THE IMPACT OF INTEGRATED COACHING AND COLLABORATION WITHIN AN INQUIRY LEARNING ENVIRONMENT

A Dissertation Presented

by

TOBY DRAGON

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2013

School of Computer Science

THE IMPACT OF INTEGRATED COACHING AND COLLABORATION WITHIN AN INQUIRY LEARNING ENVIRONMENT

A Dissertation Presented

by

TOBY DRAGON

Approved as to style and content by:

____

Beverly Woolf, Chair

____

Robert Moll, Member

____

Rick Adrion, Member

____

Tom Murray, Member

____

Merle Bruno, Member

____

Lori Clarke, Chair
School of Computer Science

DEDICATION

To my selfless parents, Don and Kathy, who gave me the ability to do this, to my loving sister, AnnaLee, who gave me the belief I could do this, and to my indescribably perfect wife, Suellen, who gives me the reason to do this.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Beverly Woolf, for giving me both the guidance to complete such an endeavor, as well as the freedom to do so in my own way. She has been a constant source of both inspiration and reassurance, without which my life and work would have not been the same.

Next, I would like to thank my committee members, all of which contributed significantly in their own ways to both my learning process and this document. I thank each of them for the time and dedication that they have offered as educators and colleagues.

I would also like to thank all of my fellow researchers, programmers, teachers, and other classroom facilitators who have provided their time and effort in common cause. An effort this large could never be undertaken alone, and I am grateful for all those who shared in this work. Additionally, I would like to thank the funding agencies that made my research possible, particularly the National Science Foundation.

Lastly, I would like to thank my friends and family, with special thanks to my statistics advisor, morale booster, and general counsel, Joe Smith. This work speaks as much to those around me as it does to my own achievements.

ABSTRACT

THE IMPACT OF INTEGRATED COACHING AND COLLABORATION WITHIN AN INQUIRY LEARNING ENVIRONMENT

MAY 2013

TOBY DRAGON, B.A., ITHACA COLLEGE

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Beverly Woolf

Over the last fifty years, technological advances have brought computer systems into the classroom and motivated a re-consideration of teaching practices and educational methods. Innovative computer systems today go far beyond providing standard instructional material. These *Intelligent Tutoring Systems* (ITS) employ *Artificial Intelligence* (AI) techniques to adapt to students and provide pedagogical support, taking some burden off of teachers and supporting more learner-centered classrooms. The same technological advances also enable more complex, interactive pedagogy. ITSs can be designed to support progressive learning techniques such as *inquiry learning* and *collaboration*, focusing on higher-level learning skills necessary for the 21st century. To offer realistic and complex problems where these higher-level skills can be practiced, these systems often present students with *ill-defined problem spaces*, where problems have neither one correct solution nor one correct solution path.

This thesis explores the design and evaluation of a collaborative, inquiry learning ITS for ill-defined domains. We consider the common ground in the fields of *Artificial*

Intelligence in Education (AIED) and *Computer-Supported Collaborative Learning* (CSCL) to investigate the ways in which an inquiry-based tutoring system can employ both automated coaching techniques and collaborative techniques to support students as they learn in ill-defined problem spaces. We describe our design considerations and the resulting system, a collaborative ITS called Rashi. The Rashi system offers feedback on student work by using an *Expert Knowledge Base* (EKB) to recognize students' solutions and collaborative contributions.

We have used the Rashi system to evaluate the effects of coaching and collaboration on students' behavior, and have investigated the potential to combine these tactics. We find that collaboration significantly improves students' contributions. Students with access to collaborative tools create larger and more complex solutions. The effects of coaching were not as clear. There was no significant effect of allowing access to coaching capabilities, but some evidence suggests that there is a positive correlation between the amount of coaching received and certain metrics that represent positive inquiry behavior. We highlight the potential for combining coaching and collaboration by demonstrating that opportunities can be identified automatically where 1) collaborative work can create more opportunity to provide automated coaching and 2) automated coaching can identify key moments when collaboration should be encouraged.

Finally, we present evidence of the importance of a clear and well-presented classroom pedagogy when introducing an ITS that relies on pedagogy with which students are unfamiliar. We found that giving students a more thorough introduction to the pedagogical approach resulted in significant improvement in students' solutions.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS v

ABSTRACT vi

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTER

1. INTRODUCTION 1

1.1 Motivation 2
1.1.1 Ill-Defined Problem Spaces 3
1.1.2 Inquiry Learning 4
1.1.3 Collaboration 5
1.1.4 System Intelligence 6
1.2 The Problem and The Approach 9
1.3 Research Goals and Hypotheses 12
1.4 Overview of the Dissertation 16

2. PEDAGOGICAL THEORY – INQUIRY LEARNING AND COLLABORATION IN ILL-DEFINED PROBLEM SPACES 17

2.1 Introduction 17
2.2 Ill-Defined Problem Spaces 17
2.2.1 Theory - Ill-Defined Problem Spaces 18
2.2.2 Implementation - Ill-Defined Problem Space of Medical Diagnosis in Biology 24
2.3 Inquiry Learning 26
2.3.1 Theory - The Inquiry Learning Process 27
2.3.2 Implementation - An Inquiry Learning Environment 34
2.3.2.1 Data Collection Tools 34
2.3.2.1.1 Case Description 35
2.3.2.1.2 Interview Environment 36
2.3.2.1.3 Physical Exam Environment – Image Explorer 37
2.3.2.1.4 Medical Lab Environment – The Lookup Tool 38
2.3.2.1.5 Concept Library 39
2.3.2.2 Critical Thinking / Cognitive Tools 40
2.3.2.2.1 The Notebook 41

2.3.2.2.2 The Notebook - Data Table ... 42
2.3.2.2.3 The Notebook - Argument Editor ... 43
2.3.2.2.4 Source Editor ... 44
2.3.2.2.5 Report Editor ... 45
2.4 Collaborative Inquiry Learning ... 46
2.4.1 Theory – Collaboration Supporting Inquiry in Ill-defined Problem Spaces ... 46
2.4.2 Implementation - Collaborative Tools in an Inquiry Learning Environment ... 51
2.4.2.1 The Chat Tool ... 51
2.4.2.2 The Collaborative Notebook ... 53
2.4.2.3 Content-focused Collaborative Features ... 54
2.5 Summary ... 57

3. SYSTEM INTELLIGENCE – RECOGNIZING STUDENT CONTRIBUTIONS AND PROVIDING FEEDBACK ... 60

3.1 Introduction ... 60
3.2 Understanding Student Work ... 60
3.2.1 Recognizing Structure of Student Solutions ... 63
3.2.1.1 Implementing Structural Recognition of Student Solutions ... 64
3.2.2 Recognizing Content of Student Solutions ... 66
3.2.2.1 Implementing Knowledge Bases ... 68
3.2.2.2 Using a Knowledge Base for Content Recognition ... 72
3.2.2.3 Implementing Knowledge Base Content Recognition ... 74
3.2.3 Recognizing Content within Collaboration ... 78
3.2.3.1 Implementing Content Recognition in Student Collaboration ... 79
3.3 Providing Support ... 81
3.3.1 Providing Structural Support ... 82
3.3.1.1 Implementing Structural Support ... 82
3.3.2 Providing Content Support ... 85
3.3.2.1 Implementing Content Support ... 86
3.3.3 Providing Process Support ... 88
3.3.4 Providing Support using Uncertain Assessment ... 89
3.3.5 Feedback Display: Interruption and Visualization ... 91
3.3.5.1 Implementing Feedback Display ... 93
3.4 Providing Content-Focused Collaborative Support ... 96
3.4.1 Using Collaborative Contributions as Coaching Opportunities ... 98
3.4.2 Using Coaching as an Opportunity for Collaboration ... 100
3.4.2.1 Implementing Content-Focused Collaborative Support ... 102
3.5 Summary ... 103

4. EVALUATION OF COACHING AND COLLABORATIVE FEATURES IN AN INQUIRY LEARNING ENVIRONMENT 107

4.1 Introduction 107
4.2 Hypotheses 108
4.2.1 Collaboration 108
4.2.2 Coaching 109
4.2.3 Effects of Improvement to Classroom Pedagogy 110
4.2.4 Combining Coaching and Collaboration 111
4.3 Classroom Studies 112
4.3.1 College Introduction to Biology – Fall 2007, 2008 115
4.3.2 Middle-School Summer Camp – Summer 2007 - 2010 117
4.3.3 University – Biology 101, 2009-2010 118
4.3.4 Middle School – Spring 2010 120
4.4 Data Analysis Techniques 121
4.4.1 Quantitative Assessment 121
4.4.2 Qualitative Assessment 122
4.4.2.1 Validation of Matching Algorithm 125
4.5 Data Analysis Results 127
4.5.1 Effects of Adding Collaboration 127
4.5.1.1 College Student Evaluations 129
4.5.1.2 Middle School Evaluations 130
4.5.1.3 Discussion 131
4.5.2 Effects of Adding Coaching Tools 133
4.5.2.1 Across Group Analysis 134
4.5.2.2 Within Group Analysis 134
4.5.2.3 Discussion 136
4.5.3 Effects of Improvement to Classroom Pedagogy 138
4.5.3.1 Across Group Analysis 139
4.5.3.2 Discussion 140
4.5.4 Combining Coaching and Collaboration 141
4.5.4.1 Using Collaborative Contributions as Coaching Opportunities 142
4.5.4.2 Discussion 143
4.5.4.3 Using Coaching as an Opportunity for Collaboration 145
4.5.4.4 Discussion 146
4.6 Summary 147

5. CONCLUSIONS & FUTURE WORK 150

5.1 Introduction – Hypotheses Revisited 150
5.2 Lessons Learned 151
5.2.1 On Classroom Pedagogy 152
5.2.2 On Collaboration 153
5.2.3 On Coaching 156

5.2.4 On Integrating Coaching and Collaboration 159
5.3 Summary and Future Directions 161
5.3.1 Develop Explicit Classroom Pedagogy... 162
5.3.2 Increase Human Involvement in Coaching 164
5.3.2.1 Implement Wizard of Oz Studies 165
5.3.2.2 Define the Facilitator Role in the Final System 168
5.3.3 Focus on Higher-Order Skills 170
5.4 Summary 171

APPENDIX: GLOSSARY OF TERMS 175

BIBLIOGRAPHY 178

LIST OF TABLES

Page

Table 2.1: Data collection tools and their respective functionality in the medical diagnosis domain. 35

Table 2.2: Cognitive Tools and their respective functionality 41

Table 3.1. Example productive inquiry behaviors as given in (Paolucci et al., 1996), and their matching constraints in the Rashi system. 65

Table 4.1. Detail of the classroom studies. 115

Table 4.2: Quantitative indicators to evaluate student solutions in Rashi 122

Table 4.3: Qualitative indicators to evaluate student solutions in Rashi 123

Table 4.4: Results of the evaluation of the matching algorithm for hypotheses. 126

Table 4.5: Effects of adding collaborative features on data collection and argument creation. Significant effects are observed in each case except for the mean or arguments in the middle-school data. 128

Table 4.6: Results of the automated chat-matching algorithm in correctly identifying the specific content of each student message.. 143

LIST OF FIGURES

Page

Figure 1.1. The Rashi system consists of tools for data collection that can be defined for use in different domains (left), and tools for critical thinking, that are consistent across domains (right). 10

Figure 2.1. The space of ill-defined problems and ill-defined tasks, as presented in Mitrovic & Weerasingh 2009. Moving from well-defined to ill-defined, the top-right quadrant represents well-defined domains with well-defined tasks (WDWT). The bottom-right quadrant represents well-defined domains with ill-defined tasks (WDIT). Finally, the bottom-left quadrant represents ill-defined domains and ill-defined tasks (IDIT). 22

Figure 2.2. The Inquiry Cycle. 29

Figure 2.3. The Case Description provides a brief explanation of the medical case the students will attempt to diagnose. 36

Figure 2.4. The Interview Environment provides a space where the student questions the patient and other persons involved in the case. 37

Figure 2.5. The Exam Environment provides a space where the student can perform a physical exam, which is realized through the Image Explorer. 38

Figure 2.6. The Medical Lab Environment provides a space where student can check results of lab tests, which is realized through the Look-up Tool. 39

Figure 2.7. The Concept Library provides a set of documents in the form of an e-book where students can access information about medical diagnosis. 40

Figure 2.8. The Notebook consists of the Argument Editor (top) and the Data Table (bottom). 43

Figure 2.9. The Source Editor displays a pre-loaded set of external biology resources. 44

Figure 2.10. The Report Editor displays the automated report generated from the Notebook. The report shown here was generated from the Notebook shown in Figure 2.8. 45

Figure 2.11. The Chat Tool provides standard chat functionality but also provides extra features specifically aimed to support content-focused collaboration. Here one student chooses to focus on Hyperthyroidism. 52

Figure 2.12. Comparing Notebooks allows users to compare their work with others. Here, the student Jill has her own Notebook open (left), and is also viewing a classmate's notebook (right). She is then moving data from this other notebook to her own with the drag-and-drop functionality. 54

Figure 2.13. The Critique-Rebuttal feature provides a way for students to dialog about specific Notebook items. Here one student questions a hypothesis and the author of that hypothesis is explains her thinking. 56

Figure 3.1: A student solution in the LARGO ITS, as presented in (Pinkwart et al., 2007) 61

Figure 3.2. A portion of the Expert Knowledge Base for Biology, as seen within the author's view of the Argument Editor of the Rashi Notebook. 70

Figure 3.3. The On-Demand Prompting Coach in the *specification phase*. The system highlights a particular hypothesis, "hyperthyroidism," in the Notebook (top), and asks the student to find more support for this hypothesis in the coaching window (bottom). 94

Figure 3.4: The On-Demand Prompting Coach in the *solution phase*. The coach system opens the Medical Lab environment (top-right), and highlights the supporting information while the coaching window (bottom) tells the student to look at this supporting information. 95

Figure 3.5. The Suggested Links Coach integrated with the Notebook. The user has highlighted "hyperthyroidism" in their Argument Editor (top-left), and the Suggested Links Coach (right) displays knowledge base elements that are associated with this hypothesis. Green highlighted links are supporting data elements, and the user can click these links to be shown the data in a data collection tool. 96

Figure 3.6. The Suggested Links Coach integrated with the Chat tool. One student (Madison) has mentioned "pregnant" in the chat (left), about which the system has much refuting evidence. The Suggested Links Coach (right) displays the knowledge base element about pregnancy, and links to the associated refuting data in red. The student can click these links to be shown the data in a data collection tool. 99

Figure 4.1: Box plots of the argument size (left) and the data collection size (right) for the college-level study. The difference between both groups is significant as judged by the Wilcoxon signed-rank test. 130

Figure 4.2: Box plots of the argument size (left) and the data collection size (right) for the middle-school level study. Argument size did not vary between groups (left), but data collection showed a significant increase (right), as judged by the Wilcoxon signed-rank test. 131

Figure 4.3:No positive effect of the coaching on solution score when comparing control and experimental subsets. Control represents students who had no access to coaching (dataset 8 from Table 4.1). Experimental represents students that had access to coaching (dataset 9 from Table 4.1). 134

Figure 4.4: Linear Regressions with best-fit line showing a significant relationship between Student Support Related Score and coaching (left), and Student Support Score and coaching (right) for dataset 9 from Table 4.1, the experimental group of students who had access to coaching. 136

Figure 4.5: Solution scores as compared between dataset 7, Table 4.1 and dataset 8, Table 4.1. Dataset 7 represents students from the 2009 group having little pedagogical explanation and dataset 8 represents the 2010 group with a clearly defined and promoted classroom pedagogy. The groups are significantly different according to the Wilcoxon signed-rank test. 140

CHAPTER 1

INTRODUCTION

Rapid improvements in technology during the latter part of the 20th century, particularly the advent of personal computer systems, have motivated a re-consideration of teaching practices and educational methods. The first computer systems for education were developed more than fifty years ago (Woolf, 2009). Since then, the idea of using computers in education, or *computer-based instruction* (CBI), has gone from an experimental concept to a reality. A meta-analysis by Kulik and Kulik (1991) showed that these systems are useful and can significantly raise scores and decrease training time.

Most systems used today are static. Even when they provide open-ended experiences, they tend to not be adaptive, nor game-based, nor interactive, and therefore not engaging for students (Woolf, 2009). In a well-designed experimental study, Bloom (1984) demonstrated that one-on-one tutoring (in which a human tutor's responses are individualized for each student) is far more effective than group instruction. However, promoting one-on-one tutoring as educational best practice is problematic, since it is not practical to provide a human teacher for each student.

Woolf (2009) reviewed investigations that focused on developing new methods to employ CBI to offer some of the advantages of individualized tutoring by designing computer systems that understand and adapt to the user. In this way, the computer system can provide some of the adaptive support that individual human tutors provide. Situated within this context, the field of *artificial intelligence in education* (AIED) seeks to provide CBI in which artificial intelligence techniques are used to understand and support students as an individual human tutor might in a one-on-one tutoring scenario. CBI systems that can understand and adapt to the user are termed *intelligent tutoring systems* (ITSs). The community of AIED researchers has been steadily growing in size and scope for over forty years.

