Towards Formative Feedback on Student Arguments

Nancy L. Green
University of North Carolina
Greensboro
Greensboro, NC 27402 USA
1-336-256-1133
nlgreen@uncg.edu

ABSTRACT
This paper presents our ideas on generating formative feedback in the Genetics Argumentation Inquiry Learning (GAIL) system. GAIL will provide undergraduate biology students with tools for constructing Toulmin-style arguments on questions in genetics. Feedback will be based in part on the output of GAIL's argument analyzer, which will compare learner arguments to automatically constructed expert arguments. In addition to identifying problems in the learner's arguments, the analyzer will recognize the argumentation scheme used to construct acceptable arguments. From that, GAIL can instantiate critical questions, a unique form of feedback in intelligent learning environments.

Keywords
Educational Argumentation Systems, Undergraduate Genetics Education.

1. INTRODUCTION
We are developing the Genetics Argumentation Inquiry Learning (GAIL) system for improving undergraduate biology students' argumentation skills in the domain of genetics. As in many educational argumentation systems, GAIL will provide the learner with tools for representing arguments in diagrams due to the cognitive benefit of diagrams [1-3]. In addition, educational systems can exploit the learner's argument diagram as a source of information for providing educational feedback. A prototype graphical user interface (GUI) for GAIL is shown in Figure 1. The top left-hand side of the screen presents a problem, e.g., to make an argument for the claim that J.B., an imaginary patient, has the genetic condition called cystic fibrosis. Below that are possible hypotheses, data about the patient and his biological family members, and biomedical principles that may be relevant to the current problem. The learner can drag these elements into the argument diagramming workspace in the center of the screen to construct an argument in a Toulmin-influenced [4] box-and-arrow notation; a vertical arrow from the *data* points upward to the *claim/conclusion* and the *warrant* is attached at a right-angle to the arrow.

In this paper we describe our planned approach to providing formative feedback based upon automatic analysis of learners' argument diagrams. Expert models for argument analysis will be automatically constructed by GAIL using an argument generator module similar to the argument generator developed for the GenIE Assistant [5]. The expert model will contain all acceptable arguments that can be generated automatically for a given claim from an underlying knowledge base (KB) representing the problem domain. GAIL's argument analyzer will compare the user's argument to the generated expert arguments to identify acceptable learner arguments and weaknesses in the learner's argument. Weaknesses in student arguments are identified using non-domain-specific, non-content-specific rules that recognize common error types, e.g., those observed in a pilot study reported in section 3. In addition, if an argument is acceptable, the analyzer will recognize and output the argumentation scheme underlying the student's argument and its associated critical questions. The output of GAIL's argument analyzer will be utilized by GAIL's feedback generator to provide formative feedback.

In some previous educational argumentation systems, the student's argument diagram is compared to a manually-constructed expert model to provide problem-specific support. However, expert models are expensive to construct and may not cover all possible solutions or errors [6]. In GAIL's approach the expert model is constructed automatically. Other systems use simulation of reasoning to evaluate formal validity but do not provide problem-specific support [6]. GAIL's approach is similar in that it reasons like an expert to generate an argument. Unlike those systems, however, GAIL's approach will provide problem-specific support.

This paper presents how the expert model is generated (section 2), a pilot study of GAIL's GUI prototype that motivated the classification of weaknesses in learners' arguments (section 3), implementation of a prototype argument analyzer (section 4), some issues to be addressed in the planned feedback generator (section 5), and conclusions (section 6).

2. EXPERT MODEL
Generation of expert arguments in GAIL will be done following the approach to argument generation used in the GenIE Assistant, a proof-of-concept system for generating first-drafts of genetic counseling patient letters [5]. Written by genetic counselors to their clients, this type of letter contains biomedical arguments to justify diagnostic testing, the diagnosis of genetic conditions, and the probable genotypes of family members. GenIE's internal components include

- *domain models*, causal models of genetic conditions used by genetic counselors in communication with their clients [7],
- an *argumentation engine* that uses computational definitions of *argumentation schemes* [8] to guide search in the domain model for data and warrant needed to support a particular claim, and
- a *letter drafter* that organizes and expresses the arguments as English text using natural language generation techniques.

