Int J Artif Intell Educ (2017) 27:515–533
DOI 10.1007/s40593-016-0115-y

ARTICLE

Argumentation Scheme-Based Argument Generation to Support Feedback in Educational Argument Modeling Systems

Nancy L. Green[1]

Published online: 14 June 2016
© International Artificial Intelligence in Education Society 2016

Abstract This paper describes an educational argument modeling system, GAIL (Genetics Argumentation Inquiry Learning). Using GAIL's graphical interface, learners can select from possible argument content elements (hypotheses, data, etc.) displayed on the screen with which to construct argument diagrams. Unlike previous systems, GAIL uses domain-independent argumentation schemes to generate expert arguments as a knowledge source. By comparing the learner's argument diagram to a generated argument, GAIL can provide problem-specific feedback on both the structure and meaning of the learner's argument, e.g., that the learner's argument contains an irrelevant premise. To generate arguments, the argumentation schemes are instantiated from causal domain models specified by lesson authors. Thus, this approach to generating expert arguments has the potential to be used in other domains. In this paper we describe use of GAIL's Authoring Tool to create the domain model and content elements to be provided for a specific lesson, how expert arguments are generated in GAIL, and how the feedback is produced. As GAIL is a work-in-progress, the paper also describes plans for the next design iteration.

Keywords Educational argument modeling systems · Formative feedback · Argumentation schemes · Critical questions

Introduction

There has been significant interest within the field of science education in argumentation (e.g., Bell and Linn 2000; Bricker and Bell 2008; Jiminéz-Aleixandre et al. 2000;

✉ Nancy L. Green
nlgreen@uncg.edu

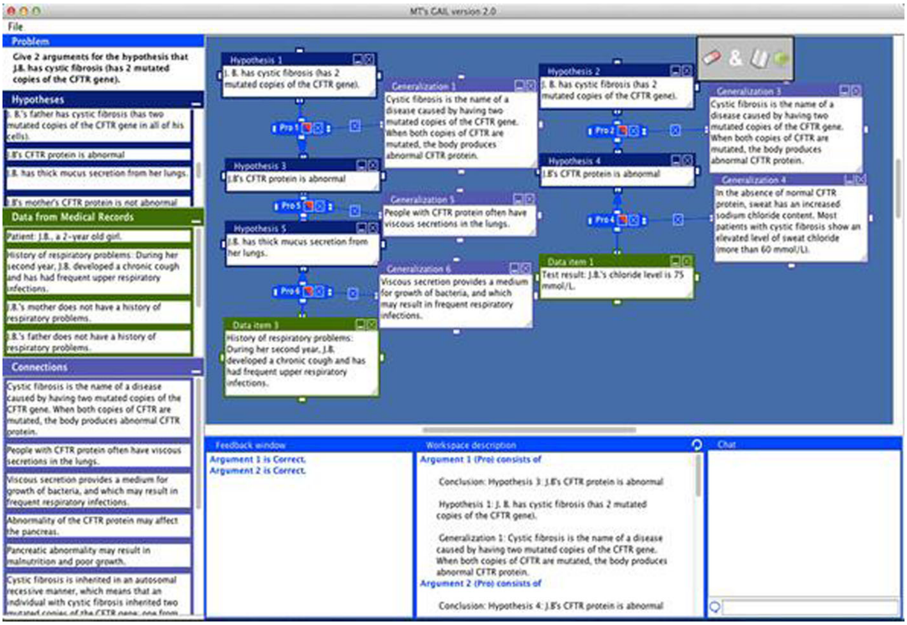
[1] Department of Computer Science, University of North Carolina Greensboro, Greensboro, NC 27402, USA

Sampson and Clark 2008; Sandoval and Reiser 2004; Toth et al. 2002; Zohar and Nemet 2002). According to Bricker and Bell (2008), argumentation should be a "core component of school science"; it can be used to help students learn science content and to help them better understand the nature of the scientific enterprise, scientific discourse, and scientific knowledge. Chinn (2006) contends that "learning to argue well" may enhance content learning, interest and motivation, and problem-solving and writing. However, learning argumentation skills poses significant challenges. Without guidance in how to "argue well," student-produced arguments have been shown to be deficient in a number of ways, e.g., lacking support for claims (Bell and Linn 2000; Jiminéz-Aleixandre et al. 2000), failing to provide alternative explanations (Lawson 2003; Schwarz et al. 2003), and using inaccurate or irrelevant support (Zohar and Nemet 2002).

To begin to address this problem we have implemented a prototype Genetics Argumentation Inquiry Learning (GAIL) system. GAIL supports learning to argue about cases in human genetics, a field that applies findings from genetics research to biomedical reasoning. GAIL is designed for use in an introductory genetics course for undergraduates. Each GAIL lesson requires learners to construct arguments for and against certain hypotheses about a genetics case, e.g., about an infant who may have an inherited metabolic disorder or someone who inherited a genetic variant that is associated with increased risk of colon cancer. A prototype user interface is shown in Fig. 1. Information related to the lesson is provided by GAIL on the left-hand side of the screen: the *Problem* (to give a certain argument); *Hypotheses* (including both correct and incorrect hypotheses); *Data* from medical records about the patient and the patient's biological family; and *Connections,* a list of facts or principles of genetics. The center of the screen shows two arguments constructed by a learner. To construct the arguments, the learner searched for appropriate text components on the left-hand side of the screen, dragged them into the workspace in the center of the screen, and connected the components. Arrows point from support to conclusion. The *Connection* between support and conclusion – known as the *warrant* in argumentation theory (Toulmin 1998) – is linked by a line to the arrow.

In Fig. 1, the problem is to give two arguments for the hypothesis that the patient (referred to as J.B.) has cystic fibrosis, i.e., has two variant alleles of the *CFTR* gene. The leftmost "chain" of arguments begins with data (at the bottom of the argument diagram) about J.B.'s respiratory problems. The learner used that data to support an intermediate hypothesis that J.B. has thickened mucus in the lungs, which is used to support an intermediate hypothesis that J.B. has abnormal CFTR protein, which is used to support the main hypothesis/conclusion that J.B. has cystic fibrosis. Branching from the right hand side of the diagram, connections (warrants) provide justification for each step of the argument. The second argument for the same hypothesis, on the right-hand side of the screen, begins with data about J.B.'s lab test result.

A number of educational argument systems have been developed to support development of argumentation skills in science and other academic areas (Kirschner et al. 2003; Scheuer et al. 2010; Pinkwart and McLaren 2012). *Argument modeling systems* support the creation of new arguments or analysis of existing arguments, often by manipulating graphical objects (Scheuer et al. 2012). Unlike previous educational



Problem

Give 2 arguments for the hypothesis that J.B. has cystic fibrosis (has 2 mutated copies of the CFTR gene).