1.1 Motivation

A broad range of ITSs exist today, covering many domains, topics, and teaching strategies. Some systems teach well-defined subject matter in traditional ways. Many adaptive systems have been deployed on a fairly large scale and have demonstrated solid evidence of improved learning (Koedinger et al., 2000; Singh et al., 2011). Yet research in the field of AIED also includes a broader concept of teaching and involves multiple teaching strategies that go beyond traditional methods, e.g., collaborative learning and inquiry learning. Tutoring systems that employ these alternative teaching methods potentially provide more genuine, real-life experience and unique forms of user interaction (Baghaei et al., 2007; Kim et al., 2009; Pinkwart et al., 2007; Suthers, 1999). These systems stimulate students by offering hands-on learning experiences that can convey new information and concepts that would be difficult to express with standard systems.

For our research, we consider two major advances in both teaching environments and teaching methodology: instruction in *ill-defined problem spaces* and *collaboration*. Each of these components can enhance the learning process (Gokhale, 1995; Lynch et al., 2009) and each brings unique challenges as well. We consider how each effort is handled individually in the current research field, and how we can combine these approaches in an attempt to offer a complete system that is greater than the sum of its parts.

1.1.1 Ill-Defined Problem Spaces

First, we consider a class of learning environments that explore ill-defined problem spaces. McGraw-Hill Dictionary of Scientific and Technical Terms (2002) defines *problem space* as,

> "*A mental representation of a problem that contains knowledge of the initial state and the goal state of the problem as well as possible intermediate states that must be searched in order to link up the beginning and the end of the task*."

An *ill-defined problem space* is a problem space that does not necessarily have clear start states, goal states, and/or intermediate states, or where the transitions among these states are unclear. An entire domain may be considered ill-defined, in that declarative knowledge about the subject matter is debatable, e.g., legal argumentation, software design, medical diagnosis, art history (Lynch et al., 2009; Mitrovic & Weerasinghe, 2009). Alternatively, tasks within a given domain may be considered ill-defined (Mitrovic & Weerasinghe, 2009), meaning that the procedures involved have disputable aspects, even when the domain is well-defined.

CBI systems that teach within ill-defined problem spaces move beyond simplistic interfaces that allow for multiple choice answer selection or fill-in-the-blank activities. Such systems provide environments where students can explore and apply critical reasoning skills to gather information and reach conclusions. In such environments, there is not one correct path to a solution, or even a specified set of correct solutions. Systems that teach within ill-defined problem spaces typically offer more freedom of exploration and expression than do CBI systems in well-defined problem spaces (Lynch, 2009).

However, these types of learning environments also have liabilities. Particularly, the freedom given to students when working in ill-defined problem spaces creates increased chances of students getting lost or floundering (Kirschner, Sweller & Clarke, 2006; Land, 2000), and therefore generally requires more interaction with instructors to be successful (Suebnukarn & Haddaway, 2006).

1.1.2 Inquiry Learning

To help tackle the challenges offered by instruction in ill-defined problem spaces, we employ a specific pedagogical approach known as the *inquiry learning* method. While there are many ways inquiry learning is defined, the over-arching concept is to provide students with a structure and process that guide them through constructing knowledge while they actively engage in solving realistic problems (Collins & Stevens, 1991; De Jong et al., 2010; Krajcik et al., 1998; Mulholland et al., 2012; Shute & Glaser, 1990; Suebnukarn & Haddawy, 2004). Students engaging successfully in the inquiry process follow certain basic steps: they form hypotheses about the problem at hand, engage in data collection, and relate observable facts collected to support or refute their hypotheses (Ketelhut, 2007; Krajcik et al., 1998). Computer systems to promote or teach inquiry learning are prevalent (De Jong et al., 2010; Ketelhut, 2007, Mulholland et al., 2012; Sabourin et al., 2012; Shute & Glaser, 1990; Suebnukarn & Haddawy, 2004). However, applying inquiry approaches within normal classroom settings offers its own challenges (Clarke et al., 2003) particularly with respect to time and resource limitations. We discuss how ITSs can help alleviate many of these issues and make inquiry learning a viable option for a CBI in ill-defined problem spaces.

1.1.3 Collaboration

Moving beyond traditional ITS approaches that support students to work individually with specific well-defined problems, we consider more open systems that improve learning opportunities by supporting collaboration. Previous research demonstrates that collaboration provides many benefits (Dillenbourg, 1995; Gokhale, 1995; Soller, 2001), but it has been shown that to be effective, collaboration must be applied within specific learning scenarios and conditions (Dillenbourg, 1995, Soller, 2001).

Ill-defined problem spaces using inquiry learning provide an ideal case for collaborative efforts. CBI and ITS systems in ill-defined problem spaces have shown promising results as environments for collaborative work (Baghaei et al., 2007; Constantino-Gonzales et al., 2002). Collaborative components built into CBI generally manifest in two ways. The first is to focus purely on dialog support, where students chat and discuss with one another through the computer (Chaudhuri et al., 2009; McAlister et al., 2004; Morgan et al., 2012). The second form of such systems focuses on shared workspaces, giving students mutual access to learning objects and environments that can be manipulated and discussed (Baghaei et al., 2007; Constantino-Gonzales & Suthers, 2003; Stahl, 2009). A subset of these shared tools focuses specifically on supporting argumentation through shared diagrams (Muller & Mizra, 2007; Pinkwart et al., 2007; Suthers, 1999).

Our research combines three distinct threads of research (ill-defined problem spaces, inquiry learning, and collaboration) and results in a collaborative inquiry learning environment for ill-defined problem spaces. The primary issue addressed here is the divide between data collection/simulation tools and organization/communication tools. The division between content work and collaborative contributions can result in collaborative difficulties and decreased student learning (De Jong et al., 2010; Muhlpfordt & Wessner, 2009). The separation between investigative activities and collaborative activities make it difficult for students to focus their collaboration around specific content at hand. While Muhlpfordt & Wessner, (2009) and De Jong et al., (2010) have offered tentative solutions, this problem of content/communication is still very much an open research area.

One solution to this problem is *content-focused collaboration,* a manner of focusing collaboration specifically on and around the domain content at hand. Anjewierden et al., (2011) have shown that such content focus in discussion is correlated with learning gains. Systems can support content-focused collaboration by integrating features that promote the inclusion of domain knowledge in students' collaborative efforts through interface design. In addition, *artificial intelligence* (AI) techniques can use domain knowledge to recognize when collaboration could be most fruitful, and can then promote targeted collaboration.

1.1.4 System Intelligence

In order for an ITS to reason about student learning, the intelligent portion of the ITS system must understand student work and intervene or adapt material appropriately to successfully support or enhance students' learning processes (Corbett, et al., 1997). This intelligent support is particularly important in ill-defined problem spaces, due to the increased likelihood that a student might require prompting or other intervention if they become lost or stuck to the point that they lose their motivation to pursue the problem (Kirschner et al., 2006; Land, 2000). But to promote inquiry when students work on ill-defined problems, it is important not to intervene too early before students have had a chance to deeply engage in the problem. The great challenge is to identify when students require intervention and what kind of support will promote inquiry rather than shut it down (Mavrikis et al., 2012). Systems that consider only the structure of the students' learning artifacts provide shallow analysis of student learning (Pinkwart et al., 2007; Scheuer et al., 2009). This type of analysis provides only a limited understanding of student work because it accounts for only structure and not the actual content of student contributions.

An approach to understanding student work on a deeper content level is to use an expert system (Baghaei et al., 2007; Pinkwart et al, 2007). Such systems may employ a knowledge base to encapsulate domain knowledge and provide analysis of student work based on a comparison with this knowledge base (Constantino-Gonzales et al., 2002; Crowley & Medvedeva, 2006; Kabanza et al., 2006; Kazi et al., 2009). Researchers must make connections between students' input and the expert knowledge base, which in turn places constraints on the method of student input. The spectrum of student input ranges from *free input*, (e.g., unrestricted user input where students enter any text) to *restricted input* (e.g., limiting user input to specific words, numbers, etc.). Similar input considerations must be made when adding collaborative features to a system. The designers must decide how that collaboration will be carried out and how the content of students' collaboration can be understood by the analysis component of the system. Free input supports more student expression but limits the amount of information the system can understand. Restricted input simplifies the process of content recognition and also scaffolds the learning process (McAlister et al., 2004), yet it is more restrictive for the students and can narrow their contributions (Constantino-Gonzales et al., 2003).

Once an analysis system can recognize and assess student input using an expert system, the designers must choose the type of support to provide. This support can focus on structure (Pinkwart et al., 2007; Scheuer et al., 2009) or content of student solutions (Constantino-Gonzales et al., 2002; Crowley & Medvedeva, 2006; Kabanza et al., 2006; Kazi et al., 2009) or even the students' learning or collaboration processes (Baghaei et al., 2007). Finally, the designer must also consider when this support should be provided and how it should be visualized (Mavrikis et al., 2012). All these considerations play important roles in the final product and will have an effect on how the system is used and how students will learn from it.

Methods of assessing and supporting students in collaborative inquiry systems for ill-defined problem spaces form the foundation for the research carried out for this thesis. The main question is how collaboration can be combined with and/or utilized by a coaching system to improve student learning. The central goals of this research are to discover the impact of collaboration and coaching techniques on student behavior in a tutoring system for ill-defined problem spaces, and to analyze how collaboration and intelligent tutoring systems may be combined to improve student learning.

1.2 The Problem and The Approach

The specific Intelligent Tutoring System on which the current research is based is *Rashi*, an intelligent, collaborative, domain-independent, inquiry learning environment for instruction in ill-defined problem spaces (Dragon et al., 2006; Dragon, Woolf & Murray, 2009). The specific ill-defined problem space on which this research focuses is the use of differential diagnosis in diagnostic medicine. The Rashi system implements both inquiry learning and collaborative approaches, and it recognizes student input through intelligent analysis and provides feedback.

Rashi is domain independent, meaning it is not tied to any given subject matter. However, the system is designed to include domain knowledge that supports specific ill-defined problems. It is built to enable authors to construct knowledge of many different domains in the same framework. Different domains are created through the joint effort of *subject matter experts* (SME) and knowledge engineers by using an authoring tool that supports standard computer users, and doesn't require programming skills. The authoring tool allows these standard users to define environments for students to explore and to define the expert knowledge base used to recognize student work and provide support (Murray et al., 2004). Once these environments and this knowledge base have been defined for a given domain, students using that domain are presented with the environment and provided organizational tools that support and promote the inquiry process (see Figure 1.1). Students explore, collect data, form hypotheses, and try to use the data to support or refute their hypotheses.

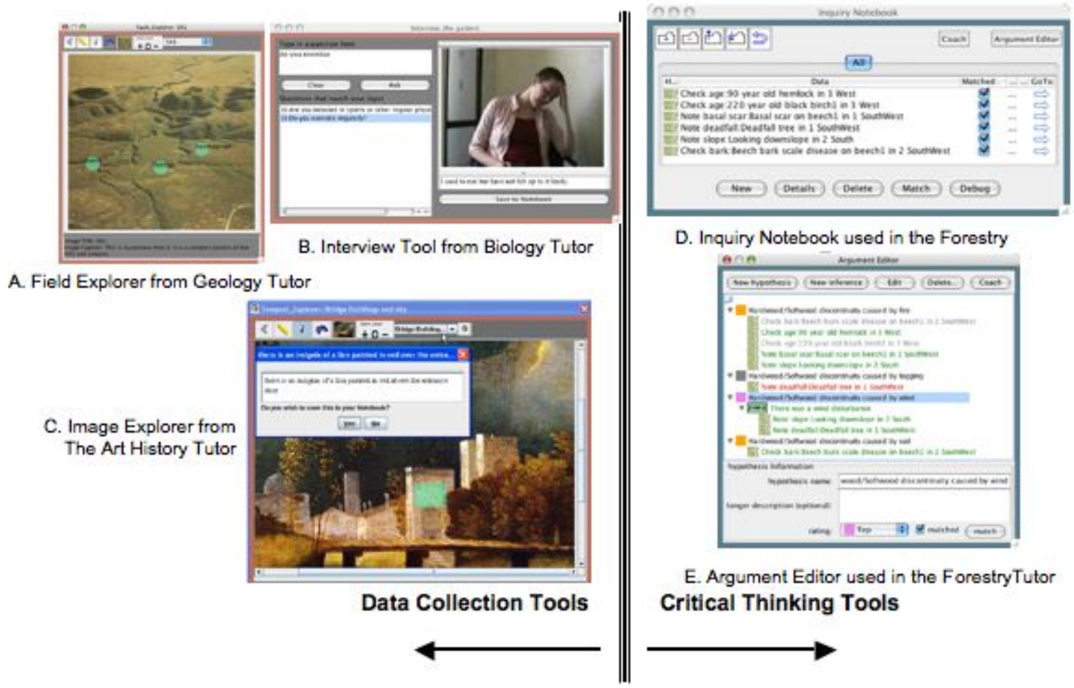

Figure 2.1. The Rashi system consists of tools for data collection that can be defined for use in different domains (left), and tools for critical thinking, that are consistent across domains (right)**.**

Rashi provides a learning environment for many problem spaces and has been used so far in disciplines of biology, forestry, geology, and art history (Dragon & Woolf, 2007). This project focuses on biology, specifically the Rashi implementation of differential medical diagnosis. In this domain, students are presented with the challenge of diagnosing a virtual patient who presents symptoms or complaints that suggest possible ill health. Students form multiple diagnoses (hypotheses about the patient's illness), collect data (interview the patient, examine the patient, request results of medical tests), and use this data to support or refute each proposed diagnosis.

This task demonstrates the concept of an ill-defined problem space, as the process of diagnosing an illness is not algorithmic, even for experts, and experts can disagree on correct diagnoses in many cases (Gauthier et al., 2007). There is a vast solution space to explore (all potential diagnoses) and many paths can be constructed to any given solution (various kinds of data to collect in an unspecified order).

The Rashi system was built initially for individual users and had no intelligent coaching component. The system has been extended to offer coaching abilities (Dragon et al., 2006) as well as collaborative features that support students' interactions with one another (Dragon et al., 2009).

The coaching system was designed and implemented to analyze student solutions and offer individualized feedback at three levels:

- Structural Support – feedback based purely on the structure of student solutions (without understanding the text input or domain knowledge).
- Content Support –feedback based on the content of student solutions (through text-based analysis of hypotheses and data in comparison with the expert knowledge base).
- Process Support – feedback to guide students on potential step-by-step paths through inquiry process.

Many computational tasks are associated with the creation of these three modes of feedback. Developers create an expert knowledge base structure and rules to analyze and assess student solutions compared to the knowledge base. They also define when and how the system will provide feedback based on this assessment.

Rashi has been extended to include several collaborative features along with the coaching capabilities. For basic collaborative capabilities, the system includes both means for students to engage in dialog and means for groups of students to examine each others' information and analytical approaches through shared workspaces. New features of Rashi have been integrated to foster content-focused collaboration. A critique/rebuttal system brings students' comments into context, while subject tagging in the chat window allows students to reference content that is not directly present in the conversation.

The synergy of coaching and collaboration is a final focus of this research to better understand how coaching can be used to further collaborative efforts and how collaboration can improve coaching capabilities. Of particular interest is how this combination can be used to promote content-focused collaboration. In this manner, coaching capabilities can be used to improve the impact of collaborative work on student learning in a specific content area, and collaboration tools can be used to promote more frequent or more meaningful use of coaching mechanisms.

1.3 Research Goals and Hypotheses

The coaching and collaborative features described above were added to the Rashi system with the intention to specifically test the primary research question:

Does targeted use of both collaborative features and feedback from coaching software improve students' solutions?

Studies were conducted in actual classrooms over several years to evaluate the impact of coaching and collaboration on students' solutions. Use patterns and other metrics of student effort were compared across similar populations of students using the system with and without these added features. The performance and behavior of student populations were compared to one another, and the effects of these added features were examined. For example, students who had access to coaching feedback were compared with students who had no access to feedback. Student populations who received coaching were further tested to look for possible impacts of an increased amount of coaching on students' performances within the learning environment. Four specific hypotheses were investigated:

H1: The addition of collaborative features improves student inquiry behavior, increasing the size and complexity of student arguments.

Results support this hypothesis, demonstrating that the addition of collaborative features led to creation of more hypotheses, collection of more data, and the establishment of more relations between data and hypotheses.

H2: The addition of coaching components improves student behavior by helping students focus on essential information and increasing the creation of semantically meaningful and content-rich student solutions.

No statistical support was found for this hypothesis in an across-group analysis of students who had the coach available versus students without the coach available. Some statistically significant correlation was seen between the amount of coaching and improvements in particular aspects of student solutions related to coaching, demonstrating some positive effect of coaching. Overall, the data thus far don't refute the value of coaching, and analysis of this aspect of the study will present theories as to why coaching did not have a more pronounced effect.