GAIL's expert arguments will be produced using a similar approach to the GenIE Assistant's domain models and argumentation engine. However, the natural language generation module, the letter drafter, will not be needed to generate expert arguments.

The domain models in the GenIE Assistant are represented computationally as qualitative probabilistic networks (QPN) [9]. A QPN consists in part of a directed acyclic graph whose nodes are random variables. In addition, a QPN specifies qualitative constraints on variables 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*.

To illustrate S^+, if a patient has two mutated BRCA1 alleles then it is more likely she will develop breast cancer; *Y+*, someone who has inherited a genetic mutation for familial hypercholesterolemia is at a higher risk of heart disease if she is obese; X^-, breast cancer can be caused by mutation of BRCA1 or some other gene; and X^0, together the mother and the father can pass an autosomal recessive mutation to their offspring. A QPN representing knowledge about a genetic condition can be reused for different patient cases. Representative domain models for testing the GenIE Assistant were built quickly using information from genetics reference books. The size of a QPN to be used in GAIL would be of the same scale as those used to generate letters in the GenIE Assistant (less than 50 nodes). For more information on domain modeling see [5].

Computational definitions of argumentation schemes are used by the GenIE Assistant's argumentation engine to construct a genetic counselor's arguments for the diagnosis and genotypes of family members [5]. The argumentation schemes are formalized in a structure including *claim*, *data*, and *warrant*. Since the argumentation engine and schemes do not encode domain-specific or patient case-specific content, they can be used to generate arguments in any domain whose domain knowledge can be represented in a similar format. 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 mentioned above. The distinction between the two types of premises reflects their difference in function and source of information. Claims and data are facts or hypotheses about a particular case, whereas warrants describe (biomedical or other) generalizations.

In addition to those components, argumentation schemes in the GenIE Assistant include a field called the *applicability constraint*, a constraint that must be true to generate an argument from that scheme. Note that conclusions of the argumentation schemes are not necessarily deductively valid, and the *applicability constraint* is a type of critical question [8]. As discussed in section 5, the *critical questions* of GAIL's argumentation schemes provide a systematic means of challenging the conclusion of an argument.

To illustrate, consider an abductive reasoning scheme used in the GenIE Assistant:

Claim: $A \geq a$
Data: $B \geq b$
Warrant: $S^*(<A,a>, <B,b>)$
App. constraint: $\neg$ *exists* $C\ X^-(\{C,A\},<B,b>): C \geq c$

In the above, uppercase-initial terms -- *A, B, C* -- are random variables in the QPN, S^* is a chain of one or more positive influence relations S^+. Lowercase-initial terms – *a, b, c* – are values of the random variables, and in this scheme are threshold values. To paraphrase this scheme, (warrant) there is a (chain of) possible positive causal influence(s) from *A* to *B*; (data) *B* is at least *b*; therefore (claim) *A* is at least *a*; (applicability constraint) provided that there is no *C* such that *C* and *A* are mutually exclusive positive influences on *B* and *C* is at least *c*. For example, (warrant) having a genotype with two mutated alleles of CFTR can lead to (abnormal CFTR protein which can lead to abnormal pancreas enzyme level which can lead to) growth failure; (data) this patient has growth failure; therefore (claim) this patient has cystic fibrosis; (applicability constraint) as long as there is no other condition believed to explain growth failure.