Hypotheses

J. B.'s father has cystic fibrosis (has two mutated copies of the CFTR gene in all of his cells).

J.B's CFTR protein is abnormal

J.B. has thick mucus secretion from her lungs.

J.B's mother's CFTR protein is not abnormal

Data from Medical Records

Patient: J.B., a 2-year old girl.

History of respiratory problems: During her second year, J.B. developed a chronic cough and has had frequent upper respiratory infections.

J.B.'s mother does not have a history of respiratory problems.

J.B.'s father does not have a history of respiratory problems.

Connections

Cystic fibrosis is the name of a disease caused by having two mutated copies of the CFTR gene. When both copies of CFTR are mutated, the body produces abnormal CFTR protein.

People with CFTR protein often have viscous secretions in the lungs.

Viscous secretion provides a medium for growth of bacteria, and which may result in frequent respiratory infections.

Abnormality of the CFTR protein may affect the pancreas.

Pancreatic abnormality may result in malnutrition and poor growth.

Cystic fibrosis is inherited in an autosomal recessive manner, which means that an individual with cystic fibrosis inherited two mutated copies of the CFTR gene: one from

Hypothesis 1

J. B. has cystic fibrosis (has 2 mutated copies of the CFTR gene).

Generalization 1

Cystic fibrosis is the name of a disease caused by having two mutated copies of the CFTR gene. When both copies of CFTR are mutated, the body produces abnormal CFTR protein.

Hypothesis 3

J.B's CFTR protein is abnormal

Generalization 5

People with CFTR protein often have viscous secretions in the lungs.

Hypothesis 5

J.B. has thick mucus secretion from her lungs.

Generalization 6

Viscous secretion provides a medium for growth of bacteria, and which may result in frequent respiratory infections.

Data item 3

History of respiratory problems: During her second year, J.B. developed a chronic cough and has had frequent upper respiratory infections.

Hypothesis 2

J. B. has cystic fibrosis (has 2 mutated copies of the CFTR gene).

Generalization 3

Cystic fibrosis is the name of a disease caused by having two mutated copies of the CFTR gene. When both copies of CFTR are mutated, the body produces abnormal CFTR protein.

Hypothesis 4

J.B's CFTR protein is abnormal

Generalization 4

In the absence of normal CFTR protein, sweat has an increased sodium chloride content. Most patients with cystic fibrosis show an elevated level of sweat chloride (more than 60 mmol/L).

Data item 1

Test result: J.B.'s chloride level is 75 mmol/L.

Feedback window

Argument 1 is Correct.
Argument 2 is Correct.

Workspace description

Argument 1 (Pro) consists of

Conclusion: Hypothesis 3: J.B's CFTR protein is abnormal

Hypothesis 1: J. B. has cystic fibrosis (has 2 mutated copies of the CFTR gene).

Generalization 1: Cystic fibrosis is the name of a disease caused by having two mutated copies of the CFTR gene. When both copies of CFTR are mutated, the body produces abnormal CFTR protein.

Argument 2 (Pro) consists of

Conclusion: Hypothesis 4: J.B's CFTR protein is abnormal

Chat

Fig. 1 Screen shot of prototype GAIL user interface

argument modeling systems, GAIL generates expert arguments. By comparing the learner's argument diagram to the generated expert arguments, GAIL can provide problem-specific feedback on both the structure and meaning of the learner's argument, e.g., that the learner's argument contains an irrelevant premise. In contrast, previous educational systems could not provide feedback on the meaning of the learner's argument without manually encoding expert arguments. GAIL is a work-in-progress, i.e., although several versions have been implemented and informally evaluated, it is not ready for deployment and evaluation in a classroom setting. In this paper we describe use of GAIL's Authoring Tool, how expert arguments are generated automatically, how the feedback is produced, and plans for the next design iteration.

Argument Generation in GAIL

GAIL uses automatically generated arguments as a key knowledge source for generating feedback. The method for argument generation used in GAIL is similar to one we developed for an earlier application, GenIE, that generates letters to genetic counseling clients explaining the reasons for their diagnosis (Green et al. 2011). That work was based upon analysis of the forms of argument (*argumentation schemes*) used by the counselors and the model of genetics communicated to their clients. In GenIE, textual arguments were generated in two steps. First, an internal propositional representation was generated by the Argument Generator module using non-domain-specific argumentation schemes and a qualitative causal domain model. Then the internal representation was expressed in English by a Text Generation module. Note that only the Argument Generator module has been adapted for use in GAIL. As will be described in section “Feedback Generation in GAIL”, an internal representation of the learner’s argument is compared to the internal representation of the argument created by the GAIL Argument Generator in order to provide intelligent feedback. This section describes how arguments are generated in GAIL using argumentation schemes and qualitative causal domain models.

Background on Argumentation Theory

In formal-logic-based theories of argumentation, an argument is said to consist of a set of *premises* and a *conclusion*, and deductive logic is used to determine whether the argument is valid. Toulmin (1998), who was concerned with modeling argument acceptability in fields such as law and science, argued that logical validity is too restrictive a criterion for determining argument acceptability. Toulmin distinguished two types of premises: *data*, i.e., observations or conclusions of other arguments, and *warrant*, i.e., a field-dependent accepted principle (such as a legal rule or a law of science). More recently, Walton et al. (2008) described *argumentation schemes*, abstract descriptions of forms of argument that are used to construct acceptable arguments in everyday conversation, law, and science. Argumentation schemes may describe non-deductively-valid arguments, and their conclusions may be retracted when more information is obtained. For example, an abductive argumentation scheme used in genetic counseling (Green et al. 2011) describes reasoning from an observation to a hypothesized cause. *Critical questions* associated with argumentation schemes play an important role in evaluating argument acceptability (Walton et al. 2008). For example, one of the critical questions of that abductive argumentation scheme is whether there is an alternative, more plausible explanation for the observation.

Qualitative Causal Domain Models

When starting to develop the earlier GenIE system we discovered that, other than communicating numerical risk information, the information provided by genetic counselors principally involved reasoning based upon a qualitative causal domain model. In order to build such domain communication models, we first identified a small set of recurring types of concepts and causal relations used by genetic counselors in letters written to their clients (Green 2005). For example, the concept type of genotype of the patient and of his biological relatives was prominent, as was the resulting biochemical state, the resulting physiological state, and the resulting symptoms. For computational purposes, individual domain models representing different genetic conditions were then built using the knowledge representation formalism of qualitative probabilistic networks (QPN) (Druzdzel and Henrion 1993). A QPN consists of a directed acyclic graph of random variables and arcs representing qualitative constraints in terms of influence (S^+, S^-), additive synergy (Y^+, Y^-), and product synergy (X^0, X^-) relations. For (Boolean) random variables A, B and C, $S^+(A,B)$ [or $S^-(A,B)$] can be paraphrased as

If A is true then it is more [less] likely that B is true; $Y^{+}(\{A,C\},B)$ *[or* $Y^{-}(\{A,C\},B)$ as *If A and C are true then A enables [prevents] C from leading to B being true;* $X^{0}(\{A,C\},B)$*[or* $X^{-}(\{A,C\},B)$*]* as *if both [either] A and C are true then it is likely that B is true*. For further details on knowledge representation used in GenIE, see (Green et al. 2011).