H3: Clarification of the pedagogical approach with both facilitators and students improves student behavior, increasing the creation of semantically meaningful and content-rich student solutions.

Results support this hypothesis. A major problem identified in the pilot studies was students' lack of understanding of inquiry and the inquiry process. To address this issue, a clearer introduction of the classroom pedagogy of Rashi was presented to the classroom facilitators. *Facilitators* included both teachers and teaching assistants, who acted in similar roles in the classroom scenarios: giving general instructions, organizing students, and offering support when requested. After facilitators were provided with this additional information about the pedagogical approach in the classroom, they explained the concepts to their students. This pedagogical guidance drastically altered student behavior, demonstrating that a more well-defined classroom pedagogy enhanced the technical approach. Proposals about how this result may be applied to other system improvements and create more pronounced learning effects are discussed.

H4: An expert knowledge-based recognition system can identify opportunities to promote targeted content-focused collaboration.

This concept and feature set was not fully developed within the timeframe of this research, but preliminary evidence suggests this hypothesis is worth further exploration. Some of the evidence supports the hypothesis that collaboration provides a means by which additional domain content can be recognized and that students can receive additional coaching information as a direct result of their collaboration. Additionally, assessment of student solutions demonstrates that the system can recognize opportunities to promote content-focused collaboration.

Most of the evidence presented suggests the promise of both coaching and collaboration to improve students' solutions and students' learning within the Rashi system. More generally, there is great promise in combining coaching and collaboration in ill-defined problem spaces. Three main targets are suggested for future work: develop explicit classroom pedagogy; increase human involvement in coaching; and focus on higher-order skill assessment and development.

Explicit pedagogical models are necessary to guide students to proper use of advanced technology, especially inquiry systems, and effective inquiry work in general. Involving more human effort in the introduction of coaching capabilities allows for a more incremental and thoroughly tested development process and also provides means for better facilitator involvement in the finished system.

Future assessment and system enhancements should focus on higher-order skills. These types of skills gain importance as society places more emphasis on 21st Century problem solving, scientific reasoning, and collaboration (Kellner 2002; Rotherham & Willingham, 2010). Collaborative inquiry learning in general, and the Rashi system in particular, offer a promising test-bed for understanding and teaching these higher-order skills.

1.4 Overview of the Dissertation

The remainder of this document is presented as follows: Chapter 2 presents the educational concepts of the system, describing the basis for the thesis research in both inquiry learning and collaborative learning. This chapter also presents the specific implementation choices for the Rashi system with respect to these two learning approaches.

Chapter 3 presents the artificial intelligence techniques and approaches used by the Rashi system to analyze student work and offer several types of feedback. This includes a description of the expert knowledge base, the system that matches student work to this expert knowledge base, and the methods applied to offer feedback about both content and process.

Chapter 4 presents the evaluation of improvements to the Rashi system, including coaching collaboration, and improvements to the classroom pedagogy. This includes a description of different studies that were run with students in classrooms using the software, the state of the software at the time it was used, and finally a description of the results in regards to the hypotheses of this dissertation.

Chapter 5 presents the main conclusions of this research including the implications of the results, lessons learned that might benefit other investigators, and suggestions for future efforts that are promising based on results of these studies.

CHAPTER 2

PEDAGOGICAL THEORY – INQUIRY LEARNING AND COLLABORATION IN ILL-DEFINED PROBLEM SPACES

2.1 Introduction

Three major pedagogical frameworks have been combined in this dissertation research. First is the concept of and motivation for using computer tutoring systems in *ill-defined problem spaces*, section 2.2. The chosen subject matter, differential medical diagnosis is used as a specific instantiation of an ill-defined problem space, which offers inherent challenges to standard teaching practices. The next two sections discuss pedagogical approaches that offer support and structure when working within such problem spaces. *Inquiry learning* provides a workflow and a methodology to help students operate within ill-defined problem spaces, as presented in section 2.3. *Collaboration*, if invoked properly, has the potential to harness a student's individual abilities to support and promote shared ideas, creating an overall product and understanding that is more productive than individual work, as presented in section 2.4.

2.2 Ill-Defined Problem Spaces

Intelligent tutoring systems (ITSs) have been demonstrated to promote learning in well-defined domains such as K-12 math (Koedinger et al., 2000; Singh et al., 2011). However, it is still an open question as to how systems with artificial intelligence can support learning in ill-defined problem spaces, for which there is no single correct solution, nor single correct solution path. Such problem spaces differ in both content and learning approach. They require a more robust and dynamic pedagogical approach where learners seek multiple solution paths, and opportunistically explore the subject matter at hand (Dragon et al., 2006; Mitrovic & Weerasinghe, 2009). These problem spaces generally do not rely on memorization of domain-specific facts or processes (Lynch et al., 2009). Tutoring in such problem spaces is not a matter of simply applying current definitions that are appropriate in well-defined domains. These concepts require different approaches, both in terms of knowledge representation and teaching strategy.

The growing body of research around ill-defined problem spaces is reviewed, addressing in particular known challenges to teaching this type of subject matter and current computer-based approaches to instruction within ill-defined problem spaces. Finally, the specific focus on differential medical diagnosis is examined to analyze how it exemplifies the concept of an ill-defined problem space and how it offers an opportunity to create an ITS around these types of problems.

2.2.1 Theory - Ill-Defined Problem Spaces

Enumerating the specific qualities that make a problem space "ill-defined" is a curious challenge, since lack of a precise definition is in fact a characteristic of these problem spaces. One of the main artificial intelligence tasks in ITS research is creating a knowledge representation, or domain model (Woolf, 2008). A traditional approach of knowledge representation within artificial intelligence is to "define" that task or knowledge in the domain (Minsky, 1995). As applied to ITSs, this modeling task manifests as a descriptive or prescriptive model of potential student actions and their resulting consequences as in the entire class of knowledge tracing tutors (Corbett &

Anderson, 1994). When constructed in such a way, these knowledge representations are inherently well defined. Viewing knowledge representation in this way assumes that an ill-defined problem space is a portion of the overall space that has not been "mapped out" appropriately into some computer-recognizable language. However, there is growing consensus that some domains and problems are inherently ill-defined (Lynch et al., 2009; Mitrovic & Weerasinghe, 2009). Tutoring in such problem spaces requires more than merely applying techniques appropriate to well-defined domains.

Accepting the idea of inherently ill-defined problem spaces still leaves many potential conflicts and differing terminologies. Lynch et al., (2009) set out functional characteristics of problem spaces that are widely deemed to be "ill-defined" to include areas that,

- lack widely accepted domain theories identifying relevant concepts and functional relations.
- cannot be readily partitioned into independent sub-problems.
- have prior cases that are partially inconsistent.
- involve the need to reason analogically with cases and examples.
- have a large or complex solution space that prohibits one from enumerating all possible characterizations or solutions.
- lack formal or well-accepted methods to verify solutions.
- lack clear criteria by which solutions are judged.
- are not considered to be "solved" when one solution is presented but may be readdressed by multiple, often distinct, solutions.

- involve disagreements among domain experts regarding the adequacy of the solutions.
- require solvers to justify their solutions through argument.

These characteristics clarify both the need for providing instruction in such problem spaces and the difficulties of representing such problem spaces using traditional methods. The problem spaces defined by this list are not fringe areas of study; they are the very heart of certain subjects such as law, ethics, architecture, art history, and medical sciences (Dragon & Woolf, 2007; Gauthier et al., 2007; Lynch et al., 2009; Pinkwart et al., 2007; Mitrovic & Weerasinghe, 2009). For ITSs to offer solutions across the entire spectrum of educational challenges, ITSs for ill-defined problem spaces must be addressed. However, we also see that providing instruction and support for problems that have solutions space that prohibits enumerating all possible solutions or problems that are not considered solved when one solution is presented requires some type of pedagogical approach and knowledge representation beyond traditional means such as knowledge tracing or model tracing tutors. These approaches attempt to map out all student actions and recognize the entire solution space in reference to their models (Anderson et al., 1990; Corbett & Anderson, 1994), which is an inadequate approach to ill-defined problem spaces.

Many innovative teaching strategies lie entirely within, or are improved by, operating in ill-defined problem spaces. Examples include any technique promoting free-thinking exercises or creativity (Wegerif, 2010). When creating ITSs that use these strategies, the system should impose less structure to support more freedom for students. These systems need to enable and understand a larger set of acceptable solutions and solution paths. This flexibility provides freedom for the student to create unique and equally correct solutions from their own understanding rather than recording memorized expert solutions (Kazi et al., 2009; Lynch et al., 2009). Systems have been built that understand and support student work in such situations, and provide instruction over vast solution spaces. One particularly successful approach is termed *constraint-based tutoring* (Mitrovicet al., 2007), where solutions are judged by a set of constraints to create classes of solutions rather than enumerating individual solutions.

To further define the concept of ill-defined problem spaces, one must consider whether the task is ill-defined, the domain is ill-defined, or both. Mitrovic and Weersainghe (2009) describe the operational difference between task and domain as the differentiation of declarative knowledge and procedural knowledge. Ill-defined domains are entire domains in which the declarative knowledge can be debated—even the theoretical general knowledge about the given domain. Ill-defined tasks are tasks in which the procedural knowledge is unclear: the start state, transformations, and end states are unclear or innumerable, and therefore pre-set solutions or algorithms cannot be offered to satisfy all possibilities. With this definition, we can see that ill-defined domains are clearly tied to ill-defined tasks, yet ill-defined tasks are not necessarily related only to ill-defined domains, and in fact might be associated with well-defined domains. Figure 2.1 shows these different spaces, and places examples of subject matter in their respective locations in this task/domain mapping space. We view this given figure as a mapping of problem space, and we use the term ill-defined problem space to reference any location below the center in this space, meaning it involves some level of ill-defined task regardless of whether or not the domain is ill-defined.

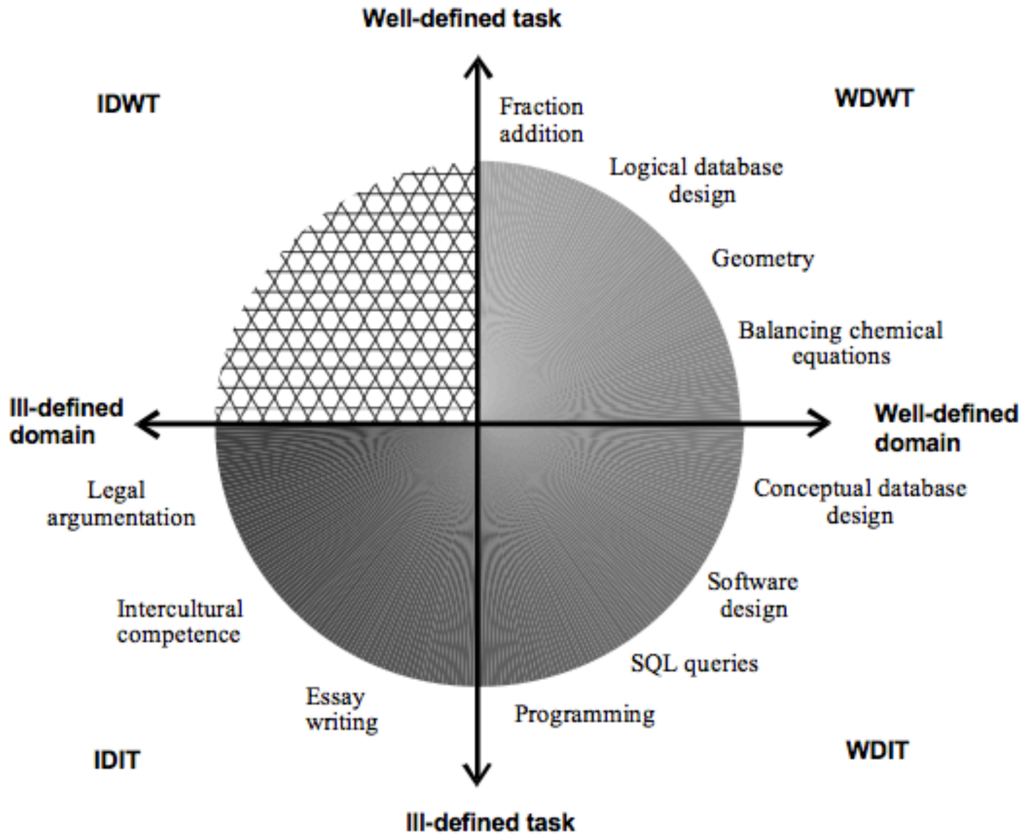

Figure 2.3. The space of ill-defined problems and ill-defined tasks, as presented in Mitrovic & Weerasingh 2009**.** Moving from well-defined to ill-defined, the top-right quadrant represents well-defined domains with well-defined tasks (WDWT). The bottom-right quadrant represents well-defined domains with ill-defined tasks (WDIT). Finally, the bottom-left quadrant represents ill-defined domains and ill-defined tasks (IDIT).

A wide variety of related work has been undertaken in the space of ill-defined tasks, as seen by the examples in Figure 2.1. Some notable examples include legal argumentation (Pinkwart et al., 2007), intercultural competence (Kim et al., 2009), and database design (Baghaei et al., 2007).

One example of such a tutoring system focuses on intercultural competence necessary for negotiation. The Bi-Lateral Negiotiation (BiLAT) system attempts to teach negotiation skills combined with cultural awareness and competence, demonstrating modest learning gains for novices and more experienced users (Kim et al., 2009). Students engage in conflict negotiation in a culturally specific scenario, offering a clear example of both an ill-defined domain and task. Specifically, students are immersed in a virtual environment and take on a role of a negotiator as they attempt to reach agreement with virtual agents. Through these interactions, the students learn the importance of cultural differences and learn to interact in the simulated culture in a more productive fashion.

Legal argumentation also falls into this category of ill-defined domains and tasks. Legal Argument Graph Observer (LARGO) offers students practice in creating argument diagrams of legal arguments to parse how the law has been, or should be. applied to specific cases at hand (Pinkwart et al., 2007). This domain is inherently ill-defined, as there is not overall agreement about correct and incorrect application of the law (there is continual debate and re-interpretation). Also, the task is ill-defined partially because students do not just apply law, but rather their task is to reason about the laws themselves (Pinkwart et al., 2007). This leaves a broad solutions space where assessment of results requires deep analysis and interpretation.

Moving towards the space of well-defined domains, we consider the COLLECT-UML system designed to tackle software design issues, specifically UML diagramming (Baghaei et al., 2007). While the underlying concept of Entity-Relationship models is well-defined, the task in this given system is ill-defined both in the loose specifications given as a start state, and the loose definition of appropriate goals states (Mitrovic et al., 2009). The COLLECT-UML system offers students practice in these tasks by providing a collaborative workspace where students can create different UML diagrams and attempt to solve the different challenges (Baghaei et al., 2007).

Each of these systems comes with a unique set of challenges and tools necessary for practice, but there are also common issues that are characteristic of work in ill-defined problem spaces. Tutoring systems for ill-defined problem spaces must allow the user freedom to explore the problem space, and accommodate personalized and unpredictable solutions and solution paths (Lynch et al., 2009). Each given system offers its own solutions to these problems, but all systems must confront them to be successful.

2.2.2 Implementation - Ill-Defined Problem Space of Medical Diagnosis in Biology

The specific ill-defined problem space that is the focus of this research is differential diagnosis in human biology. This domain fits almost all the characteristics given by Lynch et al., (2009) enumerated above that categorize an ill-defined problem space. For example, during diagnosis, domain experts often reason analogically with cases and examples, disagree regarding the adequacy of the solutions, and lack formal or well-accepted methods to verify solutions. This is largely due to the vast and ambiguously defined solution space, the lack of available methods by which to judge solutions, and the existence of multiple, distinct solutions.

In reference to Figure 2.1, differential medical diagnosis is an ill-defined task situated within the border between an ill-defined/well-defined domain. The domain of differential medical diagnosis has a certain level of structure; much settled declarative knowledge exists on the subject (Beers et al., 2006; Dorland's illustrated Medical, 2011). These sources enumerate existing diseases, and known information about their likelihood, their symptoms, and their inter-relations. However, the domain is not well-defined in that the given declarative knowledge about disease is ever changing; new diseases are discovered and new symptoms added to the collective knowledge. Even as the domain is somewhat defined, the task given to a person who is diagnosing a patient is clearly ill-defined. Practitioners are presented with patients exhibiting an initial set of symptoms. These practitioners must take steps to decide the patient's most likely illness by collecting information from the patient. There are many approaches to this problem (Baerheim, 2001), with a wide variety of equally good solutions, and large disagreement between experts as to "best" solutions (Gautheir et al., 2007).