An argument for a given claim is automatically constructed by searching the domain model and data about the patient's case for information fitting GenIE's argumentation schemes instantiated with the claim. In addition to the above abductive argumentation scheme, other schemes support abductive reasoning about alternative causes or jointly necessary causes, reasoning from cause to effect, reasoning from negative evidence, and reasoning by elimination of alternatives. The argumentation schemes reflect those used in a corpus of genetic counselor-authored letters. Note that the GenIE Assistant's argumentation engine can construct complex arguments involving multiple pieces of evidence and chains of arguments. The same approach will be used in GAIL to generate expert arguments for a given claim. In a performance evaluation of the GenIE Assistant, two letters, each containing multiple arguments, were generated in 22 seconds on a desktop computer [5]. Note that the time should be less than that in GAIL, since the arguments will not be realized in English. Also, they can be generated off-line if necessary.

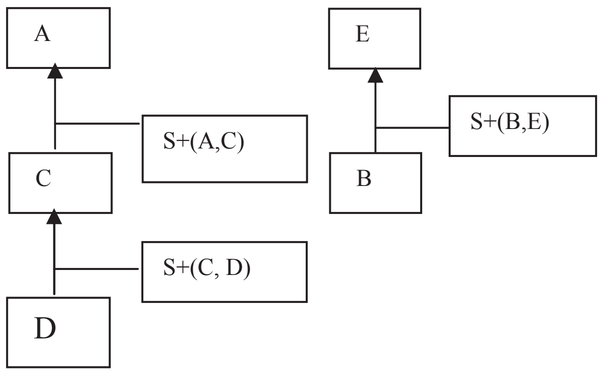

Fig. 2. Example of simple argument structures.

Some example arguments that can be generated are illustrated in Figures 2 and 3 in the box and arrow style of notation used in the GAIL interface. (To save space, the diagrams contain variables rather than the text that would be used in the GUI.) The diagram on the left of Figure 2 is a chain of two abductive arguments. The claim (A) that patient P has cystic fibrosis (two mutated CFTR alleles) is supported by the hypothesis (C) that P has abnormal CFTR protein and is warranted by the positive influence relation between CFTR alleles and CFTR protein. Hypothesis C is supported by the data (D) that P has frequent respiratory infections and the positive influence relation between CFTR protein and respiratory infections. The diagram on the right of

Figure 2 is a causal/predictive argument for the claim (E) that individual M (the patient's mother) is a carrier of a CFTR mutation. E is supported by the family history data that M has a certain ethnicity and is warranted by the higher probability of being a carrier if an individual has that ethnic background.

Figure 3 shows part of an argument for the claim (A=1) that P's mother has exactly one mutated CFTR allele. The left-hand subargument is for the hypothesis that she has one or two mutated CFTR alleles. That subargument is supported by the hypothesis (D=2) that P has cystic fibrosis (two mutated CFTR alleles), and is warranted by the synergy relation, $X^0(<A=1,B=1>, D=2)$, i.e., that a child who has two mutated alleles inherited one from the mother and one from the father. Note that the claim D=2 would be supported by another subargument (not shown in Figure 3). The right-hand subargument is for the hypothesis that the mother does not have two mutated CFTR alleles. This is supported by the data (¬C) that she does not have cystic fibrosis symptoms, and warranted by the positive influence relation between CFTR alleles and symptoms of cystic fibrosis.

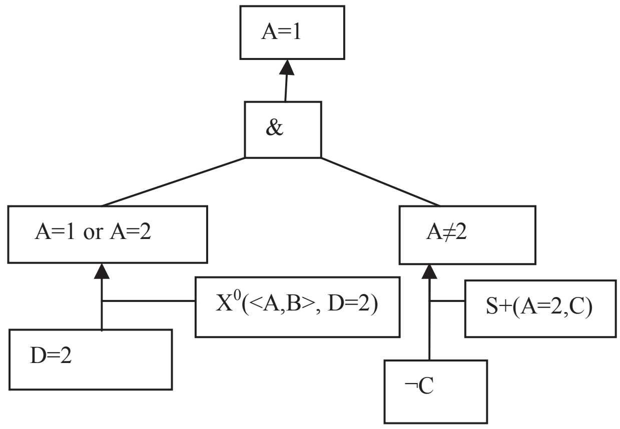

Fig. 3. Example of part of more complex argument.