The same approach is used in GAIL to represent domain knowledge internally. Different causal models can be constructed automatically by the GAIL *Authoring Tool* from XML-format descriptions provided by a lesson author (see section "Authoring Tool"). These XML descriptions link the text presented to the learner on the GAIL graphical user interface to elements of the internal domain model. To illustrate this type of domain model less formally, part of the causal model describing the case of an imaginary patient named J.B. is described in Figs. 2 and 3. Annotations on the arcs in the figure paraphrase the formal constraints (influence and synergy relations) between concepts in the model.

Argumentation Schemes

After developing a domain knowledge representation for the GenIE system, we identified a small set of mainly causal argumentation schemes describing types of arguments used by the genetic counselors in letters to their clients. We next defined these argumentation schemes in terms of formal properties of a QPN. The reason for defining the argumentation schemes in terms of formal properties rather than in terms of specific domain models was to enable the approach

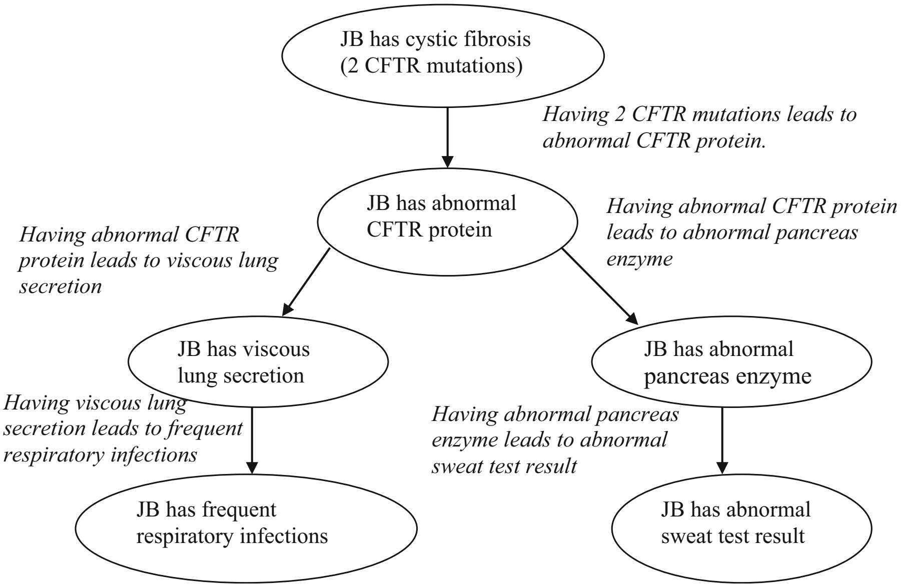

JB has cystic fibrosis (2 CFTR mutations)
Having 2 CFTR mutations leads to abnormal CFTR protein.
JB has abnormal CFTR protein
Having abnormal CFTR protein leads to viscous lung secretion
Having abnormal CFTR protein leads to abnormal pancreas enzyme
JB has viscous lung secretion
JB has abnormal pancreas enzyme
Having viscous lung secretion leads to frequent respiratory infections
Having abnormal pancreas enzyme leads to abnormal sweat test result
JB has frequent respiratory infections
JB has abnormal sweat test result

Fig. 2 Example of internal causal domain model describing effects of patient JB's CFTR genotype

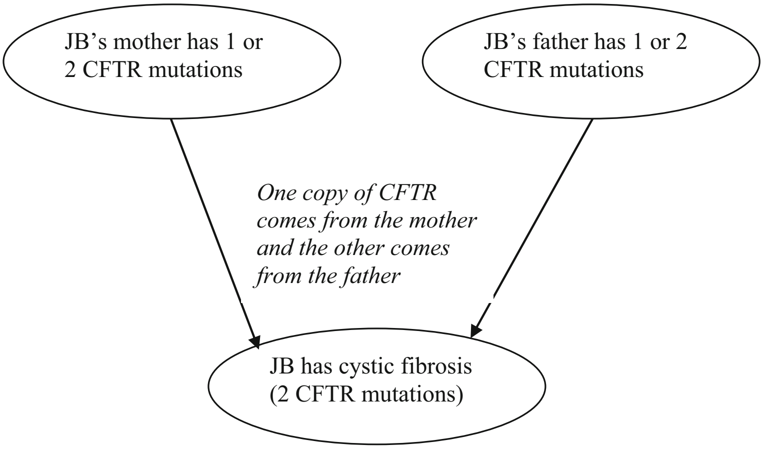

JB's mother has 1 or 2 CFTR mutations
JB's father has 1 or 2 CFTR mutations
One copy of CFTR comes from the mother and the other comes from the father
JB has cystic fibrosis (2 CFTR mutations)

Fig. 3 Example of internal causal domain model describing effect of patient JB's parents' genotypes on JB's genotype

to be reused with other domain models and in different future applications (such as GAIL). The argumentation schemes are represented in structures specifying *claim/conclusion*, *data*, and *warrant*. The propositions used as claim or data describe states of variables in a QPN. The warrant expresses formal constraints on the nodes of the QPN in terms of influence and synergy relations. The distinction between premises as data and warrant reflects their difference in function and source of information. Data premises refer to a particular case, whereas warrants describe biomedical principles and other general knowledge (Toulmin 1998).

For example, the abductive argumentation scheme *Effect to Cause*, shown below, can be instantiated from a causal model to create an argument that a patient has HNPCC (a mutation in the *MLH1* gene, a hereditary condition predisposing one to colon cancer) based upon the data that genetic testing showed a variant *MLH1* allele, and the warrant that having HNPCC typically leads to that test result. Uppercase-initial terms in the argumentation scheme – *A, B, C* – are discrete random variables in the QPN, S^{+} is a positive influence relation, lowercase-initial terms – *a, b, c* – are values of the random variables. For details see (Green et al. 2011). The *condition* for this argumentation scheme, which is not a deductively valid form of argumentation, asks whether there is an alternative explanation for the data. The condition expresses a critical question (see 2.1) of the argumentation scheme.