For our study, we will consider a number of these diagnosis tasks, or cases, used as teaching tools to instruct students in biology. Students are presented with these cases through the Rashi system, an ITS designed by our team to support learning in ill-defined domains. The Rashi system presents environments for students to explore the diagnosis case at hand (Section 2.3.2) and also allows students to collaborate while working on these cases (Section 2.4.2). Students interview, examine, and conduct lab tests for their patient, create hypotheses (potential diagnoses), and try to support and refute their hypothesis to offer a set of final diagnoses as to what might be the potential problems with the patient's health. Rashi's specific tool set (Section 2.3.2) allows students to operate in this ill-defined problem space by exploring using data collection tools (Section 2.3.2.1) and creating a wide range of solutions and solution paths with the cognitive tools (Section 2.3.2.2).

Rashi is not the only system that supports students as they practice the ill-defined tasks associated with differential medical diagnosis. Several other notable examples exist, including: BioWorld (Lajoie et al., 2001), TeachMed (Kabanza et al., 2006), and the Collaborative Intelligent Tutoring (COMET) system for Medical Problem Based Learning (Suebnukarn & Haddawy, 2006). Bioworld and TeachMed both present tools similar to the Rashi system, offer case-based problems, and also offer examinination and interviews with simulated patients. The COMET system is at a higher-level in terms of audience and content knowledge, as it is designed for medical school classrooms where students need to quickly diagnose specific medical conditions (such as head trauma).

It is important to note that our main research contribution is not focused on providing a teaching system in this specific domain, but rather to consider this domain as a vehicle within which to test teaching methods and theories. These other systems are often built explicitly for the medical domain and cannot be extended outside of their specific domains. This is true of the majority of collaborative systems that offer domain support (Magnisalis et al., 2011). In contrast, Rashi is domain-independent system and has been authored to function within many domains, including art history, geology, and forestry (Dragon & Woolf, 2007). In this way, our work has less impact on a single domain (e.g. medical diagnosis) but rather a more broad impact on teaching style that may be applied generally across many domains. We only present the work in the medical domain to provide a consistent example and to give the most detail about the domain in which our final studies were conducted.

2.3 Inquiry Learning

Ill-defined problem spaces require teaching methods that rely on more than memorization and that promote higher-order thinking skills (Lynch et al., 2009). Inquiry learning methods support the student as an active participant who does more than merely receive information and answer prompted questions. Rather, the student assumes active roles of collecting information, forming hypotheses, and investigating phenomena in his/her own way. This method of instruction is well documented and has been practiced in numerous disciplines and with different age groups (e.g., Clark et al., 2003; Collins & Stevens; 1991, De Jong et al., 2010; Geier et al., 2008; Krajcik et al., 1998, Mulholland et al., 2012; Shute & Glaser, 1990, White & Frederickson, 1998). The lack of more broadly used inquiry-based learning activities in schools is partially due to the inherent difficulties of bringing such activities into a standard classroom setup (Clarke et al., 2003; Geier et al., 2008). This section addresses the concept of inquiry learning and the challenges that may minimize its use in standard classrooms and proposes how ITSs can offer solutions to those challenges. Rashi is presented as an approach to address the challenges of implementing inquiry learning and to provide a space for students to engage in authentic inquiry learning in the classroom.

2.3.1 Theory - The Inquiry Learning Process

As with ill-defined domains, there are a wide variety of definitions and differing viewpoints about what truly defines inquiry learning. However, common issues about using inquiry learning exist and these issues often create difficulties in the normal classroom setting. This section presents common inquiry themes, common impediments to bringing this type of education to the classroom, and the ways technology might help address those impediments.

While many researchers study inquiry learning in classrooms, there is no agreed upon definition. The high-level concept of inquiry learning involves an activity placed in an open environment, with certain tasks to be attempted in this environment. The environment is usually a space to explore some questions about real-world phenomena or a statement of a real problem faced by workers in a field of study (Collins & Stevens, 1991; De Jong et al., 2010; Krajcik et al., 1998; Mulholland et al., 2012; Shute & Glaser, 1990; Suebnukarn & Haddawy, 2004) Students placed in these environments are often expected to accomplish a number of tasks, including,

1) Create hypotheses or theoretical models to explain phenomena.
2) Gather evidence in some structured way to better understand the phenomena. Relate the evidence to the hypotheses or models to create/demonstrate a theoretical understanding of the situation being investigated.

Different investigations have addressed a wide variety of subject matter for inquiry, including biology, economics, physics, and societal issues (Geier et al., 2008; Krajcik et al, 1998; Muholland et al., 2012; Paoucci et al., 1996; Sabourin et al., 2012; Shute & Glaser, 1990; Toth, et al., 2001; White & Frederickson, 1998). In this research, inquiry is characterized as a cycle (Figure 2.2) in which students move among four phases opportunistically (not necessarily in a specific order):

- Hypothesis formation
- Planning / Questioning
- Data collection
- Analysis

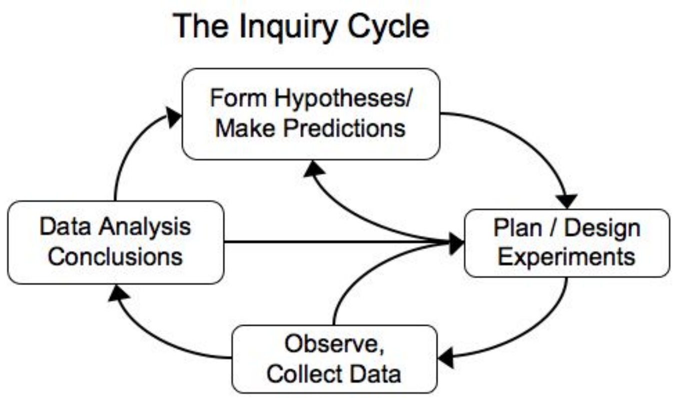

Figure 2.2. The Inquiry Cycle.

Inquiry learning is often grouped with other types of instruction and referenced with different names. The term problem-based learning (PBL) is used in some instances (Suebnukarn & Haddawy, 2004), and indeed many efforts in PBL could be considered also to be taking the inquiry learning approach. Some researchers have attributed the same goals and themes to PBL as they have to inquiry learning, and actually use the terms interchangeably. A strong connection also exists between inquiry learning and discovery learning, yet here a clear distinction can be drawn. While both inquiry and discovery learning are "learning by doing" methods, inquiry is more clearly defined as a focused, systematic search for knowledge (Shute & Glaser, 1990). Pure discovery learning is involves a student constructing his or her own knowledge with little to no guidance (Van Joolingen, 1998). Inquiry systems tend to have a primary goal of promoting systematic and organized methods of drawing conclusions from data available. In fact, searching more blindly or without a specific plan of action in an inquiry learning environment is viewed as undesirable behavior (Shute & Glaser, 1990, Suebnukarn &

Haddawy, 2004). Many educators strike some balance in their inquiry process, allowing room for discovery but also providing guidance and expecting some organization and thoughtful approach to exploration (Van Joolingen, 1998; Sabourin et al., 2012).

Inquiry learning is not broadly used in classrooms (Clarke et al., 2003; Geier et al., 2008). It is considered an alternative style of teaching and diverges from the standard classroom activities of lecture and examination. It is perceived as taking too much classroom time and too much teacher preparation, and it is more difficult to assess student progress than standard approaches. However, many researchers argue that the benefits to students' development of independent and analytical thinking as well as increased motivation outweigh traditional lecture-base teaching (Collins & Stevens, 1991; Shute & Glaser, 1990; White & Frederickson, 1998). In a sound empirical study, Geier et al. (2008) demonstrated the long-term positive learning effects of the inquiry method, showing improved standardized test scores among other measures. This section examines the potential benefits of inquiry learning over more conventional methods of instruction as well as potential issues with bringing inquiry into the classroom, and proposes ways that computer-based programs can help alleviate these issues and make inquiry more practical in a classroom setting.

The inquiry method of instruction, when used in a classroom, takes more time, space, and resources than does than standard lecture and examination teaching methods (Clarke et al., 2003; Geier et al., 2008). The increased instruction time can be caused by time wasted as students get lost, stuck in one line of thinking, or flounder as they try to work through complex problems (Dragon et al., 2006; Sabourin et al., 2012; Van Joolingen, 1998). Thus, for many teachers and administrators it is difficult to justify the use of inquiry learning in mainstream teaching scenarios, and understandable that the techniques are not widely adopted (Geier et al., 2008). During inquiry learning key information within the domain is often not presented to students in pre-defined manner because students are self-directed as they explore and examine phenomena. The instructor must provide much more individual attention to students (Clarke et al., 2003), because each student or team forms his/her own hypotheses and collects his/her own data rather than the entire classroom receiving the same presentation of information. Such individualized feedback and attention from an instructor is one of the most limiting aspects of inquiry learning in classrooms. Often, teachers have classes with thirty or more students, and so it is impractical to use a teaching method that requires one-on-one or small group interaction with students.

However, in situations where inquiry learning is possible, it has been shown to be more successful than typical classroom instruction in the following ways. The first and most obvious benefit of inquiry learning is that it stimulates interest and engagement among students (White & Frederickson, 1998). When students are able to manipulate artifacts themselves and think freely about problems, they are more actively involved in the learning process. This increased activity not only creates interest, but has also been shown to facilitate knowledge and skill acquisition (Geier et al., 2008; Shute & Glaser, 1990; White & Frederickson, 1998).

Also important to consider when weighing the strengths of inquiry learning are the additional meta-level skills learned through such a method. Beyond domain knowledge, inquiry learning also can teach students how to learn and how to learn on one's own. It teaches a process by which one can independently attain and develop explanations for observable phenomena. Certain computer systems are built with the primary goal of teaching these processes, and only secondary is the acquisition of the subject matter being discussed (Shute & Glaser 1990, White & Frederickson, 1998). The theory behind such systems is that by learning the inquiry method, one can learn more easily how to acquire new knowledge (White & Frederickson, 1998).

Given that inquiry teaching approaches have been shown to have benefits beyond passive learning approaches, it is important to consider means to alleviate the barriers to implementing this teaching method in the classroom. Technological advances in CBI and AIED offering promising directions. Current CBI systems can provide a means to experiment through simulations with real-world phenomena, from microeconomics (Shute & Glaser, 1990), to medical diagnosis (Suebnukarn & Haddaway, 2004, Lajoie et al., 2001, Kabanza et al., 2006) to cultural competence (Kim et al., 2009). The increased interactivity alone has been shown to increase learning (White, et al., 1999). CBI can provide a practical space in which students may experiment and observe with minimal time and cost (Geier et al., 2008). This alleviates issues of time/physical space/costly equipment that might be necessary in non-technology-based inquiry approaches to learning.

Second, ITSs offer means to allow teachers to work one-on-one with students while knowing that their other students are engaged and working and getting intelligent support from the ITS. The intelligent components within these systems can take on parts of the instructor's roles in inquiry environments. Students may need examples, strategies to accomplish tasks, help understanding principles, and reminders to return to task when students wander off task (Krajcik et al., 1998). By monitoring student action and providing feedback, intelligent agents in computer systems can provide all these types of support (as described in Chapter 3).

The inquiry method is not typically used in isolation. Traditional methods of lecture are beneficial when interspersed with inquiry learning experiences, especially to present useful domain principles at crucial times in the learning process (Collins & Stevens, 1991). Thus, inquiry learning does not take the place of lecture or textbooks, it incorporates those tools into the process so that students take more active roles in certain parts of the learning process. Inquiry learning helps students think more explicitly about how they are learning. Such is the case with almost all inquiry learning ITSs. These systems are generally used during some small part of a normal class curriculum, and not as a replacement for an entire section of curriculum content.

The Rashi project was one of the early ITSs to adopt the inquiry learning approach, and it has maintained a position at the forefront of research in this area. However, in recent years, several other systems have been developed. Mulholland et al. (2012) describe the development and tools used in nQuire, an ITS that offers a general framework for implementing the inquiry process including user tools to create hypotheses, collect data, and find relationships among data.. Another recent project is the Science Created by You (SCY) project, offering a suite of tools and encouraging students to engage in inquiry about real-life science challenges (De Jong et al., 2010). These tools are still under development and are taking paths and undergoing assessments similar to those taken in developing the Rashi system in terms of collecting and organizing data to enact an inquiry procedure.

2.3.2 Implementation - An Inquiry Learning Environment

For the implementation of an inquiry-learning environment, we will describe in detail the Rashi system, a domain-independent computer-based learning system. This description will focus on the biology domain, specifically differential medical diagnosis (as discussed in section 2.2). Students are asked to play the role of a physician receiving a new patient. After being presented with some introductory material, they are invited to form hypotheses about potential illnesses and collect data to inform their hypotheses. Rashi provides a set of tools to support students in collecting data (Table 2.1), and another set that supports them in thinking critically, organizing their thoughts, forming hypotheses, and establishing relations with the collected data (Table 2.2). We examine each of the tools in these two categories and explain further their role in the inquiry process.

2.3.2.1 Data Collection Tools

The data collection tools provide different *environments* for students to collect information. In this way, students using the system have the ability to explore complex, realistic phenomena (such as the experiences of a doctor's day-to-day work) in a practical way in the classroom, through simulation. These data collection tools provide access to complex and ill-structured knowledge along with some context that helps convey the realistic nature of the content. These tools are built generically, and can be applied in different ways to support different domains. Below we enumerate the different data collection tools (Table 2.1), give a brief description of each tool, and explain its function within the medical diagnosis domain.

Table 2.1: Data collection tools and their respective functionality in the medical diagnosis domain.

Data Collection Tool	Functionality
Case Description	Provides preliminary information about the case a student will attempt to solve.
Interview Environment	Provides a virtual patient to be interviewed about medical signs and symptoms.
Image Explorer – **Physical Exam Environment**	Provides the ability to examine a virtual patient's anatomy to identify physical signs of ailment.
Lookup Tool – **Medical Lab Environment**	Provides results of simulated laboratory tests on the patient as requested by students.
Concept Library	Provides multi-media information about the domain at hand in e-book format.

2.3.2.1.1 Case Description

When a student begins a new case, he/she is presented with the Case Description. This tool provides some overarching description of the particular challenge that the student faces in this instance. The student may highlight any details he/she find important and save the information directly to his/her Notebook (see section 2.3.2.2). In the medical domain, this description states initial demographic information about the patient, the symptoms the patient is presenting, and other introductory material pertinent to the case.

Figure 2.3. The Case Description provides a brief explanation of the medical case the students will attempt to diagnose.

2.3.2.1.2 Interview Environment

The Interview Environment offers a configurable simulated interview with a (low-tech) virtual human. Students are presented with pictures, audio, and/or video of the person they are interviewing. They are free to type in any questions they would like to ask. The system then matches the text entered to a database of questions the virtual human is prepared to answer (as pre-specified by a domain author). Students choose from this list of matching questions and are presented with answers. These questions/answers can be saved to the student's Notebook with one click. This system is used in the medical domain to allow the user to interview the patient, and other persons associated with the current case.

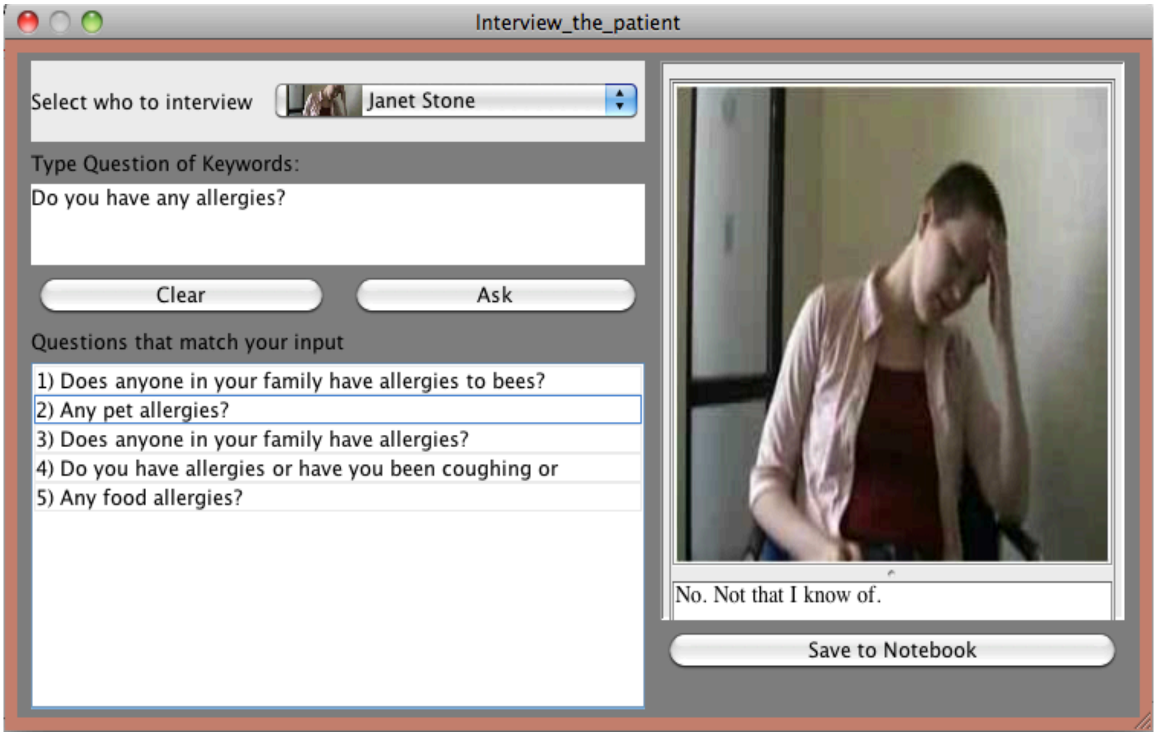

Figure 2.4. The Interview Environment provides a space where the student questions the patient and other persons involved in the case.

