal of Distance Education, 17*(1), 25–43.

Karacapilidis, N., & Papadias, D. (2001). Computer supported argumentation and collaborative decision making: The Hermes system. *Information Systems, 26*(4), 259–277.

Karacapilidis, N., Tzagarakis, M., Karousos, N., Gkotsis, G., Kallistros, V., Christodoulou, S., et al. (2009). Tackling cognitively-complex collaboration with CoPe_it! *International Journal of Web-Based Learning and Teaching Technologies, 4*(3), 22–38.

Keefer, M. W., Zeitz, C. M., & Resnick, L. B. (2000). Judging the quality of peer-led student dialogues. *Cognition and Instruction, 18*(1), 53–81.

Kim, J., Shaw, E., Ravi, S., Tavano, E., Arromratana, A., & Sarda, P. (2008). Scaffolding on-line discussions with past discussions: An analysis and pilot study of pedabot. In B. P. Woolf, E. Aimeur, R. Nkambou, & S. Lajoie (Eds.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Conference (ITS-08)* (pp. 343–352). Berlin: Springer.

Klein, M., & Iandoli, L. (2008). Supporting collaborative deliberation using a large-scale argumentation system: The MIT collaboratorium, MIT Sloan Research Paper No. 4691-08. Available online: http://ssrn.com/abstract=1099082, last visited: 2009-11-26.

Koschmann, T. (2003). CSCL, argumentation, and Deweyian inquiry: Argumentation is learning. In J. Andriessen, M. J. Baker, & D. D. Suthers (Eds.), *Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments* (pp. 261–269). Dordrecht: Kluwer Academic.

Kuhn, D. (1991). *The skills of argument*. Cambridge: Cambridge University Press.

Kumar, R., Rosé, C. P., Wang, Y.-C., Joshi, M., & Robinson, A. (2007). Tutorial dialogue as adaptive collaborative learning support. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AI-ED 2007)* (pp. 383–390). Amsterdam: IOS.

Kunz, W., & Rittel, H. (1970). *Issues as elements of information systems. Working paper #131*. Institut für Grundlagen der Planung I.A. University of Stuttgart, Germany. Available online: http://www.cc.gatech.edu/~ellendo/rittel/rittel-issues.pdf, last visited: 2009-10-29.

Leitão, S. (2000). The potential of argument in knowledge building. *Human Development, 43*(6), 332–360.

Linn, M. C., Bell, P., & Hsi, S. (1998). Using the Internet to enhance student understanding of science: The knowledge integration environment. *Interactive Learning Environments, 6*(1–2), 4–38.

Linn, M. C., Clark, D., & Slotta, J. D. (2003). Wise design for knowledge integration. *Science Education, 87*(4), 517–538.

Loll, F., & Pinkwart, N. (2009). Using collaborative filtering algorithms as eLearning tools. In R. H. Sprague (Ed.), *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS 2009)*. IEEE Computer Society Press.

Loll, F., Pinkwart, N., Scheuer, O., & McLaren, B. M. (2009). Towards a flexible intelligent tutoring system for argumentation. In I. Adeo, N. Chen, Kinshuk, D. Sampson, & L. Zaitseva (Eds.), *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies* (pp. 647–648). Los Alamitos: IEEE Computer Society.

Loui, R. P., Norman, J., Altepeter, J., Pinkard, D., Linsday, J., & Foltz, M. (1997). Progress on room 5: A testbed for public interactive semi-formal legal argumentation. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL 1997*) (pp. 207–214). ACM.

Lowrance, J. D. (2007). Graphical manipulation of evidence in structured arguments. *Law, Probability & Risk, 6*(1–4), 225–240.

Lowrance, J., Harrison, I., Rodriguez, A., Yeh, E., Boyce, T., Murdock, J., et al. (2008). Template-based structured argumentation. In A. Okada, S. Buckingham Shum, & T. Sherborne (Eds.), *Knowledge cartography: Software tools and mapping techniques* (pp. 307–333). London: Springer.

Lund, K. (2007a). The importance of gaze and gesture in interactive multimodal explanation. *Language Resources and Evaluation, 41*(3–4), 289–303.

Lund, K., Molinari, G., Séjorné, A., & Baker, M. (2007). How do argumentation diagrams compare when student pairs use them as a means for debate or as a tool for representing debate? *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 2*(2–3), 273–295.

Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). Defining ill-defined domains; a literature survey. In V. Aleven, K. Ashley, C. Lynch, & N. Pinkwart (Eds.), *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems (ITS 2008)* (pp. 1–10). Jhongli: National Central University.