3. PILOT STUDY

A formative evaluation of GAIL's prototype user interface was done in fall 2011 through spring 2012 with a total of 10 paid undergraduate volunteers, the first seven of which were recruited from biology classes and the last three computer science students. Each participant was first asked to read a seven-page patient education document, which we had found on the internet and printed for this study, on the inheritance and diagnosis of cystic fibrosis. After a participant read the document, it was put away and the research assistant narrated a silent video tutorial describing the components of an acceptable argument, and showing the features of the GAIL GUI and the process of constructing several different arguments using GAIL. Afterwards, the research assistant pointed out a chat box in the GAIL GUI for communicating with the assistant if necessary. The assistant then left the room, but could view the participant's computer screen on another computer monitor.

Listed in the upper left-hand corner of the GAIL GUI, the problems for which the first seven participants were asked to construct arguments are as follows.

Problem 1: Give two arguments for the diagnosis that J.B. has cystic fibrosis.

Problem 2: Give one argument for the diagnosis that J.B.'s brother has cystic fibrosis.

Problem 3: Give one argument against the diagnosis that J.B.'s brother has cystic fibrosis.

Problem 4: Give one argument for hypothesis that J.B.'s mother and father are both "carriers" of the CFTR gene mutation that causes cystic fibrosis

Note that the hypotheses, observations, generalizations (warrants), and problems shown on GAIL were written by the author of this paper based on information from a college genetics textbook. (J.B. refers to a fictitious patient.)

None of the first seven students created acceptable arguments. At that point in the study, it was decided to modify the materials and procedure. First, the problems were reduced in number (eliminating Problem 2, requiring an argument with conjunction). Second, when the participant submitted a response, the research assistant reviewed it using a checklist of error types created by the author after reviewing the arguments created by the first group of participants. If the participant's response contained any of those types of errors then the research assistant gave the participant feedback (as discussed below) through the chat box and asked the student to revise his argument. After three tries, the student was told to proceed to the next problem in the set. Third, to expedite the revised study, the remaining three students were recruited from computer science.

The distribution of error types is shown in Table 1. A Type 1 error was an argument whose claim did not match the claim for which the student was asked to give an argument. Type 2 was an argument where the data was not evidence for the claim. Type 3 was an argument where the warrant did not relate the data to the claim. Type 4 was an argument where the opposite type of link was required. Type 5 was a chained argument in which a subargument was missing or incorrect. For example, consider the chained argument on the left of Figure 2. If the learner failed to give a subargument in support of C, or if the learner skipped the intermediate conclusion C and showed D as directly supporting A, the error would be classified as Type 5. Type 6 errors involved incorrect use of conjunctions. Type 7 was omission of the warrant.

Table 1. Average number of errors per error type per person in each group

Error Type	Group 1	Group 2
1:Incorrect claim	1.9	0.8
2:Incorrect data	2.6	0.3
3:Incorrect warrant	2	1
4:Incorrect pro/con	0.9	0.3
5:Incorrect/missing chained claim	1.4	0
6:Incorrect/missing conjunction	0.9	NA
7: Missing warrant	0.1	0.4

In Table 1, Group 1 comprises the first seven students, who were given no feedback. Group 2 comprises the last three students, who were given feedback and three tries on each problem. The number of errors on each try for each student in Group 2 was totaled and the average was computed by dividing by nine (i.e., three students with three tries each). From the first group, it can be seen that the most frequent errors (in descending frequency) were incorrect data, incorrect warrant, and incorrect claim. Although the quantity of errors in the first and second groups cannot be compared, it should be noted that the top three error types in Group 1 remained the top three in Group 2.