2.3.2.1.3 Physical Exam Environment – Image Explorer

The Image Explorer displays images to students. These images include hotspots, which instantly take students to a related link (either some piece of information or another image). This tool can provide many types of simulations, from examinations of paintings for art history to walkthroughs of a forest (Dragon & Woolf, 2007). In the biology domain, this Image Explorer is used to create a physical examination, where students gather information about a patient's physical state. As seen in Figure 2.5, the student is presented with the human form, and hotspots are located throughout this outline that offer information about that particular part of the body (e.g., lymph nodes, ears, etc.). The tools across the bottom represent hotspots that allow access to other information (e.g., blood pressure, weight, etc.).

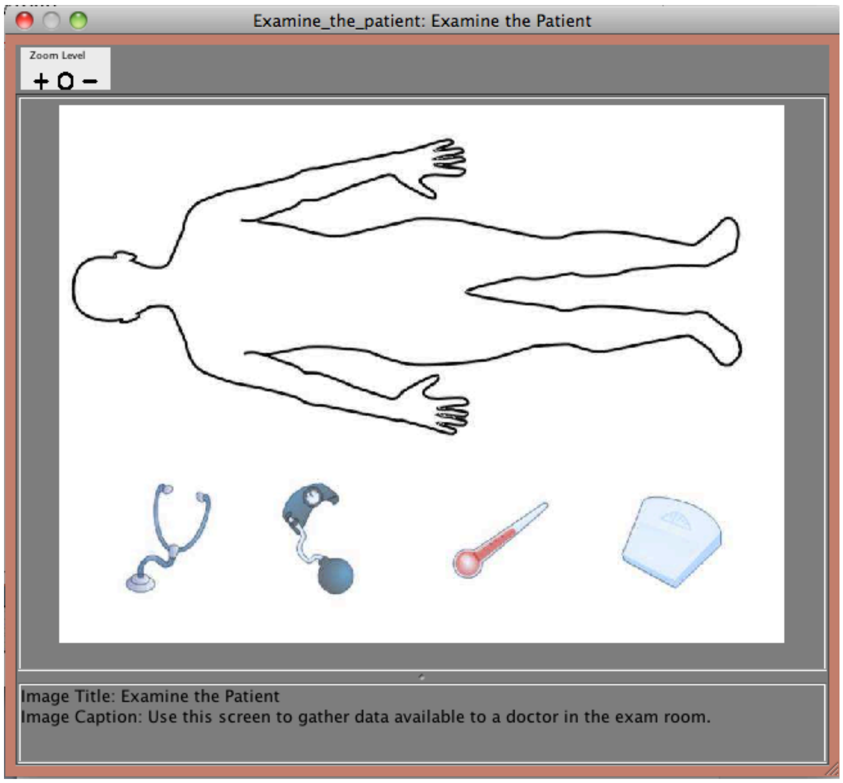

Figure 2.5. The Exam Environment provides a space where the student can perform a physical exam, which is realized through the Image Explorer.

2.3.2.1.4 Medical Lab Environment – The Lookup Tool

The Lookup Tool provides a generic location where key-value pairings can be created by authors and retrieved by students. These values can be any multimedia: images, text, or video. The entries can be categorized and sorted for organization. In the medical domain, this tool manifests as the Medical Lab Environment (Figure 2.6). Students can run different types of lab tests to inform their diagnoses. The system prompts students to provide rationale for each lab test, promoting reflection and self-explanation, which has demonstrated benefits (Aleven & Koedinger, 2002; Conati & Vanlehn, 2000). This explanation system is also meant to dissuade users from "gaming" the system as defined by Baker et al., (2008), as students might run all of the lab tests without reason in hopes of stumbling upon interesting results.

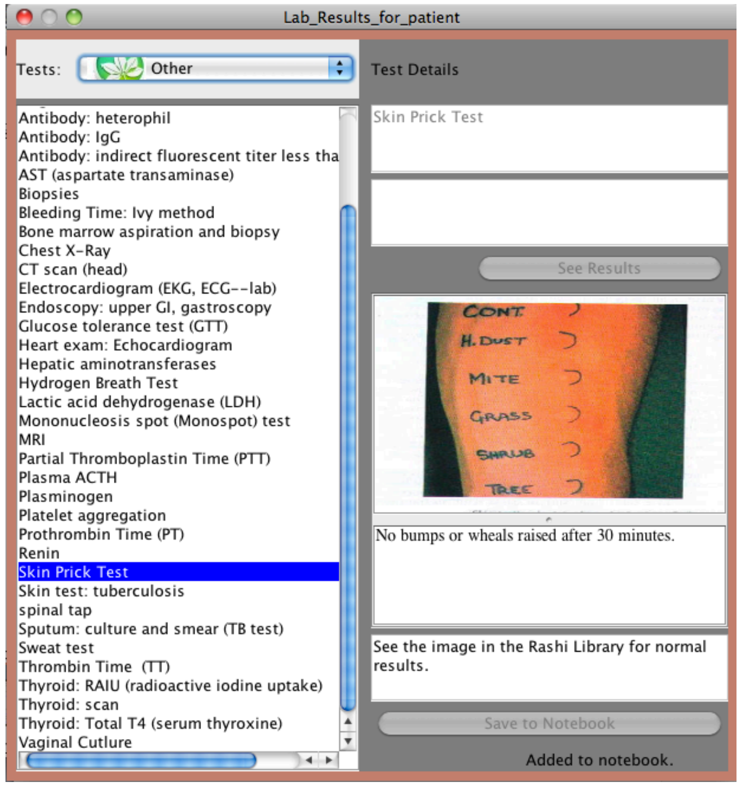

Figure 2.6. The Medical Lab Environment provides a space where student can check results of lab tests, which is realized through the Look-up Tool.

2.3.2.1.5 Concept Library

The Concept Library is a tool that provides an e-book-style resource for any general information provided by the domain author. This could be as as expansive as a full textbook or as limited as some summaries of outside sources or instructions. The content can also be multimedia, containing text, images, and video. Users can highlight text from this book and save it to their Notebook with a single click. In the biology domain, the Concept Library manifests as a repository for medical reference material, general material on the inquiry process, and grading procedures.

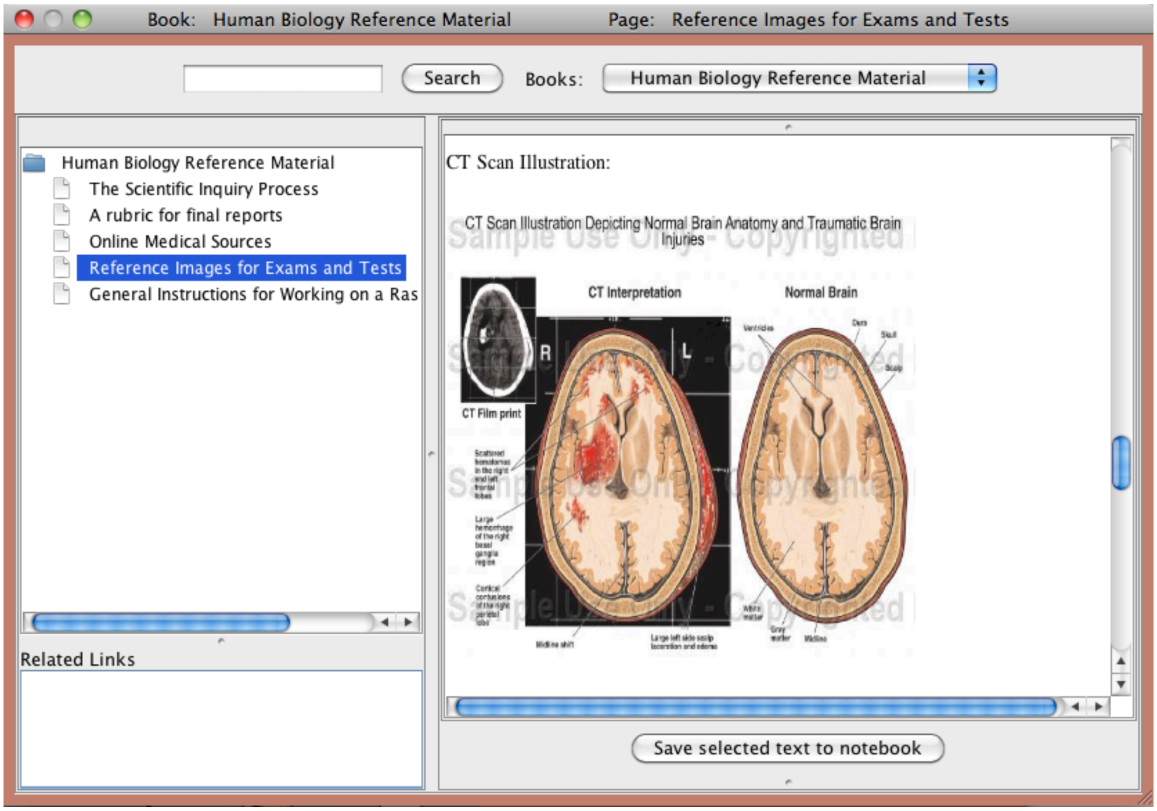

Figure 2.7. The Concept Library provides a set of documents in the form of an e-book where students can access information about medical diagnosis.

2.3.2.2 Critical Thinking / Cognitive Tools

The five data collection tools described in section 2.3.2.1 provide students with the ability to explore the task at hand. Now we consider the tools that help students think critically, organize their thoughts and the data they have collected, cite their sources, and present their findings. Cognitive tools are a vital part of any inquiry system because, by design, they prompt students to identify and explicate the structure of their solutions, a skill central to inquiry learning (Van Joolingen, 1998). These tools help students organize information from within the Rashi system, but also allow students to access and organize information from external sources (e.g., web resources, books, etc.). This is especially important in ill-defined problem spaces, because no system can generally represent all the necessary or useful knowledge in an ill-defined problem space. This also helps students prepare to operate in real-world scenarios by consulting a variety of materials, as there is rarely one definitive source for answers to real-world problems.

The Rashi system provides four main cognitive tools (Table 2.2) designed to explicitly support phases in the inquiry cycle: forming hypotheses (the Argument Editor), collecting data (the Source Editor and Data Table), and analyzing data to reach conclusions (the Argument Editor and Report Editor). We now consider each tool in detail.

Table 2.2: Cognitive Tools and their respective functionality

Cognitive Tool	Functionality
Notebook - Data Table	A repository for the evidence (data) that students gather from data collection tools (i.e., Table 1)
Notebook – Argument Editor	A space used to create hypotheses, and organize data that supports or and refutes a student's hypotheses.
Source Editor	A repository for resources outside of the Rashi system used to track and cite sources of information used during diagnosis.
Report Editor	A space to create a report on the student's work. The system can offer an automated draft created from the work the student has completed.

2.3.2.2.1 The Notebook

The Notebook consists of two distinct tools: the Data Table (Figure 2.8, bottom) and the Argument Editor (Figure 2.8, top). Each provides the student with organizational tools and each supports the inquiry process. These tools are presented in one "notebook" interface because their use is inherently linked, data from the Data Table is constantly used as material to support and refute hypotheses in the Argument Editor. However, the tools are distinct as each serves a unique purpose. We now look to each tool and to their combined functionality to understand how the student engages in the inquiry process through their use.

2.3.2.2.2 The Notebook - Data Table

The Data Table tool provides a central repository for all of the data that students collect while engaged their current case (Figure 2.8, bottom). This includes everything from facts students discover using the Image Explorer to sentences recorded from the Case Description. Students can also type freely into the Data Table to enter ideas or observations that are not recorded from within the system.

This mixed functionality demonstrates the way in which the Rashi system addresses the challenge of allowing freedom in ill-defined problem spaces (as discussed in section 2.2.1). The system supports students by making data collection as streamlined as possible, while still allowing students free input. So they may enter any data that they choose. In this way, students have both the ease of automated collection and the full functionality of an open-ended system necessary for true inquiry behavior in ill-defined problem spaces, where all pertinent information may not be explicitly encoded within the system.

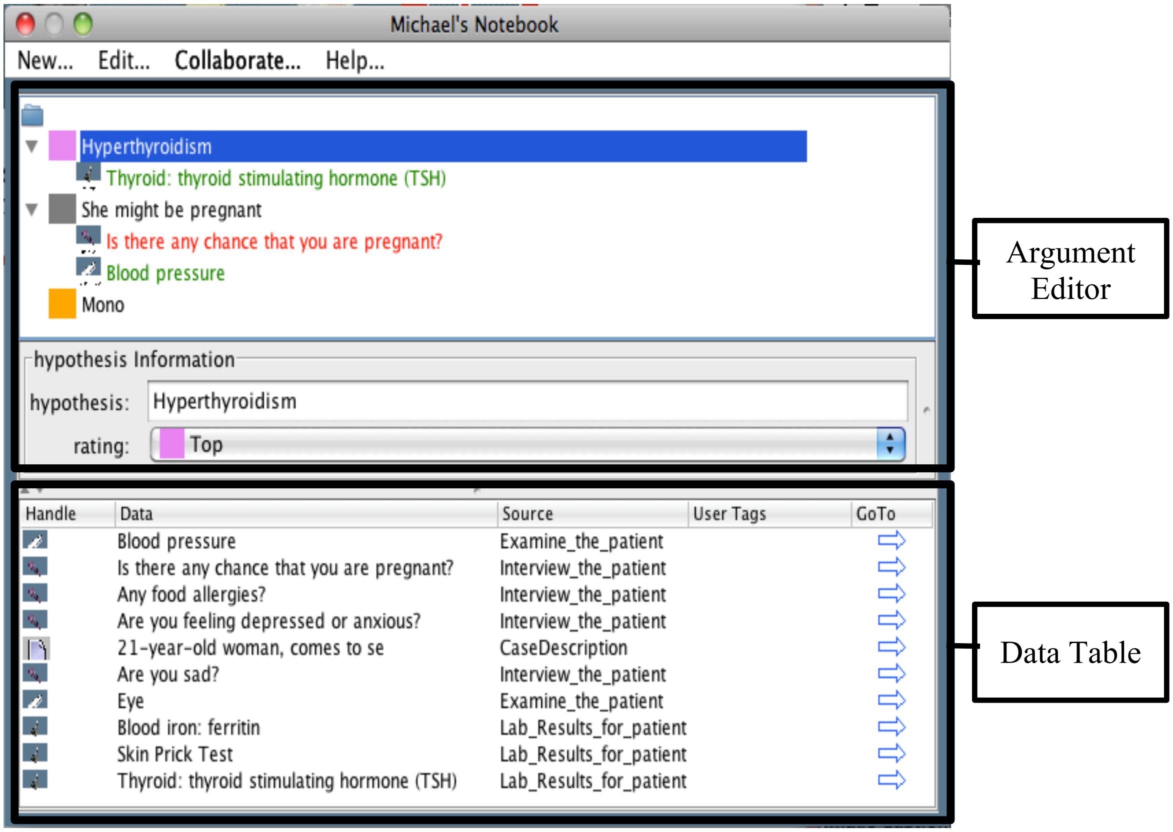

Figure 2.8. The Notebook consists of the **Argument Editor** (top) and the **Data Table** (bottom).

2.3.2.2.3 The Notebook - Argument Editor

As presented above, the Data Table provides a location where data can be collected and viewed as a whole. However, we want students to go beyond simple data collection when engaging in inquiry, developing hypotheses, and making inferences from these data. They should also consider how the data are inter-related and what conclusions can be drawn from these observations. To support the inquiry processes of forming hypotheses, collecting support and refutation, and explicating these relationships, we provide students with the Argument Editor (Figure 2.8. top).

This tool provides a tree-like structure where students can create hypotheses, and can drag and drop data from their Notebook to support and refute these theories. This tool helps students visualize the structure of their solution, to see which hypotheses seem most likely, and whether more data are needed to understand whether a hypothesis is true or false. In the biology domain, the top-level hypotheses are potential diagnoses, and data about the patient are used to support of refute these possible diagnoses.

2.3.2.2.4 Source Editor

The Source Editor allows students to edit their personal list of external sources, helping them organize and remember the location of external information (Figure 2.9). The author of the domain typically creates a pre-loaded list of sources, providing students easy access to some key external resources relevant to the domain. This Source Editor helps ensure that students can utilize the wealth of knowledge available in books and on the Internet, rather than being constrained to the limits of the Rashi system.