Effect to cause argumentation scheme

$$\begin{aligned}&\text{Claim}: A \geq a\\&\text{Data}: B \geq b\\&\text{Warrant}: S^{+}(<A, a>, <B, b>)\\&\text{Condition}: \neg\ \text{exists}\ C\ X^{-}(\{C, A\}, <B, b>): C \geq c\end{aligned}$$

Table 1 Argumentation schemes used in current GAIL implementation

Scheme	Data	Warrant	Conclusion	Critical question
E2C	E	C can cause E	C	Alternative explanation?
C2E	C	C can cause E	E	Something preventing E?
NE2C	not E	C can cause E	not C	Something mitigated C?
NC2E	not C	C can cause E	not E	Something else can cause E?
JE2C	E	C1 & C2 can jointly cause E	C1 & C2	Alternative explanation?
JC2E	C1 & C2	C1 & C2 can jointly cause E	E	Something preventing E?
Elim	(A or B) & not A		B	

The argumentation schemes currently used in GAIL, similar to those defined for GenIE, are paraphrased in Table 1. Note that in subsequent work (Green 2015), we have identified additional causal argumentation schemes used in the biomedical genetics research literature that we would like to incorporate into GAIL in the future.

Argument Generator

Following the same approach used in the GenIE system (Green et al. 2011), an argument for a given claim is automatically constructed in GAIL by searching the causal domain model for information fitting argumentation schemes instantiated with the claim. Arguments also can be constructed by chaining and/or conjoining subarguments. For example, the left side of Fig. 4 outlines the structure of a chain of E2C arguments supporting the claim (represented as A in the figure) that the patient JB has cystic fibrosis, i.e., 2 mutated *CFTR* alleles. The right side of Fig. 4 shows an argument for the claim (represented as E in the figure) that JB's mother has exactly one *CFTR* mutation. Note that conclusion A of the argument on the left side is used as a premise in the argument on the right side. The contents of the argument and argumentation scheme are given right after the figure. Note that premises labeled W are warrants.

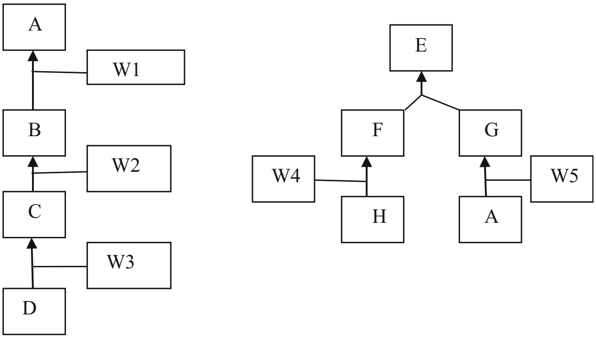


Fig. 4 Structure of two arguments generated by GAIL

E2C:
Premise B: JB has abnormal CFTR protein
Premise W1: Having 2 *CFTR* mutations leads to abnormal CFTR protein
Conclusion A: JB has cystic fibrosis, i.e., 2 mutated *CFTR* alleles
E2C:
Premise C: JB has viscous lung secretions
Premise W2: Having abnormal CFTR protein leads to viscous lung secretion
Conclusion B: JB has abnormal CFTR protein
E2C:
Premise D: JB has frequent respiratory infections
Premise W3: Having viscous lung secretions leads to frequent respiratory infections
Conclusion C: JB has viscous lung secretions
Elim:
Premise F: JB's mother does not have 2 *CFTR* mutations.
Premise G: JB's mother has 1 or 2 *CFTR* mutations.
Conclusion E: JB's mother has exactly one *CFTR* mutation
NE2C:
Premise H: JB's mother does not have cystic fibrosis symptoms.
Premise W4: Having 2 *CFTR* mutations leads to cystic fibrosis symptoms
Conclusion F: JB's mother does not have 2 *CFTR* mutations
JE2C:
Premise A: JB has cystic fibrosis, i.e., 2 mutated *CFTR* alleles
Premise W5: A child inherits one allele from the mother and one from the father
Conclusion G: JB's mother has 1 or 2 *CFTR* mutations.

Unlike previous educational systems in which all possible arguments to be used by the system had to be encoded by an author in natural language (e.g. Woolf et al. 2005) or in propositional logic (e.g. Yuan et al. 2008), GAIL's arguments are automatically generated. The generated arguments are *expert* in the sense that they are derived from the same argumentation schemes and type of domain models used by experts in genetic counseling, and were evaluated for the GenIE system (Green et al. 2011). Also note that as part of our ongoing work on argumentation mining (Green 2015), related argumentation schemes that we identified in biomedical research articles have been validated by other researchers.

Feedback Generation in GAIL

Shute (2008, p. 154) defines formative feedback as "information communicated to the learner that is intended to modify his or her thinking or behavior for the purpose of improving learning." In contrast to summative feedback, it provides "more specific and timely" information about the learner's response. Narciss (2008) classifies types of feedback as follows. Knowledge of performance (KP) is summative feedback informing learners of their overall performance at the end of a task (e.g. their grade). Knowledge of result (KR) informs learners whether their actual response was correct or not. Knowledge of correct result (KCR) provides the correct answer. Answer-until-correct (AUC) and Multiple-try feedback (MTF) approaches allow multiple attempts. Elaborated feedback (EF) elaborates on information given in KR or KCR. In interactive systems, informative tutoring feedback (ITF) provides elaborated feedback "to guide the learner toward successful task completion" (p. 137). Bug-related tutoring feedback (BRT-feedback) is a type of ITF that "allows assisted multiple response tries for an item (a) by providing strategically useful information for error correction, but no immediate KCR, and (b) by requiring the learner to apply the corrective information to a further attempt with this item" (Narciss and Huth 2006, p. 310–1). Results of a study by Narciss and Huth showed that use of BRT-feedback in computer-based training was more beneficial to achievement and motivation for fourth-graders who had learning difficulties with subtraction than providing KR on the first try and KCR on the learner's second try. In their study, BRT-feedback included color-coding of the location of errors, explanation of errors, and hints for correcting the errors. As will be seen in this section, GAIL follows a similar approach, i.e., providing information about the type and location of errors in the learner's argument diagram and hints on how to correct them so that the learner can apply the information to successive attempts to construct a good argument.