Group 2 received feedback from the research assistant based on the following guidelines:

1. Does the 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.

Table 2 shows the types of errors made by the three students in Group 2 after receiving feedback on their first and second answers on each problem. Problem 1 was solved correctly by two students on the first try, and by the third student on the second try. Problems 2 and 3 were solved correctly by only one student (on the third try). Problem 3 was solved correctly by two students on the second try. These results suggest that on the more difficult problems (Problems 2 and 3), the feedback may have helped to reduce the number of errors.

Table 2. Types of errors in group 2 (after feedback).

Student	Try	Problem 1	Problem 2	Problem 3
1	1st		1, 3 ,4	2, 3
	2nd		1, 3	7
	3rd		3, 4	2, 7
2	1st	1	1, 3	1, 7
	2nd		1, 3	
	3rd		1	
3	1st		3, 4	2, 3, 7
	2nd		3	
	3rd			

At the end of the session, students were asked to complete a user experience survey. The survey results, shown in Table 3, indicate that the students had a favorable response to using the software despite making errors.

Table 3. Average scores on user experience survey (N=10). Possible responses: 3(True), 2(Somewhat true), 1(False).

Question	Score
My background … helped me answer the problems in this study.	2.3
I found the subject of genetic conditions and inheritance interesting.	3
I found the tools for diagramming arguments easy to use.	2.8
I found the tutorial on how to use the argument diagramming tools helpful.	3
I prefer using the argument diagramming tools to writing arguments.	2.7
I would like to use a program like this in my courses on genetics	2.9

4. ARGUMENT ANALYZER

The expert model will contain all acceptable arguments that can be automatically generated for a given claim from an underlying knowledge base (KB) representing the problem domain. The generated arguments are simple or complex argument structures containing KB elements. Text elements provided to the learner through GAIL's GUI are linked internally to KB elements. The inputs to GAIL's argument analyzer will be the learner's argument and the expert model, both in the same format. Implemented in Prolog, the prototype argument analyzer determines if a student's argument diagram represents an acceptable argument and if not acceptable, identifies its weaknesses.

The algorithm to determine acceptability merely checks whether the user's argument matches one of the acceptable arguments. If the user's argument does not match an acceptable argument, its weaknesses are identified using pattern-matching rules motivated mainly by the types of errors seen in the study described in the previous section. The rules are non-domain-specific and non-problem-specific. For example, if the user's data and claim match the expert's, but the warrant does not, the analyzer identifies the problem as an unacceptable warrant (Type 3). The prototype argument analyzer implementation outputs an error message for each error detected. However, in the future implementation of GAIL, the argument analyzer's output would be used by the Feedback Generator, which will be responsible for selecting which error(s) to highlight and providing appropriate feedback.

If the learner's argument is acceptable, i.e., it matches an expert argument, then knowledge of the argumentation scheme used to generate the expert argument provides an additional resource for generation of feedback as described in the next section.

5. FEEDBACK GENERATOR

The feedback generator has not been implemented yet. Currently, we are gathering information to guide its design. As discussed in the previous section, the feedback generator will have access to the output of the argument analyzer. If the learner's argument contains errors such as those types listed in Table 1, some design questions are: which of the errors to address (and in what order), when to provide feedback, what feedback content to provide, and in what syntactic form. Before designing a feedback generator that answers these questions, we are running a think-aloud study to get a better understanding of why students make these errors. For example, a type 4 error might be due to a misunderstanding of the argument representation used in GAIL's GUI. If that is indeed the case, then it would seem that addressing such an error should be given higher priority by the feedback generator. On the other hand, we hypothesize that a type 1, 2 or 3 error may be due to a deeper problem, either in the learner's understanding of what constitutes an acceptable argument, or in understanding the genetics information provided by GAIL as possible building blocks for the learner's argument diagram.