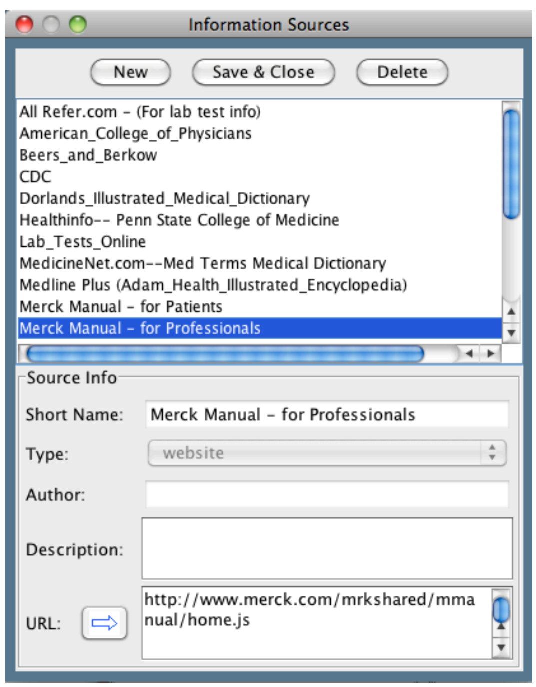

Figure 2.9. The Source Editor displays a pre-loaded set of external biology resources.

2.3.2.2.5 Report Editor

When students have reached some conclusions about the case at hand and would like to present their work in the form of a document, the Report Editor summarizes their work and presents it in a readable format (Figure 2.10). The tool uses HTML to present outline-style text versions of the Notebook contents and the contents of the Sources Tool. This output allows students to reflect on conclusions, re-phrase and re-think their work, and decide upon its meaning. They can then take their results and move outside of the Rashi system, presenting the information to facilitators or other students.

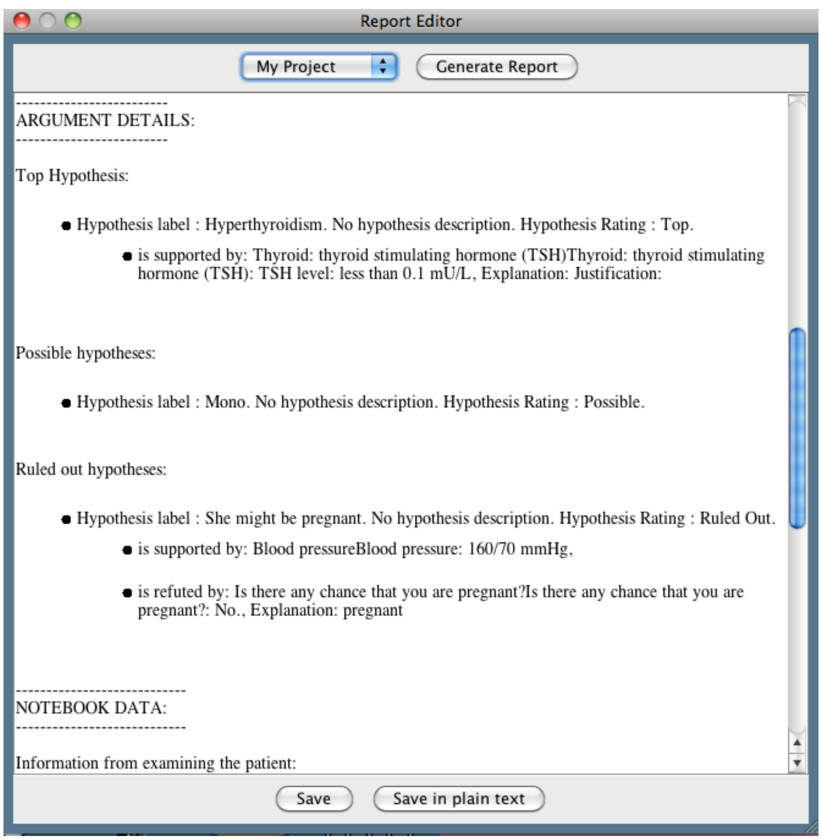

Figure 2.10. The Report Editor displays the automated report generated from the Notebook. The report shown here was generated from the Notebook shown in Figure 2.8.

2.4 Collaborative Inquiry Learning

A broad range of literature demonstrates the potential benefits of collaborative learning (e.g., Dillenbourg et al., 1995; Johnson & Johnson, 1986, Kumar et al., 2007; Soller, 2001), yet many also agree that not all collaborative work is productive or beneficial (Dillenbourg et al., 1995; Soller, 2001; Soller et al., 2005). Collaboration has been shown to be most effective on tasks where student develop complex skills (Soller et al., 2005) and engage in critical thinking (Gokhale, 1995), rather than facing less complex tasks such as rote memorization. As we have shown in sections 2.2 and 2.3, inquiry learning situated within an ill-defined problem space offers precisely this type of complex challenge that requires critical thinking. In order to harness the power of collaboration within our framework, we provide students with collaborative tools that allow for co-construction of knowledge and for students to support one another. Here we present the theory behind collaborative systems built to support students engaged in inquiry learning within ill-defined problem spaces. We also discuss how we move this theory forward in specific aspects. Finally, we describe how Rashi implements both standard features seen in many collaborative environments, and also features that improve the state of the art by encouraging content-focused collaboration.

2.4.1 Theory – Collaboration Supporting Inquiry in Ill-defined Problem Spaces

Within the general field of Computer Supported Collaborative Learning (CSCL), there are many examples of successful systems that analyze and support collaborative work in ill-defined problem spaces. We view this related work in two different categories, those that focus on general collaboration (and particularly dialog), and those that present more complex group interaction, which tend to focus on argument diagramming.

Considering systems that focus on general collaboration, there has been much dedicated research on the idea of micro-scripting, meaning the dialog in which student engage is structured somehow through categories or sentence openers (Dillenbourg & Hong, 2008). Some empirical research on systems using these approaches have demonstrated positive results e.g., (Ravenscroft, McAlister & Sagar 2004) but others have demonstrated that tools can have mixed results depending on the collaborative situation in which they are used (Soller et al., 2001). These micro-scripting techniques offer more structure to collaborative contributions, addressing two problems of collaboration in CBI at once. First, the structure guides students by having them consider ideas and attitudes that may change their perception (Ravenscoft, McAlister & Sagar 2004). Second, the structure can provide the system with some high-level information about the content of conversation without the need to use natural language processing (Soller et al., 2001). However, this tactic does place constraints on student conversation and narrows the conversation to the specific types of comments that are elicited by the imposed structure (Constantino-Gonzales, Suthers & Escamilla, 2003).

Other notable approaches take freeform student chat dialog as the main communication. Researchers have focused on recognizing and supporting discussion with conversational agents, covered further in Chapter 3. Some prime examples of this type of system are the Basilica system and the serious game Urban Science. The Basilica system supports learners by adding dialog components to existing simulation and tutoring software. Examples include the CycleTalk Tutor, teaching thermodynamics, and the

WrenchTalk Tutor, teaching concepts of force and stress (Kumar & Rose, 2011). The serious game Urban Science presents a simulation supporting students to engage in urban planning activities, and providing them with multi-party chat facilities that students can use communicate with team members and mentors (Morgan et al., 2012). The research of collaborative efforts within these systems does not extend beyond the chat facility to support group members to communicate, and yet researchers have produced interesting results. These ongoing research projects use Artificial Intelligence (AI) techniques to understand students' chat contributions and offer feedback accordingly. These approaches will be covered in Chapter 3, however here we stress that the actual collaborative tools and interfaces are quite simple.

Other systems offer more holistic support for collaboration through use of shared workspaces. Examples include COLLECT-UML (Baghaei, Mitrovic, & Irwin, 2007), a collaborative workspace for solving challenges in UML diagramming (an example of an ill-defined problem space, see section 2.2.1). This system provides not only chat functionality, but both private and joint workspaces, where students first work individually to create diagrams, then share their solutions in the public space with small groups, where they attempt to combine their ideas and offer a joint solution.

In a different direction, many systems offer more structured collaborative contributions in the form of argument mapping. Literature review covering development of such systems over the last fifteen years demonstrates that such systems have potential to improve learning (Scheuer et al. 2010) Suthers offers a classic example in this area with the Belvedere system, which provides a joint workspace where students create "inquiry diagrams," collections of hypotheses, data, and evidential relations (Suthers, 1999a). Research has demonstrated the benefits of these argument-diagramming tools, including improvements to the focus and depth of student discussion (Suthers, 2003). Another system in the realm of argumentation diagramming that has developed over the course of many years is LARGO, the legal argumentation tutoring system mentioned in section 2.2.1 (Pinkwart et al., 2007).

The research on collaborative argumentation is ongoing. One major impediment with empirical assessment of the specific argumentation setup is that each system defines its own knowledge elements, representations, and forms of interaction. This creates a problem where the current research cannot be generalized, or easily inform us as to the best design principles, even though many designs have been tested in different scenarios (Scheuer et al. 2010). Recent work addresses this problem by providing general, customizable tools that allow domain experts from any given field to cater a diagramming tool to their specific needs. The systems then support students to create discussion or argument diagrams within the confines the domain expert has defined. Two notable systems in this category are Digalo (Muller & Mizra, 2007) and LASAD (Loll et al., 2012). Each system has its own focus, but generally they provide a space where students can collaboratively build argument diagrams consisting of links and nodes. An external author specifies the types of links and nodes. Digalo is a stand-alone application that focuses on providing a diagramming space that supports students to visualize and organize their arguments about given controversial topics (Muller & Mizra, 2007). LASAD offers a generalized, web-based system that can implement many other previous systems for collaborative or individual argumentation, including the previously mentioned systems, Belvedere, LARGO, and Digalo (Loll et al., 2012).

Both these projects both have been successful in meeting their own particular goals, and have demonstrated the importance of sharing not only textual statements, but also of sharing more complex representations to communicate ideas within small groups (Suthers, 2003). However, the work students complete in these systems is generally detached from any data collection or other thinking tools that might provide students support and offer a more holistic learning experience, such as that provided in Rashi. Having dual interaction spaces, e.g., one to collect data and one to reason about that data, offers a challenge for learners, and the less integrated the spaces, the more difficult it is for learners to collaborate (Mühlpford & Wessner, 2009).

To this end, our work seeks to build from this knowledge and move beyond such isolated environments. Our approach attempts to solve several issues that arise from using separate, isolated tools. First, we consider that student collaboration should not be done solely through dialog, but rather students should also be able to clearly see and share work about the larger context in order to bring relevant inter-relations to the forefront of conversation (De Jong et al., 2010). The goal is to promote not only general discussion, but also to have dialog linked with specific content. This goal is complicated within the above-mentioned argumentation systems where all domain and task–level ideas and statements are taken from outside sources, not collected or controlled within the system.

Our goal is to develop environments that not only encourage students to collaborate, but support them in referencing and focusing attention on the specific relevant content within their current work as they collaborate. We call this *content-focused collaboration* and as we design our system, we seek to promote this specific, focused type of collaboration as much as possible. We are in a unique position to do this, as the data collection tools are part of the Rashi system (see section 2.3.2.1), rather than data being collected from some external source. We now describe the implementation of collaboration within Rashi, and demonstrate how it meets the goals put forth by other projects, but also moves beyond these systems, to help students focus their collaborative efforts on content.

2.4.2 Implementation - Collaborative Tools in an Inquiry Learning Environment

Collaboration in the Rashi system is implemented in two locations: a Chat Tool, and additional collaborative features included in the Notebook. First we discuss this new Chat Tool, which is built explicitly to support content-focused collaboration among students working in groups. We then discuss how the Notebook has been altered to support a wide-range of collaboration, but also to specifically promote content-focused collaboration. With the additional functionality provided by these two enhancements, we can see that Rashi is well situated among collaborative systems, providing both dialog support and shared workspaces.

2.4.2.1 The Chat Tool

The Rashi Chat Tool is built to function as a standard chat tool (e.g., AOL Instant Messenger). A different "chat room" is associated with each group (usually 3-6 students) and presents the statements made by different users chronologically (Figure 2.11). Additional features given in the top control panel allow the user to move to different chat rooms as well as to filter the chat messages by different criteria (e.g., username). This functionality allows class-wide interaction and the ability to search messages for content. The text is color-coded, providing each student with a different color to differentiate between speakers. Finally, the bottom panel allows users to specify for whom the message in intended (the entire group or individual students), as well as a subject for the message. Students can provide a subject for each chat message, which allows the team to recognize and denote when they focus on a specific topic. Chat messages can be filtered by these topics and students can easily respond to the subject by clicking on it.

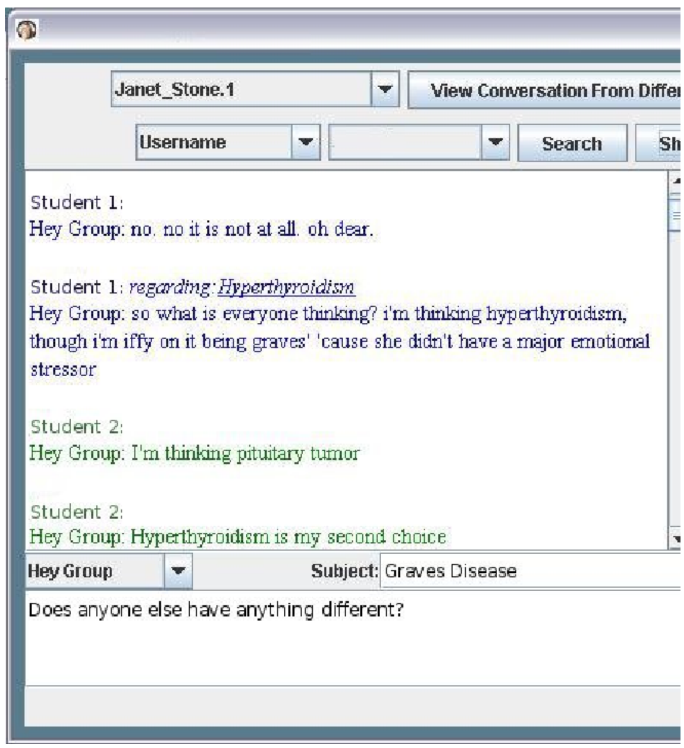

Figure 2.11. The Chat Tool provides standard chat functionality but also provides extra features specifically aimed to support content-focused collaboration. Here one student chooses to focus on Hyperthyroidism.

2.4.2.2 The Collaborative Notebook

The Rashi Notebook has been modified in a number of ways to enable students to view and share work within a group. There are three basic functional developments that promote collaboration beyond discussion, allowing students to share work within their groups. First, the "compare" option supports students to compare their own notebooks with other members of their group. This option presents users with a read-only version of another student's notebook (Figure 2.12). Next, students can copy individual pieces of content from other students' notebooks through drag-and-drop functionality. Lastly, the system can be configured to offer a "group notebook," a shared notebook editable by all members of the group. Using the combination of these three functionalities (compare, copy and group space), the system supports a variety of collaborative activities ranging from students working in tightly knit groups, where each student takes on a role and contributes in a specific manner in a group notebook, to students working mostly independently but sharing ideas and thoughts when they reach an impasse.

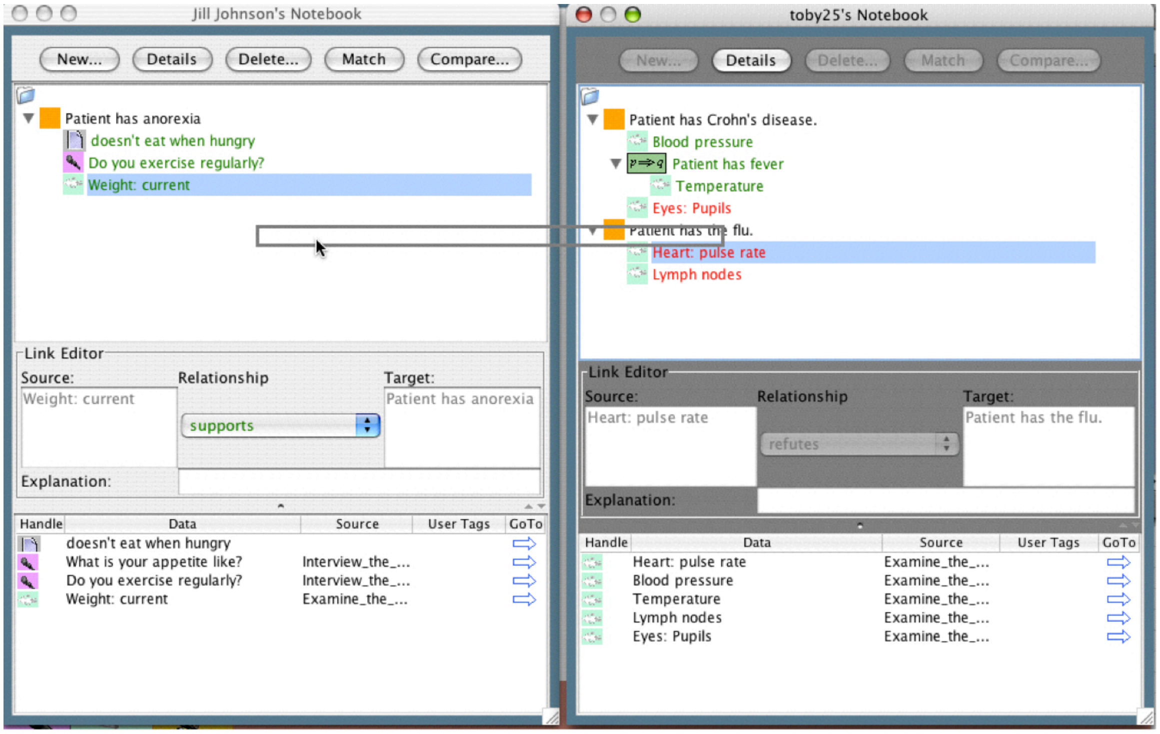

Figure 2.12. Comparing Notebooks allows users to compare their work with others. Here, the student Jill has her own Notebook open (left), and is also viewing a classmate's notebook (right). She is then moving data from this other notebook to her own with the drag-and-drop functionality.