Formative Evaluation of Argument Diagramming

Before we describe the process of generating feedback in GAIL, section "Formative Evaluation of Argument Diagramming" describes a formative evaluation of the user interface[1] motivating the need to provide formative feedback on the content of the learner's argument. The study was done in fall 2011 through spring 2012 with ten undergraduate volunteers (Green 2013). The user interface provided possible hypotheses, data, and warrants and argument diagramming tools similar to the user interface shown in Fig. 1. After watching a short tutorial on argumentation and GAIL and reading a patient education document on cystic fibrosis, participants were asked to construct graphical arguments (similar to examples in section "Argument Generation in GAIL") for the diagnosis that J.B. has cystic fibrosis, for and against the diagnosis that J.B.'s brother has cystic fibrosis, and an argument that J.B.'s mother and father each have exactly one CFTR mutation. None of the first seven participants created any acceptable arguments. The three most frequent types of errors in descending order were (1) not providing acceptable evidence for the conclusion; (2) the given warrant did not relate the data to the conclusion; and (3) the main claim of the participant's argument diagram did not match the problem. Since the pilot study so far had not revealed any difficulties in participants' using the interface, and since the participants seemed to have understood the proper graphical syntax for the argument diagrams, it was decided to modify the procedure for the last three participants to see if the participants' success rate could be improved by providing formative feedback and allowing three attempts on each problem. Note that while the first seven participants were biology students, for convenience the last three participants were computer science students. Also, one of the problems was eliminated.

[1] Only the user interface had been implemented, i.e., none of the automatic feedback components had been implemented yet.

Using the following guidelines, which were based upon the types of errors made by the first group of participants, a research assistant gave the participant feedback through a chat window after each attempt:

1. Does the main hypothesis match the problem? If not, tell the student that the hypothesis must match the problem.
2. Is everything OK except that the student has used Pro instead of Con or vice versa? If so, explain the difference.
3. Is the data relevant to the hypothesis (could you make a good argument using that data)? If not, suggest he/she try to use some other data.
4. Is the data relevant but the generalization (warrant) does not link the data to the hypothesis? If yes, suggest he/she try a generalization that links the two.
5. Is the generalization (warrant) relevant (could you make a good argument with it) but the data does not fit the warrant? If yes, suggest that he/she try different data that fits the warrant.
6. Did the student include some data in a conjunction that is unnecessary? If so, suggest that he/she remove the conjuncts that do not fit the warrant.
7. Did the student appear to skip a step in a chained argument that has a sub-argument for the data of the top argument? If yes, help the student break it into the main argument and the sub-argument.

The three most frequent types of errors made by the second group of participants were the same as in the first group. However, the success rate of this group was more encouraging. The first problem was solved on the first attempt by two participants (with no feedback) and by the third participant on the second attempt (after feedback). The second problem was solved only by one out of three participants by the third attempt, although the other two participants had made fewer errors by the third attempt. Two of the three participants solved the third problem on the second attempt (after feedback); the other did not solve it in three attempts.

All ten participants responded to a short user experience survey at the end of the study. Overall, the survey showed a positive user experience with GAIL. In conclusion, the pilot suggested that without feedback or other assistance some students will not have much success using a system such as GAIL, despite their positive attitude towards the system. The rest of this section describes the process of automatic feedback generation in the current version of GAIL, as well as some possible future extensions, to address this problem.

Authoring Tool

Learners create argument diagrams in GAIL by selecting blocks of text from the left-hand side of the screen, dragging them to the central workspace, and connecting the blocks to represent an argument's structure. As described in section "Argument Generation in GAIL", GAIL uses a domain model and argumentation schemes to generate expert arguments in non-linguistic form. The elements of the learner's argument are mapped to an internal representation in the same format as the expert argument. The learner's argument is then compared to the expert argument as a knowledge source for feedback selection/generation, as described in section “Argument Evaluation and Feedback Generation”.

To enable the learner’s argument to be mapped to an internal representation that can be compared to the automatically generated expert argument, the author of an argumentation lesson to be used in GAIL creates an XML-formatted file that contains: (a) strings of natural language text – the problem and possible hypotheses, data, and connections (warrants) – to be displayed to learners on the left-hand side of the graphical user interface; (b) a specification of an internal causal domain model; and (c) mappings of the natural language strings in (a) to concepts and relations in the domain model. Then GAIL’s *Authoring Tool* provides (a) to the user interface, uses (b) to build an internal domain model, and stores (c) to enable GAIL’s *Argument Evaluator* to semantically interpret learners’ argument diagrams.

To illustrate, Fig. 5 contains part of an XML file provided by a lesson author to the Authoring Tool for a lesson similar to the one shown in Fig. 1. In the <ui> section, the file contains the text to appear on the left-hand side of GAIL’s user interface. For example, a hypothesis h1 to be expressed as *J.B. has cystic fibrosis (has 2 mutated copies of the CFTR gene)* is mapped to node 7 of the internal domain model. Similar to

```
<ui>
<question quest="p1" node_id="7">Give 2 arguments for the hypothesis
that J.B. has cystic fibrosis (has 2 mutated copies of the CFTR gene).
</question>
…
<hypothesis hypo="h1" node_id ="7">J. B. has cystic fibrosis (has 2
mutated copies of the CFTR gene).</hypothesis>


…
<!--data-->
…
<generalizations gen="g1" node_id="I1">Cystic fibrosis is the name of
a disease caused by having two mutated copies of the CFTR gene. When
both copies of CFTR are mutated, the body produces abnormal CFTR
protein.</generalizations>
…
</ui>
<genie>
<KB_Node>
 <genotype node_id="7" person_id="3">
 <gene_id>FLAG</gene_id>
 <gene_name>CFTR</gene_name>
 <mutated>2</mutated>
 <autosomal_type>recessive</autosomal_type>
 <disease>Cystic Fibrosis</disease>
 </genotype>
 …
 </KB_Node>
  <KB_Arc>
  <influence_arc arc_id="I1">
  <influence_type>+</influence_type>
  <influence_parent>7</influence_parent>
  <influence_child>9</influence_child>
  </influence_arc>
 …
```

Fig. 5 Excerpts from XML file created by lesson author

the domain model shown in Fig. 2, the domain model is defined by the lesson author in the <genie> section (named in honor of the GenIE system). In that section, node 7 is defined as referring to person 3 (J.B.) and having node type ‘genotype’, gene name ‘CFTR’, and two as the number of mutated alleles. The domain model also specifies that there is an influence (S^+ synergy relation) arc I1 from node 7 to node 9, where node 9 refers to J.B.’s abnormal CFTR protein. This arc is paraphrased in the <ui> section as generalization (warrant) g1.

In summary, the Authoring Tool enables a lesson author to choose any topic that can be modeled in a causal domain model as outlined in section “Argument Generation in GAIL”, and then define the domain model and the text to appear on the user interface. Then the expert arguments will be system-generated using the internal domain model and argumentation schemes as described in section “Argument Generation in GAIL”.