A key point to note is that our approach supports content-based feedback. Many of the types of errors listed in Table 1 are content-based errors that can be detected by the argument analyzer based on the expert model. In addition to using it to identify content-based errors, GAIL will be able to use the expert model to provide content-based feedback. This is illustrated in the following imaginary scenario. Figure 4 depicts abstractly a student argument diagram in which the data, B, is not related by the warrant, S+(A,C), to the conclusion A. Our approach supports providing feedback to the effect that this argument is not acceptable because the warrant does not relate the data to the conclusion; and supports giving the advice to look for other data that is consistent with the given warrant or to look for another warrant that links the given data to the conclusion. Suppose that the expert model contains an argument similar to that in Fig. 4, but using C as data. If the student is unable to make use of the more general advice to replace the data or warrant in the diagram, a hint could then be generated asking whether C is in the observations or hypotheses on the GUI screen.

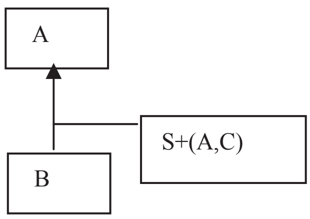

Fig. 4. Abstract example of unacceptable argument.

Figure 5 shows that with the help of this feedback, the imaginary student has replaced the data in the argument diagram with C. However, suppose that C was listed on the GUI screen as a hypothesis rather than an observation. In that case, a sub-argument for C would be required. The argument analyzer could recognize that the sub-argument for C in the expert model is missing in the student's diagram. Then the feedback generator could inform the student that C must be supported by a sub-argument since it is only a hypothesis.

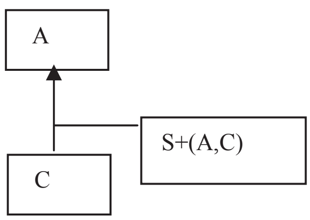

Fig. 5. Abstract example of partly fixed, unacceptable argument.

Figure 6 shows that with the help of this feedback the student adds a sub-argument for C to the diagram, matching an acceptable expert argument.

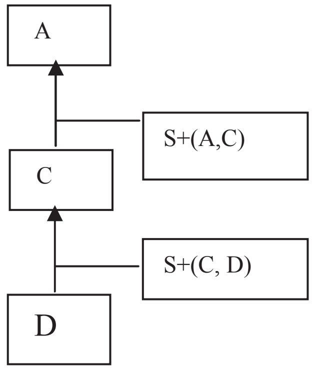

Fig. 6. Abstract example of acceptable argument.

In this domain, however, the conclusions of acceptable arguments are not necessarily deductively valid. As discussed in Section 2, each abstract argumentation scheme is associated with certain critical questions, which provide a way of challenging an argument constructed from that scheme. Critical questions support a different type of feedback, which could inspire a learner to consider multiple arguments pro and con the same claim. To illustrate, one of the critical questions of the abductive argumentation scheme is whether there is another plausible explanation of a certain observation. Having recognized the learner's argument as an instance of this scheme, the feedback generator could instantiate this critical question. Suppose that the learner has constructed an acceptable abductive argument for a diagnosis of cystic fibrosis; instantiating this critical question could support generating feedback such as *Can you make an argument for an alternative diagnosis that explains the patient's frequent respiratory infections?* or, *What if he has some other condition that could explain those symptoms?*

Some other critical questions of GAIL's abductive argumentation schemes, where B is an observation and A is a putative cause of B, include (Green 2010):

- **(Missing Enabler)** is there a C such that C is required for A to cause B, and C is absent? (Example: *Has exposure to bacteria occurred, which is required for thickened mucous to lead to frequent respiratory infections?*)
- **(Mitigation)** is there a C whose presence may mitigate the effect of A on B? (Example: *Is the patient taking antibiotics, which will prevent respiratory infections?)*
- **(Inapplicable Warrant)** Despite the similarity of individual I to the population described by the warrant, is there is a difference that could make it inapplicable to I? (Example: *Although the mother is from a geographic region with a high rate of cystic fibrosis, is her ethnic background different from most of the population there?*)
- **(False Positive)** Is $p(\neg A \mid B)$ too high? (Example: *Is the false positive rate for the laboratory test used to diagnose this condition high?*)
- **(Low Certainty of Data)** Is $p(B)$ too low? (Example: *Are we confident that there is accurate information about the health of the biological mother who gave the patient up for adoption when he was an infant?*)

Again note that feedback can be given without requiring problem-specific knowledge to be embedded in the feedback generator. Also note that semantic, not syntactic, forms of critical questions are associated with argumentation schemes. Thus, using natural language generation from semantic forms to generate syntactic variations, one could study the varying effectiveness of different ways of asking the same critical question.