2.4.2.3 Content-focused Collaborative Features

Beyond the standard features that situate Rashi within the current state of other collaborative research presented in section 2.4.1, we seek to create new features that take the lessons learnt and extend the system to offer students more intuitive and improved ways to focus their discussions on the relevant content. These system enhancements rely on a base concept of *discussable objects*. Discussable objects represent any piece of information within Rashi that might become the subject of conversation, e.g., hypotheses and data that students collect and create. We now describe the two features that promote higher-level content focus using these discussable objects.

First, the system enables students to automatically create chat messages referring to a certain discussable object. Rashi allows users a simple method of automatically setting the subject of a new conversation to the content of an existing Rashi notebook item. For example, a student might be confused about her current hypothesis "hyperthyroidism." She can right-click that item in the notebook, and choose "discuss this" from the contextual menu. The system then creates a new chat message with the subject "hyperthyroidism" and also creates an internal link between the conversation and the notebook item, allowing a group member to click on the chat subject and be taken directly to the related work in a group member's notebook. These features potentially enhance the ability for students to focus their conversation on the most relevant information with respect to their current content interest.

The ability to chat about a certain notebook topic is useful, but there are also clear issues with a system taking work out of a student's visual context and moving into a chat environment. Research has demonstrated that comments located in a separate space (where one must shift context to view comments or have conversation) can be less helpful to students than co-located comments (Mühlpford & Wessner, 2009). To address this issue, we created an interface where students can directly communicate about a certain notebook item, in a set of critique/rebuttal responses (Figure 2.13).

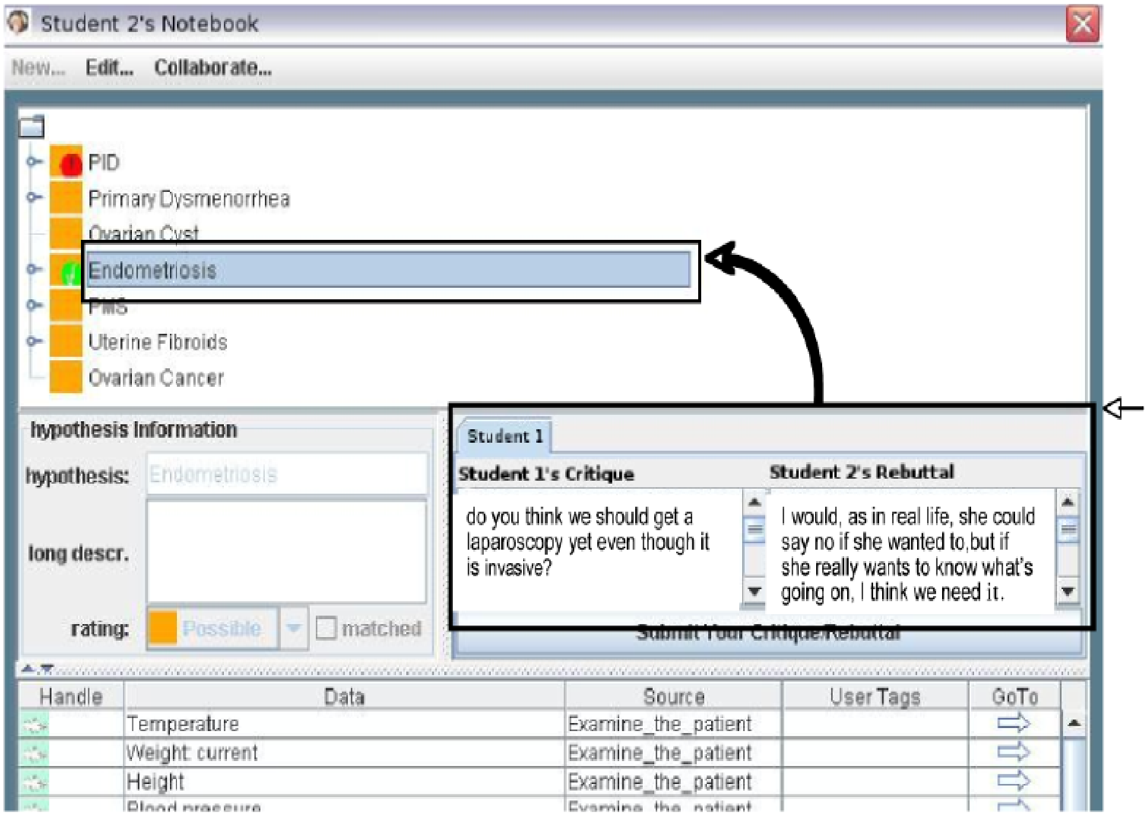

Figure 2.13. The Critique-Rebuttal feature provides a way for students to dialog about specific Notebook items. Here one student questions a hypothesis and the author of that hypothesis is explains her thinking.

This Critique-Rebuttal feature supports students' engagement in topic-oriented discussions. Built into the notebook, this feature enables students to select any item in a group member's Argument Editor and to offer critiques about them. When a critique is given, the owner of the notebook item is notified, and he/she can respond with a rebuttal, a defense for his or her position (Figure 2.13, middle). This back and forth discussion is by definition focused around subject matter. The two parties can continue to update and resend their critiques and rebuttals, fostering a dialog specific to an issue. This feature more tightly couples conversations with notebook content.

All of the above functionality helps to address a current concern in systems with shared learning spaces, by focusing students on content. This has the potential to promote more engaged and organized discussion. With these various features centered on the concept of discussable objects, Rashi offers a collaborative space that can promote content-focused collaboration, centering discussion on pertinent domain knowledge, and bringing that content to the forefront of dialog.

2.5 Summary

This chapter presents three distinct pedagogical concepts (ill-defined problem spaces, inquiry learning, and collaboration), the ways in which current research in the field of computer-based instruction has been enlisted to support these concepts, and the ways in which our specific research project implements these concepts. We consider the large class of ill-defined problem spaces that are crucial for education, and we look to two different educational approaches that can help students learn in these ill-defined problem spaces, inquiry learning and collaboration.

Ill-defined problem spaces present interesting teaching opportunities, offering many real-world, complex problems that promote deep knowledge acquisition and critical thinking, if approached correctly. We present the subject matter of medical differential diagnosis as a fruitful example of an ill-defined problem space that can be used to teach biology. However, with differential diagnosis, as with other subject matter in ill-defined problem spaces, a successful learning experience requires both structure and support, as students can become confused, lost, or stuck.

One specific pedagogical approach that can provide this structure and support is inquiry learning. In the inquiry learning process, students form hypotheses, collect data, and find the relations between their data and their hypotheses, supporting and refuting initial ideas and forming new ones. We discuss the tenets of inquiry learning, how it raises concerns of classroom implementation, and how ultimately these concerns can be addressed with computer systems that can allow for and support inquiry learning in a classroom. We present our system, Rashi, which implements the pedagogical approach of inquiry learning in a domain-independent way. To continue with our example, we present an instance of the Rashi system based on differential medical diagnosis, and we discuss in detail how students can engage in inquiry learning in biology by using the tools provided within the Rashi system.

Finally, we consider the concept of collaboration within ill-defined problem spaces. Efforts to employ collaboration fit naturally into ill-defined problem spaces, as research has demonstrated that collaboration is most successful when dealing with complex situations requiring critical thinking skills. We see a large number of attempts to merge inquiry learning and collaboration in computer supported collaborative learning for the same reasons. We see the benefits of systems that contain both space for dialog and shared workspaces. We offer our own implementation of tools in Rashi that encourage students to collaborate through dialog and shared workspaces. Moving beyond those concepts, we present functionality included with our tools to promote content-focused collaboration. These enhanced functionalities support students to directly link learning content in their conversations, and allow them to have conversations situated within their learning content, tightening the link between their collaborative efforts and their domain-related work.

Overall, this chapter presents a challenge to the CBI community: namely that of supporting instruction in ill-defined domains with computer systems. We present two pedagogical approaches that can provide structure and support within these domains, inquiry learning and collaboration. Finally, we present ways in which a system can focus students' collaborative efforts on content and ease the processes of both the collaboration and group inquiry. With all of this taken into consideration, we argue that computer systems can tackle instruction in ill-defined problem spaces, and that promising approaches include inquiry learning and collaboration. Chapter 3 presents a further argument for this case, namely that a computer system can understand students' inquiry work and their collaborative contributions in such a system, and therefore offer automated support to improve the learning process.

CHAPTER 3

SYSTEM INTELLIGENCE – RECOGNIZING STUDENT CONTRIBUTIONS AND PROVIDING FEEDBACK

3.1 Introduction

One general challenge of Intelligent Tutoring Systems (ITSs) is to provide a computer system that can understand some aspects of the student's knowledge and provide specific support tailored to this understanding (Corbett et al., 1997). When taking this view, there are two major steps to creating an ITS that should be considered: understanding student work, and offering feedback based on this understanding. This chapter will consider each of these matters as it pertains to our target teaching system, an inquiry-learning environment for ill-defined problem spaces.

3.2 Understanding Student Work

Classic ITS approaches to understanding student work in well-defined domains tend to focus on aspects such as subject matter mastery and problem difficulty (e.g., Koedinger et al., 2000; Singh et al., 2011). These aspects can be recognized from the small, well-defined interaction space provided, for example, by multiple-choice problems that enable a solution space to be completely mapped. Knowledge tracing tutors (Anderson et al., 1995; Koedinger & Anderson, 1993) use mappings of entire solution spaces to identify a student's knowledge, and to identify the kind of feedback that might be given in any specific situation.

We face a greater challenge when considering collaborative and inquiry-learning environments for ill-defined problem spaces (see Chapter 2 for definition). Students explore open-ended spaces in unique sequences and often use tools or simulations that result in intractable or infinite solution spaces (Lynch et al., 2009; Mitrovic & Weerasinghe, 2009). In Chapter 2, we presented examples of these learning environments, through both our system Rashi and examples from other systems including Belevedere (Paolucciet al., 1995; Suthers 1999a), LARGO (Pinkwart et al., 2007), LASAD (Loll et al., 2012), and COLLECT-UML (Baghaei et al., 2007). These other systems all offer diagramming spaces with free text entry, an input mechanism with an infinite solution space. For example, in LARGO, students study transcripts of Supreme Court cases, and attempt to make graphical representations of the arguments involved. As can be seen in Figure 3.1, this creates a large and complex solution space.

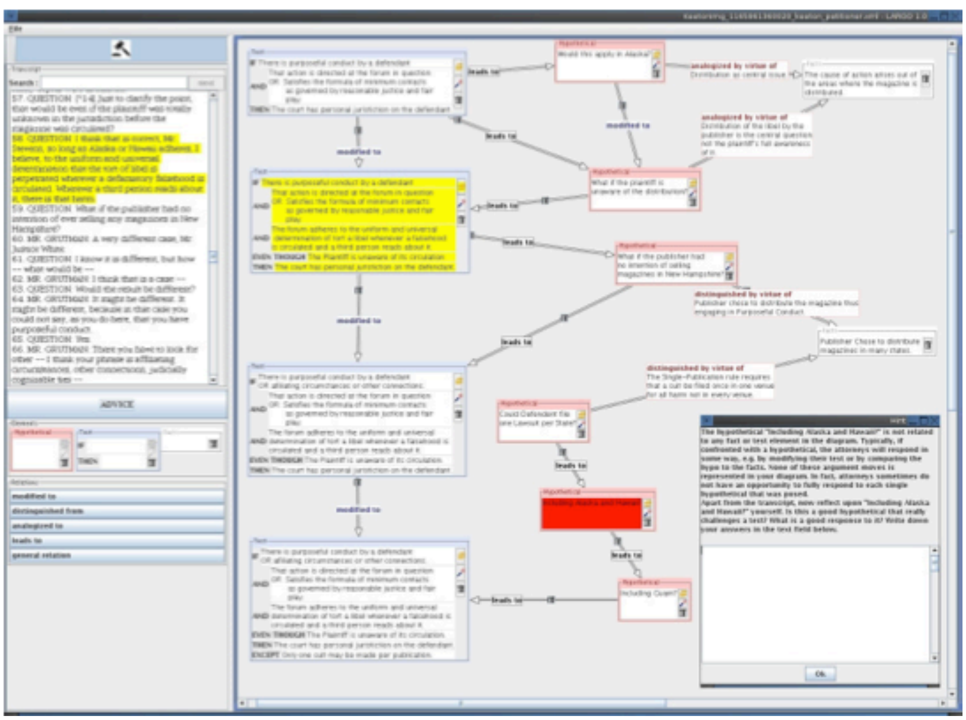

Figure 3.1: A student solution in the LARGO ITS, as presented in (Pinkwart et al., 2007)

Similarly, in Rashi students collect or enter data for a given case, enter free text hypotheses, and make any arbitrary set of relationships between data and hypotheses using the Argument Editor in their notebook (see Section 2.3.2.). In these types of systems, the vast solution space means that novel tactics are necessary for understanding student work. One accepted tactic to recognize important information across these open-ended solutions spaces is termed *constraint-based tutoring* (CBT) (Mitrovic et al., 2007). The concept of CBT is that the author places constraints on the space of acceptable solutions, rather than trying to map out each individual, unique, acceptable solution. In this way, CBT systems do not completely recognize all intricacies of any given student solution, but rather the systems recognize and categorize when student solutions do not meet certain constraints. This high-level idea of defining constraints on solutions space rather than enumerating all solutions within a space is employed by most systems that operate in ill-defined problem spaces or collaborative environments. Such constraints usually take the form of rules about characteristics of the solution space, and are often applied using a rule-based programming language (e.g., Dragon et al., 2006; Guiteirrez-Santos et al., 2010; Scheuer et al., 2010).

Constraint-based modeling provides an overall approach, but a question remains for anyone seeking to implement the technique: on which aspects of the student solution will the constraints, or rules, be defined? Two major types of solutions are offered in this respect: those that purely analyze the structure of student work (e.g., the syntax of diagrams), and those that go further and attempt to recognize the content of student work (e.g., the subject-matter of free text entry). We discuss each of these approaches in turn, and consider the implementation of these different approaches within the Rashi system.

3.2.1 Recognizing Structure of Student Solutions

The first and most scalable approach to defining rules over a solution space is to consider the structure, or syntax, of a student solution rather than the content. This approach is often the first attempted, as the key components of such a structure are well-defined within each given system. For example, all of the diagramming tools mentioned earlier have (at their base) components of visual graphing tools, namely nodes and links. Often the types of nodes and links are pre-defined within the system and are inherently tied to certain domains or tasks (Pinkwart et al., 2007; Baghaei et al., 2007). Depending on the meaning and purpose of the tool, different rules can be defined as to how these nodes and links should be used. For example, when creating a legal argument in LARGO, the student can create a 'hypothetical' node to represent a hypothetical statement from their analysis of a legal case. However, when creating a 'hypothetical' node, the student should also link this with actual facts of the current legal case to clarify the usefulness of this hypothetical in the current situation (Pinkwart et al., 2007). Therefore the system can apply a rule such as the following:

- IF ('hypothetical' node) -> THEN ('hypothetical' node should be connected to 'fact' node)

Students' solutions can then be split into those that meet the constraint and those that violate the constraint. Using this method, constraints can be checked against student work without understanding the actual content of the students' 'hypothetical' nodes or 'fact' nodes. In Figure 3.1, the red box is highlighted to alert the student that this ‘hypothetical’ node is not connected with any ‘fact’ nodes, so it is a potential problem. Similarly, other systems use the pre-defined aspects of their tools to allow for recognition of potentially successful and potentially problematic solutions (Baghaei et al., 2007, Scheuer et al., 2010).