Malone, T. W., & Klein, M. (2007). Harnessing collective intelligence to address global climate change. *Innovations, 2*(3), 15–26.

Marshall, C. C., Halasz, F. G., Rogers, R. A., & Janssen, W. C. (1991). Aquanet: A hypertext tool to hold your knowledge in place. In *Proceedings of the 3rd annual ACM conference on Hypertext (HYPERTEXT '91)* (pp. 261–275). New York: ACM.

McAlister, S., Ravenscroft, A., & Scanlon, E. (2004). Combining interaction and context design to support collaborative argumentation using a tool for synchronous CMC. *Journal of Computer Assisted Learning: Special Issue: Developing Dialogue for Learning, 20*(3), 194–204.

McLaren, B. M., Scheuer, O., & Mikšátko, J. (in press). Supporting collaborative learning and e-discussions using artificial intelligence techniques. *International Journal of Artificial Intelligence in Education.*

McManus, M., & Aiken, R. (1995). Monitoring computer-based problem solving. *Journal of Artificial Intelligence in Education, 6*(4), 307–336.

Muller Mirza, N., & Perret-Clermont, A.-N. (2009). *Argumentation and education: Theoretical foundations and practices* (1st ed.). Berlin: Springer.

Muller Mirza, N., Tartas, V., Perret-Clermont, A.-N., & de Pietro, J.-F. (2007). Using graphical tools in a phased activity for enhancing dialogical skills: An example with digalo. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 2*(2–3), 247–272.

Munneke, L., van Amelsvoort, M., & Andriessen, J. (2003). The role of diagrams in collaborative argumentation-based learning. *International Journal of Educational Research, 39*(1–2), 113–131.

Murray, T., Woolf, B., & Marshall, D. (2004). Lessons learned from authoring for inquiry learning: A tale of authoring tool evolution. In J. C. Lester, R. M. Vicari, & F. Paraguaçu (Eds.), *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS 2004)* (pp. 197–206). Berlin: Springer.

Nussbaum, E. M., Winsor, D. L., Aqui, Y. M., & Poliquin, A. M. (2007). Putting the pieces together: Online argumentation vee diagrams enhance thinking during discussions. *International Journal on Computer-Supported Collaborative Learning (ijCSCL), 2*(4), 479–500.

Okada, A., & Buckingham Shum, S. (2008). Evidence-based dialogue maps as a research tool to investigate the quality of school pupils' scientific argumentation. *International Journal of Research & Method in Education, 31*(3), 291–315.

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006a). Toward legal argument instruction with graph grammars and collaborative filtering techniques. In M. Ikeda, K. Ashley, & T. W. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)* (pp. 227–236). Berlin: Springer.

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006b). Schwachstellenermittlung und Rückmeldungsprinzipen in einem intelligenten Tutorensystem für juristische Argumentation. In M. Mühlhäuser, G. Rößling & R. Steinmetz (Eds.), *GI lecture notes in informatics—Tagungsband der 4. e-Learning Fachtagung Informatik*.

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2007). Evaluating legal argument instruction with graphical representations using largo. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AI-ED 2007)* (pp. 101–108). Amsterdam: IOS.

Pinkwart, N., Ashley, K., Lynch, C., & Aleven, V. (2008a). Graph grammars: An ITS technology for diagram representations. In D. Wilson & H. C. Lane (Eds.), *Proceedings of 21st International FLAIRS Conference* (*FLAIRS-21)* (pp. 433–438). Coconut Grove: AAAI.

Pinkwart, N., Lynch, C., Ashley, K., & Aleven, V. (2008b). Re-evaluating Largo in the classroom: Are diagrams better than text for teaching argumentation skills? In B. P. Woolf, E. Aimeur, R. Nkambou, & S. Lajoie (Eds.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS 2008)* (pp. 90–100). Berlin: Springer.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257–286.

Ranney, M., & Schank, P. (1998). Toward an integration of the social and the scientific: Observing, modeling, and promoting the explanatory coherence of reasoning. In S. Read & L. Miller (Eds.), *Connectionist models of social reasoning and social behavior* (pp. 245–274). Mahwah: Erlbaum.

Ravenscroft, A., Sagar, M., Baur, E., & Oriogun, P. (2008). Ambient pedagogies, meaningful learning and social software. In S. Hatzipanagos & S. Warburton (Eds.), *Social software & developing community ontologies* (pp. 432–450). Hershey: IGI Global.