6. CONCLUSIONS

This paper presents our ideas on generating formative feedback in the Genetics Argumentation Inquiry Learning (GAIL) system. GAIL will provide learners with tools for constructing Toulmin-style arguments in diagrams using blocks of text provided by the system. The text is linked internally to KB elements. An argument generator like one previously developed for another application will use the KB and abstract argumentation schemes to automatically generate expert arguments. GAIL's argument analyzer will determine if a learner's argument is acceptable by comparing it to the expert arguments. A prototype argument analyzer has been implemented using non-domain-specific, non-content-specific rules that recognize common error types. The error types are based on those observed in a pilot study. GAIL's formative feedback generator will use the argument analyzer's output. In addition to identifying problems in the learner's argument, if the argument is acceptable the analyzer will inform the feedback generator of critical questions of the argumentation scheme underlying the student's argument. The critical questions can be used to generate feedback stimulating the learner's critical thinking.

7. ACKNOWLEDGMENTS

Graduate students B. Wyatt and C. Martensen implemented the prototype of GAIL's GUI in summer 2011, and Martensen ran the user study in fall 2011 through spring 2012; both received support from a UNCG Faculty Research Grant. We would like to thank the reviewers as well for asking many interesting questions that we have tried to address in the camera-ready version of this paper or would like to address in future work.

8. REFERENCES

[1] Kirschner et al. 2003. Kirschner, P.A., Buckingham Shum, S.J., and Carr, C.S. (Eds.) 2003. *Visualizing Argumentation.* London: Springer.

[2] Scheuer et al. 2010. Scheuer, O., Loll, F., Pinkwart, N., and McLaren, B.M. 2010. Computer-Supported Argumentation: A Review of the State of the Art. *Computer-Supported Collaborative Learning* 5(1): 43-102.

[3] Pinkwart and McLaren 2012. Pinkwart, N., McLaren, B.M. (Eds.) 2012. *Educational Technologies for Teaching Argumentation Skills*. Sharjah: Bentham Science Publishers

[4] Toulmin, S. 1998. Toulmin, S.E. 1998. *The uses of argument*, Cambridge: Cambridge University Press.

[5] Green, N., R. Dwight, K. Navoraphan, and B. Stadler. 2011. Natural language generation of transparent arguments for lay audiences. *Argument and Computation*, 2(1): 23-50.

[6] Scheuer et al. 2012. Scheuer, O., McLaren, B.M., Loll, F., Pinkwart, N. 2012. Automated Analysis and Feedback Techniques to Support and Teach Argumentation: A Survey. In Pinkwart and McLaren (Eds.) 2012. *Educational Technologies for Teaching Argumentation Skills*.

[7] Green, N. 2005 A Bayesian network coding scheme for annotating biomedical information presented to genetic counseling clients. *Journal of Biomedical Informatics* 38, 130-144.

[8] Walton et al. 2008. Walton, D., C. Reed, and F. Macagno. 2008. *Argumentation Schemes*, Cambridge: Cambridge University Press.

[9] Druzdzel and Henrion 1993. Druzdzel, M. J., and Henrion, M. 1993. Efficient Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the 11th National Conference on AI*, 548-553. Washington, DC.

[10] Green, N. 2010. Towards intelligent learning environments for scientific argumentation. In *Workshop on Ill-defined Problems and Ill-defined Domains, Intelligent Tutoring Systems 2010* (Pittsburgh, PA).