Argument Evaluation and Feedback Generation

After a learner submits an argument diagram to GAIL for evaluation, the *Argument Evaluator’s* task is to evaluate the acceptability of the structure and content of the learners’ argument diagram. First, the learner’s diagram is translated into an argument structure containing domain model concepts and relations. The translation process uses the mappings provided via the *Authoring Tool*. The translated structure is in the same representation as arguments produced automatically by the *Argument Generator*. Then the “best matching” expert argument created for the given problem by the *Argument Generator* is determined. Note that there may be several expert arguments for a claim since different data may support the same claim. The current approach to selecting the “best matching” expert argument is to attempt to overlay the learner’s argument structure on each of the expert structures and to compare the learner’s to the expert’s while traversing the structures top-down. The traversal is complicated by the occurrence of arguments with conjoined premises. Since the order of the conjuncts should not affect the match, permutations of conjoined subtrees are considered. The expert argument with the fewest differences to the learner’s argument identified by this process is selected as the best matching. (Future improvements to this approach are discussed in section “Current Limitations”.)

After the best matching expert argument has been selected, the *Feedback Generator* uses the errors detected in comparing the learner’s argument to the best matching expert argument to instantiate error templates. The current templates are listed below. Template slots, indicated by brackets, are filled with text from the learner’s argument diagram. Note that most of the errors are semantic in nature.

1. Syntactic errors:

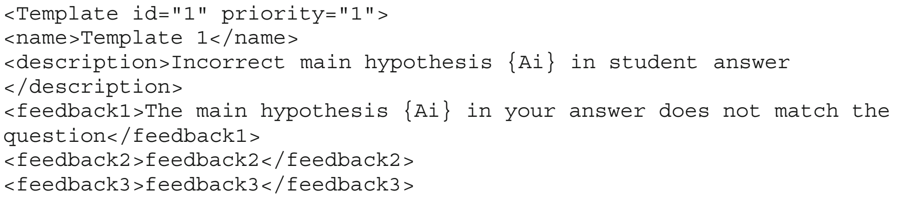
   a. No support is given for <hypothesis>.
   b. The warrant is missing between <data or hypothesis> and <conclusion>.
2. Semantic errors:

   a. <main conclusion> does not match the claim to be argued for in the problem.
   b. <data or hypothesis> does not support <conclusion>.

c. <warrant> is not relevant to <data or hypothesis> and <conclusion>. (That is, it does not provide justification linking data or hypothesis to conclusion.)
d. <data or hypothesis> does not support <conclusion> directly; one or more hypotheses are missing between it and <conclusion>.
e. <data or hypothesis> is not sufficient; the argument requires multiple conjoined premises, but the learner's argument does not include all of them.
f. <data or hypothesis> is given as supporting <data_2 or hypothesis_2> but it should be conjoined to <data_2 or hypothesis_2>.

For each error template, the author of a GAIL lesson can provide a severity code and three levels of feedback message templates in an XML-formatted file. In the current implementation of GAIL, a student is allowed three tries to construct an acceptable argument. After each try, the *Feedback Generator* selects the most general (lowest level) unused message for an error; each time the student makes the same error on a subsequent try, the next more specific (next higher level) message is selected. A positive feedback message is generated when an error is corrected on the next try. The *Feedback Generator* displays only the message for the most serious error to the student, but writes all of the detected errors to a logfile for inspection by the instructor. An excerpt of the feedback message template file is shown in Fig. 6.

To illustrate a learner's interaction with GAIL, suppose the problem was to give an argument for the hypothesis that J.B.'s brother might have malnutrition and poor growth. Internally, GAIL generates a chained argument beginning with the data that J.B.'s brother has been diagnosed as having cystic fibrosis, which supports an intermediate hypothesis that his CFTR protein is abnormal, which supports an intermediate hypothesis that he might have pancreatic abnormality, which supports the main hypothesis that he might (now or in the future) have malnutrition and poor growth. (In addition, the expert argument includes warrants, which are not described here to avoid verbosity.) However, on the first try the learner's argument contains the main claim that J.B. (rather than J.B.'s *brother*) has cystic fibrosis, which does not match the problem. Since this type of error has been given the highest severity code, GAIL would tell the student that the main claim of his argument does not match the problem. On the second try, the student fixes the main claim and constructs a new argument. GAIL informs him that the problem noted on the last try has been fixed. However, the student's argument is missing the intermediate hypothesis that J.B.'s brother might have pancreatic abnormality, so GAIL also informs the student that one or more intermediate hypotheses are missing between J.B.'s brother having abnormal CFTR protein and J.B.'s brother having malnutrition and poor growth. On the third try, the student adds the missing



```
<Template id="1" priority="1">
<name>Template 1</name>
<description>Incorrect main hypothesis {Ai} in student answer
</description>
<feedback1>The main hypothesis {Ai} in your answer does not match the
question</feedback1>
<feedback2>feedback2</feedback2>
<feedback3>feedback3</feedback3>
```

Fig. 6 Example of a feedback message template

hypothesis but provides an irrelevant warrant. GAIL would inform the student that he has made progress but that the warrant he just added is irrelevant to that subargument.

Current Limitations

As stated in the Introduction section, GAIL is a work-in-progress that has not yet been deployed or evaluated in a classroom setting. (All of the components described in section "Argument Generation in GAIL", "Authoring Tool" and "Argument Evaluation and Feedback Generation" have been implemented so far.) There are a number of improvements that we would like to make before deploying GAIL.

Display and Quantity

Currently feedback messages are textual and presented at the bottom of the graphical user interface after the learner submits her argument for evaluation by GAIL. Utilizing these messages places a cognitive burden on the learner, who must figure out which part of her argument diagram is referred to in the message before she can make use of the information. Also, only one feedback message (the highest priority message) is displayed at a time, although the system may have detected multiple problems with the learner's diagram. In the next design iteration, GAIL will provide a copy of the learner's argument diagram to provide feedback. The copy will appear in a non-editable panel vertically aligned with the learner's diagram. Using color encoding to suggest the presence of problems and their severity, the copy will provide the learner with "mouse-over" controls to see feedback messages on different highlighted parts of the copy. This will enable the learner to more easily identify the location of a problem when reading the message about it. This also will enable her to choose which problem(s) to address before resubmitting her diagram for evaluation. Also, rather than limiting the learner to a fixed number of attempts, the learner will be allowed to submit her diagram for feedback any number of times.

Our plans to deliver feedback via user-controlled "mouse-overs" on a copy of the learner's diagram in the next version of GAIL conforms to the guideline of "presenting elaborated feedback in manageable units" (Shute 2008, p. 177). Also, use of this graphical technique follows the guideline to "Exploit the potential of multimedia to avoid cognitive overload" (p. 179). In terms of timing issues, immediate feedback is recommended "to promote learning and performance on verbal [and] procedural … tasks" (p. 179). In addition, "high-achieving students may … benefit by delayed feedback," while "low-achieving students need the support of immediate feedback" (p. 180). In the current version of GAIL, feedback is available only after the learner submits her first, second, and third attempt. In the next iteration of GAIL, however, the learner may submit her diagram at any stage of completion any number of times. Thus, the learner will control when and how often feedback is given.