Ravi, S., & Kim, J. (2007). Profiling student interactions in threaded discussions with speech act classifiers. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AI-ED 2007)* (pp. 357–364). Amsterdam: IOS.

Reed, C., & Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools, 14*(3–4), 961–980.

Reitman, W. R. (1964). Heuristic decision procedures open constraints and the structure of ill-defined problems. In M. W. Shelly & G. L. Bryan (Eds.), *Human judgment and optimality* (pp. 282–315). New York: Wiley.

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

Robertson, J., Good, J., & Pain, H. (1998). BetterBlether: The design and evaluation of a discussion tool for education. *International Journal of Artificial Intelligence in Education, 9*, 219–236.

Rolf, B., & Magnusson, C. (2002). Developing the art of argumentation. A software approach. In Proceedings of *the 5th International Conference on Argumentation. International Society for the Study of Argumentation (ISSA-2002)* (pp. 919–926).

Roschelle, J. (1992). Learning by collaborating: Convergent conceptual change. *Journal of the Learning Sciences, 2*(3), 235–276.

Rosé, C., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., et al. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 3*(3), 237–271.

Roth, W.-M. (2000). From gesture to scientific language. *Journal of Pragmatics, 32*(11), 1683–1714.

Rummel, N., & Spada, H. (2005). Learning to collaborate: An instructional approach to promoting collaborative problem-solving in computer-mediated settings. *Journal of the Learning Sciences, 14*(2), 201–241.

Schank, P. (1995). *Computational tools for modeling and aiding reasoning: Assessing and applying the theory of explanatory coherence*. Doctoral dissertation. University of California, Berkeley. (University Microfilms No. 9621352).

Schellens, T., van Keer, H., de Wever, B., & Valcke, M. (2007). Scripting by assigning roles: Does it improve knowledge construction in asynchronous discussion groups? *International Journal of Computer Supported Collaborative Learning (ijCSCL), 2*(2–3), 225–246.

Schneider, D. C., Voigt, C., & Betz, G. (2007). ArguNet—a software tool for collaborative argumentation analysis and research. *Paper presented at the 7th Workshop on Computational Models of Natural Argument* (*CMNA VII).*

Schwarz, B. B., & Glassner, A. (2003). The blind and the paralytic: Fostering argumentation in social and scientific issues. In J. Andriessen, M. J. Baker, & D. D. Suthers (Eds.), *Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments* (pp. 227–260). Dordrecht: Kluwer Academic.

Schwarz, B. B., & Glassner, A. (2007). The role of floor control and of ontology in argumentative activities with discussion-based tools. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 2*(4), 449–478.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research, 78*(1), 153–189.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education, 12*, 40–62.

Soller, A. (2004). Computational modeling and analysis of knowledge sharing in collaborative distance learning. *User Modeling and User-Adapted Interaction, 14*(4), 351–381.

Soller, A., Goodman, B. A., Linton, F., & Gaimari, R. (1998). Promoting effective peer interaction in an intelligent collaborative learning system. In B. P. Goettl, H. M. Halff, C. L. Redfield, & V. J. Shute

(Eds.), *Proceedings of the 4th International Conference on Intelligent Tutoring Systems (ITS 1998)* (pp. 186–195). Berlin: Springer.

Soller, A., Monés, A. M., Jermann, P., & Mühlenbrock, M. (2005). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *International Journal on Artificial Intelligence in Education, 15*, 261–290.

Stahl, G. (2007). CSCL and its flash themes. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 2*(4), 359–362.

Stegmann, K., Weinberger, A., & Fischer, F. (2007). Facilitating argumentative knowledge construction with computer-supported collaboration scripts. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 2*(4), 421–447.

Suthers, D. D. (2001). Architectures for computer supported collaborative learning. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001)* (pp. 25–28), Madison.

Suthers, D. D. (2003). Representational guidance for collaborative inquiry. In J. Andriessen, M. J. Baker, & D. D. Suthers (Eds.), *Arguing to learn: Confronting cognitions in computer-supported collaborative learning environments* (pp. 27–46). Dordrecht: Kluwer Academic.

Suthers, D. D., & Hundhausen, C. (2003). An experimental study of the effects of representational guidance on collaborative learning processes. *Journal of the Learning Sciences, 12*(2), 183–219.

Suthers, D. D., Weiner, A., Connelly, J., & Paolucci, M. (1995). Belvedere: Engaging students in critical discussion of science and public policy issues. In J. Greer (Ed.), *Proceedings of the 7th World Conference on Artificial Intelligence in Education (AI-ED 1995)* (pp. 266–273). Charlottesville: Association for the Advancement of Computing in Education.