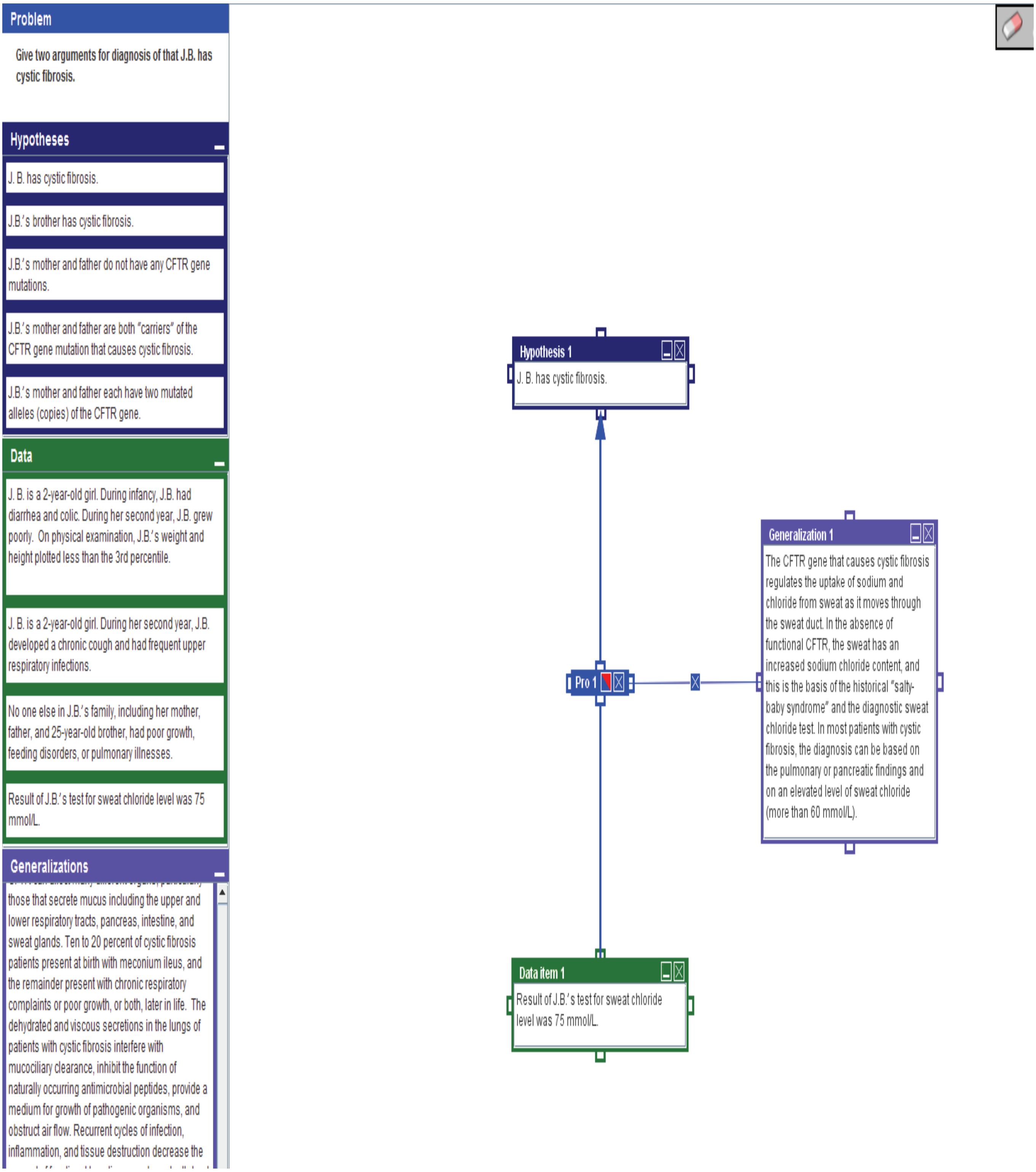

Fig. 1. Screen shot of GAIL prototype user interface in formative evaluation of fall 2011 – spring 2012.