Comparison to Expert Argument(s)

When there is more than one expert argument supporting a claim, the current version of GAIL uses simple heuristics to select the "best matching" expert argument before proceeding to generate feedback. However, these heuristics may fail to recognize correct subarguments when the pair of arguments being compared get too far out of alignment. To address this issue, we plan to experiment with applying standard graph edit distance algorithms for identifying matching subarguments. Also, in the study reported in (Green 2013), we observed that in some cases the learner's argument may contain elements of multiple expert arguments. For example, as shown in Fig. 7, the learner has simply combined all the data supporting A – D and C – into one argument, whereas in fact there are two distinct expert arguments for A – one based upon C and one based upon D. Another problem with the learner's argument is that although he has identified the correct warrants – W1, W2, and W3 – he has combined them into one warrant rather than showing the correct sequence of inferential steps from B to A, C to B, and D to A. In other words, the learner's argument contains good content but is not structured correctly and needs to be separated into two arguments. Thus, we would like to generalize the feedback algorithm to handle such cases.

Critical Questions

A novel type of feedback is possible in GAIL in the future since it generates expert arguments from argumentation schemes. As discussed in Section "Argument Generation in GAIL", each argumentation scheme is associated with certain critical questions. Having generated an argument by instantiating an argumentation scheme, it is possible to instantiate its critical questions as well. Critical questions support a different type of feedback, encouraging the student to consider counterarguments. To illustrate, one of the critical questions of the *Effect to Cause* argumentation scheme is whether there is another plausible explanation of a certain observation. Now suppose that the learner has constructed an acceptable argument for a diagnosis of cystic fibrosis. Instantiating this critical question from the expert argument could support generating feedback such as *Can you make an argument for a diagnosis other than cystic fibrosis that explains the patient's malnutrition as well as his frequent respiratory infections?* Note that, since abstract, not syntactic, forms of critical questions are associated with argumentation schemes, one could study the varying effectiveness of different ways of asking the same critical question. Other critical questions are summarized in Table 1.

Authoring Tool

Lastly, the Authoring Tool could be made more "author-friendly" by use of a graphical user interface for defining the lesson and building the internal domain model.

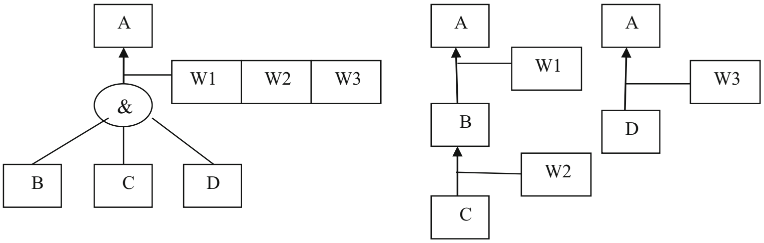


Fig. 7 Learner's argument on left side. Two expert arguments on the right side

Evaluation Plan

To evaluate the potential benefits of GAIL's approach to generating formative feedback, the following study could be performed. GAIL's novel approach affords the provision of feedback, not only on *structural* errors (e.g. omission of supporting data), but also on the *meaning* of the learner's argument (e.g. irrelevant data). Thus, it would be informative to compare the effect of three conditions on achievement and motivation. In all three conditions, GAIL users would be given the opportunity to make unlimited attempts until the argument is correct, or the student gives up, or a timeout occurs (AUC).

- In the baseline condition (KC), learners would be provided only with knowledge of correctness feedback on the argument, indicated by color coding of nodes and arcs of the graphical representation of the learner's argument. When the learner moved the mouse over color-coded parts denoting errors, the same generic message would be provided. Also in this and the other two conditions, at the end of the learner's attempt to construct the argument, a summative KC message would be provided on the argument as a whole.
- In the second condition (KC-SO), learners would be provided with knowledge of correctness feedback as in the baseline condition except that ITF on *structural* (*only*) errors would be provided when the learner moved the mouse over corresponding parts of the graphical representation. Errors due to meaning would be color-coded but only the generic message used in the baseline condition would be provided.
- In the third condition (KC-SM), learners would be provided with knowledge of correctness feedback as in the baseline condition. However, unlike the other two conditions, ITF on structural errors as well as errors due to *meaning* would be provided when the learner moved the mouse over corresponding parts of the graphical representation.

Due to practical constraints, the study would be performed using students in an undergraduate genetics course at our university. The course has a large common lecture section but is divided into separate smaller lab sections. Each condition would be randomly assigned to one of the labs. All students would be given a paper-and-pen pre-test in lecture before the use of GAIL. The pre-test will assess current performance on a similar argument problem, collect demographic information such as gender, age, and computer-usage, and include a brief survey on motivation. Interleaved with regular lab assignments, the students will use GAIL several times over the semester. Data to be collected at each session include number of errors corrected, number of attempts until a correct argument is given or until the learner stops working on a problem, and number of correct arguments. In addition, it will be possible to investigate learners' use of feedback by analysis of logs, e.g., the order in which errors are addressed or whether feedback is acted upon or not. After the last use of GAIL, all students will be given a paper-and-pen post-test in lecture similar to the pre-test. Our hypothesis is that KC-SM will be more beneficial than KC-SO, which in turn will be more beneficial than KC, for learner achievement and motivation. The influence of individual differences in demographics, initial performance, and initial motivation on the effects will be analyzed also.

Related Work

Scheuer et al. (2012) present a survey of automatic analysis and feedback techniques used in educational argument modeling systems. One approach is to detect syntactic patterns in the student's argument diagram that suggest that the argument is not acceptable, e.g. in Belvedere (Suthers et al. 2001) and LARGO (Pinkwart et al. 2006). However, such an approach cannot detect content errors or give problem-specific hints. Thus, an additional approach is used in Belvedere and LARGO and in other systems, e.g. Rashi (Woolf et al. 2005) and CATO (Aleven and Ashley 1997), namely, to compare the student's argument to an expert's manually encoded solution. As Scheuer et al. note, a disadvantage is the effort required to encode expert solutions. In contrast to the above approaches, GAIL generates expert arguments as a knowledge source for feedback selection/generation. Another problem noted by Scheuer et al. with using comparison to an expert model is that feedback is based upon heuristics since the system does not represent the expert's argument computationally. An educational system that is based upon a computational model of dialogue moves enables a student to engage in human-computer debates (Yuan et al. 2008). However, the debate content must be expressed in propositional logic and all deductively valid arguments and counterarguments must be predefined by an expert. In contrast, GAIL generates expert arguments using a computational model of argumentation based upon argumentation schemes rather than deductive logic. Note that future versions of GAIL could support human-computer debate using GAIL's computational model to generate counterarguments.