Suthers, D. D., Toth, E., & Weiner, A. (1997). An integrated approach to implementing collaborative inquiry in the classroom. In R. Hall, N. Miyake, & N. Enyedy (Eds.), *Proceedings of the 2nd International Conference on Computer Support for Collaborative Learning (CSCL 1997)* (pp. 272–279). Toronto: University of Toronto Press.

Suthers, D. D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E. E., Toth, J., et al. (2001). Representational and advisory guidance for students learning scientific inquiry. In K. D. Forbus & P. J. Feltovich (Eds.), *Smart machines in education: The coming revolution in educational technology* (pp. 7–35). Menlo Park: AAAI/MIT.

Suthers, D. D., Vatrapu, R., Medina, R., Joseph, S., & Dwyer, N. (2008). Beyond threaded discussion: Representational guidance in asynchronous collaborative learning environments. *Computers & Education, 50*(4), 1103–1127.

Taentzer, G. (2001). Towards common exchange formats for graphs and graph transformation systems. *Electronic Notes in Theoretical Computer Science, 44*(4), 28–40.

Tannen, D. (1998). *The argument culture: Moving from debate to dialogue*. New York: Random House Trade.

Tedesco, P. (2003). MArCo: Building an artificial conflict mediator to support group planning interactions. *International Journal of Artificial Intelligence in Education, 13*, 117–155.

Thagard, P. (2006). Evaluating explanations in law, science, and everyday life. *Current Directions in Psychological Science, 15*(3), 141–145.

Toulmin, S. E. (1958). *The uses of argument*. Cambridge: Cambridge University Press.

Tsovaltzi, D., Rummel, N., McLaren, B. M., Pinkwart, N., Scheuer, O., Harrer, A., et al. (2010). Extending a virtual chemistry laboratory with a collaboration script to promote conceptual learning. *International Journal of Technology Enhanced Learning (IJTEL), 2*(1–2), 91–110.

van den Braak, S., & Vreeswijk, G. (2006). AVER: Argument visualization for evidential reasoning. In T. M. van Engers (Ed.), *Proceedings of the 19th Conference on Legal Knowledge and Information Systems (JURIX 2006)* (pp. 151–156). Amsterdam: IOS.

van Eemeren, F. H., & Grootendorst, R. (2004). *A systematic theory of argumentation: The Pragma-Dialectical Approach*. Cambridge: Cambridge University Press.

van Gelder, T. (2002). Argument mapping with Reason!Able. *The American Philosophical Association Newsletter on Philosophy and Computers, 2*(1), 85–90.

van Gelder, T. (2003). Enhancing deliberation through computer-supported argument visualization. In P. A. Kirschner, S. J. Buckingham-Shum, & C. S. Carr (Eds.), *Visualizing argumentation: Software tools for collaborative and educational sense-making* (pp. 97–115). London: Springer.

van Gelder, T. (2007). The rationale for rationale. *Law, Probability and Risk, 6*(1–4), 23–42.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227–265.

Vizcaino, A., Contreras, J., Favela, J., & Prieto, M. (2000). An adaptive, collaborative environment to develop good habits in programming. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Proceedings of*

the 5th International Conference on Intelligent Tutoring Systems (ITS 2000) (pp. 262–271). Berlin: Springer.
Verheij, B. (2003). Artificial argument assistants for defeasible argumentation. *Artificial Intelligence, 150*(1–2), 291–324.
Voss, J. F. (2006). Toulmin's model and the solving of ill-structured problems. In D. Hitchcock & B. Verheij (Eds.), *Arguing on the Toulmin model: New essays in argument analysis and evaluation* (pp. 303–311). Berlin: Springer.
Walton, D. (2008). *Informal logic: A pragmatic approach* (2nd ed.). Cambridge: Cambridge University Press.
Walton, D., Reed, C., & Macagno, F. (2008). *Argumentation schemes*. Cambridge: Cambridge University Press.
Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education, 46*(1), 71–95.
Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Los Altos: Kaufmann.
Whitehead, A. N., & Russell, B. (1910). *Principia Mathematica*. Cambridge: The University Press.
Wigmore, J. H. (1931). *The principles of judicial proof* (2nd ed.). Boston: Little, Brown & Co.
Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Kaufmann.
Woolf, B. P., Murray, T., Marshall, D., Dragon, T., Kohler, K., Mattingly, M., et al. (2005). Critical thinking environments for science education. In C. K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence and Education (AI-ED 2005)* (pp. 702–709). Amsterdam: IOS.