Conclusion

Due to the fundamental role of argumentation in science and the significant problems that have been observed in students' arguments, science educators have long advocated providing support to improve argumentation skills. This paper describes the design of GAIL, an educational argument modeling system for learning to argue about cases in human genetics. Unlike other educational systems, GAIL uses a unique argumentation-scheme-based approach to generate arguments, which are used as a knowledge source for generating feedback on structure and content of the learner's argument. Argumentation schemes are abstract non-domain-specific patterns of reasoning used in fields such as science and law. The argumentation schemes used in GAIL refer to formal properties of causal domain models. To create an argumentation lesson for GAIL, rather than manually encoding all expert arguments, a lesson author specifies a causal domain model for the topic, and mappings from certain hypotheses, data, and general information (warrants) appearing on the user interface to concepts and relations in the domain model. Thus, GAIL's approach has the potential to be used in other domains beyond the current implementation for genetics.

Acknowledgments Former graduate students Mark Hinshaw, Carl Martensen, Meghana Narasimhan, and Tshering Tobgay contributed to the most recent implementation of GAIL for their MS Projects. Former graduate students Benjamin Wyatt and Chris Cain also contributed to the implementation of GAIL. Wyatt and Martensen received support from a UNCG Regular Faculty grant and Cain received support from the Computer Science Department. Dr. Malcolm Schug of the UNCG Biology Department has provided helpful feedback on the project.

References

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.), *Proc. 8th World Conf. on Artificial Intelligence in Education (AIED-97)* (pp. 87–94). Amsterdam: Ios Press.

Bell, P., & Linn, M. (2000). Scientific arguments as learning artifacts: designing for learning from the web with KIE. *International Journal of Science Education, 22*(8), 797–817.

Bricker, L. A. & Bell, P. (2008). Conceptualizations of argumentation from science studies and the learning sciences and their implications for the practices of science education. *Science Education, 92*, 473–498.

Chinn, C. A. (2006). Learning to argue. In A.M. O'Donnell, C.E. Hmelo-Silver, & G. Erkins (Eds.), *Collaborative learning, reasoning, and technology* (pp. 355–383). Mahwah, NJ: Erlbaum.

Druzdzel, M. J. & Henrion, M. (1993). Efficient reasoning in qualitative probabilistic networks. In *Proc. of the 11th Nat. Conf. on AI (AAAI-93)* (pp. 548–53).

Green, N. L. (2005). A Bayesian network-based coding scheme for annotating biomedical information presented to genetic counseling clients. *Journal of Biomedical Informatics, 38*, 130–144.

Green, N. L. (2013). Towards formative feedback on student arguments. In *Proceedings of the Workshop on Formative Feedback in Intelligent Learning Environment, at Artificial Intelligence in Education 2013.*

Green, N. L. (2015). Identifying argumentation schemes in genetics research articles. In *Second Workshop on Argumentation Mining*, Denver, CO USA, June 4, 2015.

Green, N. L., Dwight, R., Navoraphan, K., & Stadler, B. (2011). Natural language generation of biomedical argumentation for lay audiences. *Argument and Computation, 2*(1), 23–50.

Jiminéz-Aleixandre, M., Rodriguez, M., & Duschl, R. A. (2000). 'Doing the lesson' or 'doing science': argument in high school genetics. *Science Education, 84*(6), 757–792.

Kirschner, P. A., Buckingham Shum, S. J., & Carr, C. S. (Eds.). (2003). *Visualizing argumentation*. London: Springer.

Lawson, A. (2003). The nature and development of hypothetico-predictive argumentation with implications for science teaching. *International Journal of Science Education 25*(11), 1387–1408.

Narciss, S. (2008). Feedback strategies for interactive learning tasks. In J. M. Spector, M. D. Merrill, J. van Merriënboer, & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology* (3rd ed., pp. 125–143). New York: Lawrence Erlbaum Associates.

Narciss, S., & Huth, K. (2006). Fostering achievement and motivation with bug-related tutoring feedback in a computer-based training for written subtraction. *Learning and Instruction, 16*, 310–322.

Pinkwart, N., & McLaren, B. M. (Eds.). (2012). *Educational technologies for teaching argumentation skills*. Sharjah: Bentham Science Publishers.

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006). Toward legal argument instruction with graph grammars and collaborative filtering techniques. In M. Ikeda, K. Ashley, & T. W. Chan (Eds.), *Proc. of the 8th Int. Conf. on Intelligent Tutoring Systems (ITS-06)* (pp. 227–236). Berlin: Springer.

Sampson, V., & Clark, D. B. (2008). Assessment of the ways students generate arguments in science education: current perspectives and recommendations for future directions. *Science Education 92*(3), 447–472.

Sandoval, W. A. & Reiser, B. J. (2004). Explanation-driven inquiry: integrating conceptual and epistemic scaffolds for scientific inquiry. *Science Education 88*, 345–372.

Scheuer, O., Loll, F., Pinkwart, N., & McLaren, B. M. (2010). Computer-supported argumentation: a review of the state of the art. *Computer-Supported Collaborative Learning, 5*(1), 43–102.

Scheuer, O., McLaren, B. M, Loll, F., Pinkwart, N. (2012). Automated analysis and feedback techniques to support and teach argumentation: A survey. In N. Pinkwart, B. M. McLaren (Eds.), *Educational Technologies for Teaching Argumentation Skills*, Bentham Science Publishers.

Schwarz, B., Neuman, Y., Gil, J., & Ilya, M. (2003). Construction of collective and individual knowledge in argumentative activity. *Journal of the Learning Sciences, 12*(2), 219–256.

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

Suthers, D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E., Toth, J., & Weiner, A. (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 Press.

Toth, E. E., Suthers, D. D., & Lesgold, A. (2002). "Mapping to know": the effects of representational guidance and reflective assessment on scientific inquiry. *Science Education 86*, 264–286.

Toulmin, S. E. (1998). *The uses of argument*. Cambridge: Cambridge University Press.

Walton, D., Reed, C., & Macagno, F. (2008). *Argumentation schemes*. Cambridge: Cambridge University Press.

Woolf, B., Murray, T., Marshall, D., Dragon, T., Kohler, K., Mattingly, M., Bruno, M., Murray, D., & Sammons, J. (2005). Critical thinking environments for science education. In C. K. Looi, G. McCalla, B. Brewdeweg, & J. Breuker (Eds.), *Proc. 12th Intl. Conf. on AI and Education* (pp. 702–709). Amsterdam: IOS.

Yuan, T., Moore, D., & Grierson, A. (2008). A human-computer dialogue system for educational debate: a computational dialectics approach. *International Journal of AI in Education, 18*, 3–26.

Zohar, A., & Nemet, F. (2002). Fostering students' knowledge and argumentation skills through dilemmas in human genetics. *Journal of Research in Science Teaching, 39*(1), 35–62.