While this tactic of recognizing only the structure of an argument holds promise for offering some limited understanding of student work, we recognize some inherent weakness of the approach due to the shallow level of understanding. First, the system relies on the idea that students are appropriately contributing content within the structure. To return to the example of LARGO, the system’s understanding is dependent on the student actually entering an appropriate hypothetical statement in the ‘hypothetical’ node, and appropriate facts in the ‘fact’ nodes. The system has no ability to recognize an error in this respect, such as a student entering a factual statement in a “hypothetical” node. This is one example of how structural recognition does not offer an over-arching solution to understanding student work, although it can be useful in simple situations. One can also note the inherent weakness of recognition based solely on structure in the general trend that the more mature systems move away from pure structural constraints. In general, as analysis techniques and systems mature, researchers tend towards offering some type of content support, either through automated recognition (Suthers, 1999c) or through collaborative efforts (Ashley & Goldin, 2011).

3.2.1.1 Implementing Structural Recognition of Student Solutions

Within the Rashi system, we discuss the strategy of considering student’s structural patterns within the Argument Editor section of their Notebook to analyze the structure of a student solution. To understand the structural patterns considered, one must first understand the pre-defined base components of the student solution in Rashi: hypotheses created in the Notebook, data collected or entered in the Data Table, and the relationships created between hypotheses and data (see Section 2.3.2). Several tenets of productive inquiry behavior (Paolucci et al., 1996) can be converted into constraints on student work within the Notebook. Table 3.1 demonstrates how these behaviors can be directly represented as rules imposed on student solutions in Rashi.

Table 3.1. Example productive inquiry behaviors as given in (Paolucci et al., 1996), and their matching constraints in the Rashi system.

	Productive Inquiry behavior	**Constraint in Rashi system**
1	Follow multiple hypotheses	(number of 'hypotheses' > 3)
2	Attempt to identify both supporting and refuting evidence for each hypothesis.	IF ('hypothesis') -> THEN ('support' and 'refutation' present for 'hypothesis')
3	Lines of argument should not be circular	IF ('node1') SUPPORTS ('node2') -> THEN ('node2') NOT SUPPORT ('node1') (where SUPPORT and NOT SUPPORT are defined recursively)

Many such rules have been implemented and tested in the Rashi system, and some are still in use today. Through early pilot testing and classroom feedback, we found the most simplistic of these rules to be effective, e.g., identifying lack of hypotheses, or lack of data collection, etc.

However, we also found during development that the purely structural type of recognition can be non-productive in complex scenarios. For example, considering Rule 2 from Table 3.1 ("IF ('hypothesis') -> ('support' and 'refutation' present for 'hypothesis')"), practical scenarios arise where the advice is not productive. For example, there may be a case in which there is no distinct refuting evidence. This can be designated as a weakness in the authoring of the case, but given that we have an ever-growing system where independent authors can offer case details, we consider this a general problem that must be taken into account.

On a deeper level, we also found cases where these complex instances of structural support were irrelevant or misleading when considering the content of the solution. For example, one piece of refuting evidence can make a hypothesis impossible, leaving no reason to pursue supporting evidence. These clashes, where theoretically useful structural pattern recognition was deemed counter-productive, were identified when working with *Subject Matter Experts* (SMEs) and reviewing actual feedback scenarios from structural recognition, demonstrating that there are practical problems with complex structural analysis, not just theoretical concerns.

3.2.2 Recognizing Content of Student Solutions

With these early results in mind, we pursued methods to understand student solutions on a deeper level that could be used in addition to the simpler, structural measures presented in Section 3.2.1. Some tactics used by other researchers include applying search techniques over large corpuses of existing data, possibly being other students' work (Bernhard & Gurevych, 2008; Ravi et al., 2007). These researchers attempt to match new student information with related content generated by prior users. Ravi et al. search prior student work to find adequate matches (Ravi et al., 2007), while Bernhard and Gurevych analyze the wikiAnswers data store in order to support students (Bernhard & Gurevych, 2008). These efforts frame the problem in terms of data-mining and natural language recognition challenges, sorting and filtering through large data sets for information that is relevant to specific student input. However, the quality of the retrieved information and the semantic meaning of this information (how it is "situated" among other knowledge components) are unknown. Using this type of strategy, a student might be provided with a variety of possibly relevant information, but the system cannot rest assured support has been provided. The system also gains little to no understanding of students' solutions during this process.

A different, well-established method for enabling a computer system to understand a deeper level of student knowledge is by providing an *expert model* that encapsulates expert understanding of subject matter at hand (Beck et al., 1996). This expert model is then used as a means of understanding the student. We consider here the class of expert models that works from a *knowledge base,* a semantic representation of individual knowledge components and their inter-relations. Student input is then matched to these individual knowledge components in some manner. The system can then use these links to the knowledge base to assess student work. The concept is well established, with research conducted in this realm for over thirty years (Brusilovsky, et al., 1996; Clancy & Letsinger 1982; Crowley & Medvedeva 2006; Kazi et al., 2009; Paolucci et al., 1996;). However, the subject is less popular in current research, and when employed, developers tend to use pre-defined knowledge bases or ontologies (Kazi et al., 2009). This lack of current research is based in part on the belief that creation of an expert model is inherently time-consuming and potentially not worth the return on time-invested (Anderson et al. 1996; Kazi et al., 2009; Mitrovic, 1998). We argue that a team including an SME can produce small, focused knowledge bases directly useful to assessing the student's work on the task at hand without prohibitive cost, provided an appropriate authoring tool (Murray et al., 2004). We now describe how our knowledge base is defined, implemented, and used to recognize student solutions.

3.2.2.1 Implementing Knowledge Bases

Rashi is a pioneering system in providing several limited, succinct knowledge bases for identification of domain content within the larger, domain-independent framework. The Rashi system accomplishes this through the use of an authoring tool, where developers and SMEs work as a team to rapidly develop knowledge bases geared towards specific tasks and domains (Murray et al., 2004). Our knowledge bases consist of knowledge components we term *propositions.* There are three given types of propositions:

- Hypotheses: high-level possibilities of reasonable explanations of the phenomena presented within the given domain.
- Data/Observations: facts and low-level details that can be directly observed within the environments.
- Inferences: mid-level conclusions that do not answer the question at hand, but offer insight beyond observable fact.

These propositions are inter-connected with relationships, which we define as supporting or refuting. Inferences are related to hypotheses, and data are related to either inferences or hypotheses. Thus, the knowledge base is a directed, acyclic graph with 'hypotheses' being the root nodes of the graph and 'data' being the leaf nodes (see Figure 3.2).

When understood at this abstract level, we see then that the SME's task is similar to a student's task: creating a graph that offers a 'solution' in the given domain (a set of hypotheses, and a representation of the relationships with the data available in the system). This expert solution may be similar (but more complete) than that of an ideal student. The resulting graph we term the *expert knowledge base* (EKB).

The fact that the Rashi environment includes data collection (see Section 2.3.2) is a major contributing factor to the scalability of our approach. Since data nodes are already defined when authoring the domains and cases within the system, the most populous portion of the EKB (these data nodes) is already available in our database to be used as expert knowledge. The expert need only create higher-level inferences and hypotheses and relate them with these data nodes.

In Figure 3.2, the knowledge base is presented within the Argument Editor of the Notebook within Rashi. Although the SME does not build the knowledge base within this tool, they can use it to view the knowledge base as a student solution for a holistic view of the current knowledge base.

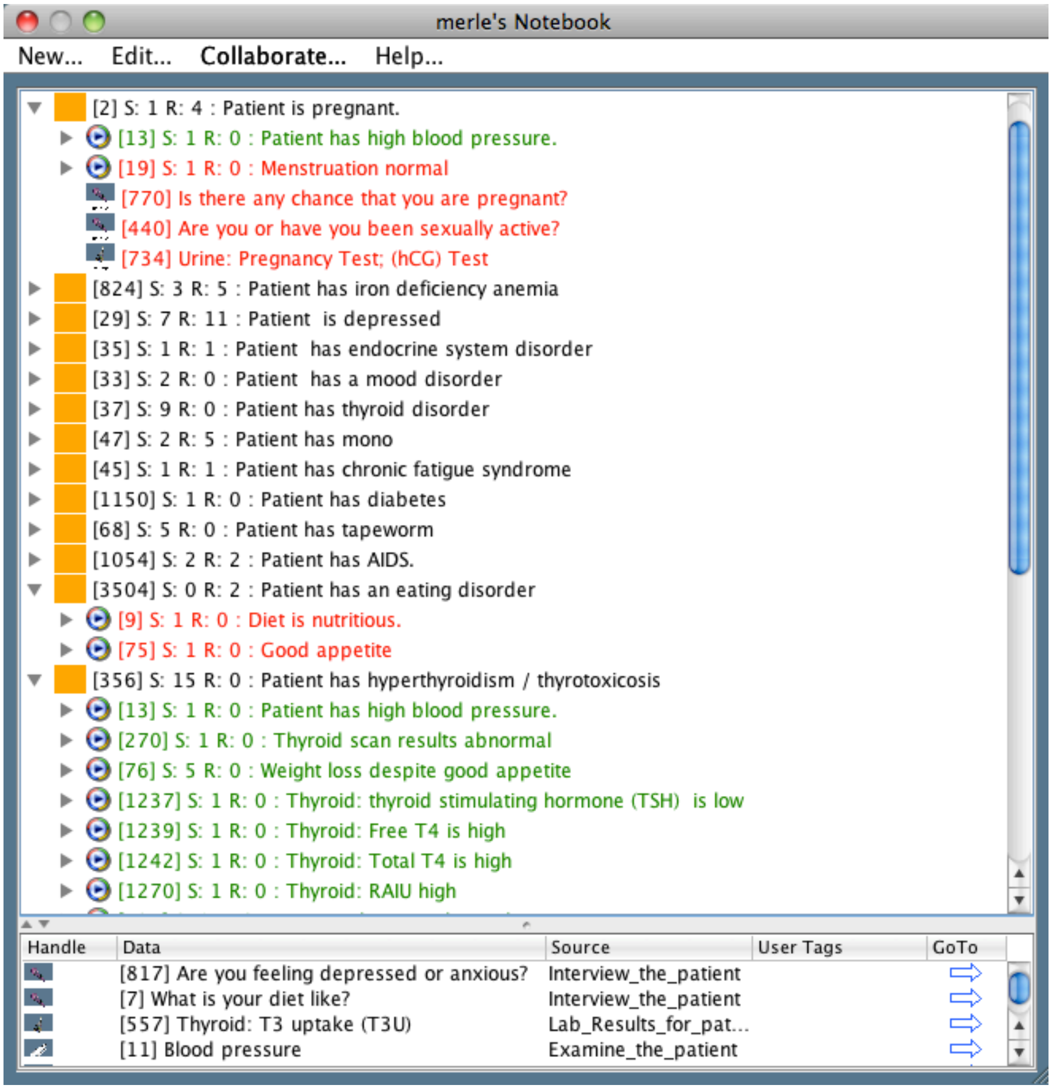

Figure 3.2. A portion of the Expert Knowledge Base for Biology, as seen within the author's view of the Argument Editor of the Rashi Notebook.

For our discussion we focus on the knowledge base for the biology domain, which is defined specifically to help recognize certain medical diagnostic tasks. In this domain, hypotheses are potential diagnoses (e.g., "Patient is pregnant," "Patient has hyperthyroidism," etc., as seen in Figure 3.2). The knowledge base has been built iteratively over years using input from classroom studies to feed back into the authoring process. This knowledge base is supplemented to suit new cases individually as they are added to the system. In addition, student work is analyzed post hoc to identify important missing portions of the knowledge base to be added in later iterations.

Rashi is one of the longest-standing systems to provide this type of knowledge-based recognition in a medical diagnosis framework, along with Bioworld (Lajoie, 2001). Over time however, other systems have taken the same approach, for example COMET (Kazi et al., 2009). This system uses a large, pre-defined medical database as an expert model adapted for teaching purposes.

The Rashi system differs from most other medical systems in that we use a succinct EKB specifically tailored for the domain and case at hand. Another system that attempts some generality in their knowledge base content is the VCT system, which allows for different tasks to be encoded in a generic knowledge base geared towards their specific domain of visual classification (Crowley & Medvedeva, 2006). However, this system is still tied to the domain of visual classification. In contrast, Rashi is completely domain-independent (not tied to any one pre-existing expert model), and can be used within a wide range of domains, including geology, art history, forestry and biology (Dragon et al., 2006). Our knowledge base design and authoring tool make the creation of these distinct knowledge bases and the matching of student work viable without complex *Natural Language Understanding* (NLU) techniques, as can be seen by our empirical results (see matching validation, Section 4.4.2). We have also demonstrated, as have others (Adamson & Rosé, 2012), that by limiting the size of the knowledge base, one can actually improve the confidence of the recognition (Floryan et al., 2012). This type of personalized manipulation to improve recognition results lends more weight to the concept of defining one's own, more concise databases rather than relying on external data sources.

3.2.2.2 Using a Knowledge Base for Content Recognition

Once a knowledge base has been established, the system must connect student input to individual knowledge elements within the knowledge base in order to gain some understanding of a student solution. A spectrum of student's allowable student input exists in this regard. One end of the spectrum, which we term *restricted input*, provides students with access to the basic components of the knowledge base and only allows students to explore/manipulate the concepts represented within that knowledge base (e.g., through drop-down menus). The other end of the spectrum, which we term *free input*, allows students to freely enter their own information (e.g. free text entry) and does not provide students with any information about the knowledge base itself. In a free input system, there is no explicit link between the student input and the knowledge base, and so the system must actively connect student work to the knowledge base in some automated fashion.

A major trade-off exists here, one that balances the goal of recognizing student input with the overall goals of the pedagogical approach. In a *free input* system, there is possibility a match will not be found. When a student is allowed to type any sequence of characters, his/her input may not be recognized due to typographical errors, different phrasing, etc. The student may be on a perfectly valid line of thought that is not properly associated with a knowledge base element, or not present in the current knowledge base at all. In either case, the system cannot link the student work with the knowledge base because of the input mechanism.

Alternatively, in the case of a *restricted input* system, the student would not be allowed to enter arbitrary input and instead would be forced to work solely with knowledge base elements. This could potentially prevent the student from following his/her own line of thinking, and instead force him/her to use only the specific phrasing encoded in the expert knowledge base. The analysis of a student's argument will then be thorough, but may have ceased to be in line with the student's natural way of considering the problem. A final concern is that a restricted input system may also provide the student with too much information by displaying terms and solutions with which they are not familiar, but are presented by the input mechanism in a way that students can use them without real understanding of their meaning. For example, a drop-down list of possible hypotheses might provide lead a student to utilize a hypothesis that he/she does not understand.

Authors of systems that recognize student input by linking to a knowledge base must decide on their placement along this spectrum, according to their pedagogical goals of freedom and self-sufficiency, balanced by their desire for accurate and complete system recognition. Although all researchers must decide a position on this spectrum, we find that this decision may be arbitrary or left to practicality, which is a weakness in the current field. In future endeavors, we suggest that this concept be tackled directly when decisions about free input versus restricted input are being made.

3.2.2.3 Implementing Knowledge Base Content Recognition

In Rashi, we take a mixed approach to the restricted versus free input question, finding ourselves somewhere in the middle of the spectrum, with a slight bias towards free input. To consider our position more closely, we examine how students create each of the components of their solutions: data, hypotheses, and relationships.

For data collection, we do not restrict input. Students are free to type any entry into their data table at any time. However, we have automated ways of collecting data propositions through the simulation tools (see Section 2.3.2.1). This data collection ability is used significantly more often than typing entries into the data table, both because it is intuitive and because our pedagogical approach in the classroom encourages students to take this tact (i.e., when introduced to the system, students are told that they should collect data from different simulation tools). For these reasons, in practice, almost all data are collected directly within our system from our tools. Therefore, our data collection method provides the most of benefits of restricted input, in that most data propositions (e.g., patient's temperature, lab results, etc.) are already recognized and connected to the knowledge base. However, students are still free to work outside of this schema.

Next we consider relationships, and here Rashi uses a completely restricted input method. When creating relationships, students choose from a pre-specified set of supporting and refuting relationship types. They are not allowed to enter their own types of relationships. This enables the tutor to automatically understand the type of relationship that students are creating, which is crucial for the knowledge base assessment methods.

Finally, we consider hypotheses, about which we take quite the opposite tact. The system for entering hypotheses is total free-text entry with no knowledge base visibility for students, meaning it is a free input method. Students have only blank lines in which to type their hypotheses, and the system must attempt to connect these free-text entries with the knowledge base elements. As the other portions of the system are mostly restricted input, the main recognition task within the Rashi system is to link high-level hypotheses created by students to the EKB. Our SMEs strongly believe that providing any list of hypotheses or other more restricted input methods would narrow students' focus and have a negative impact on their inquiry behavior. This potential concern is exacerbated by the fact that our knowledge base is custom built and therefore specific to our cases, leading students quickly to "the right answer" and not supporting their exploration and inquiry skill formation. For these reasons we chose the free input method, but this could easily be modified in the future to include partial or restricted input according to teachers' needs.

The input methods covering data, hypotheses, and relationships were chosen to balance the needs of the recognition system with the pedagogical requirements from our SMEs and teaching methodology. The result of these balancing factors is that Rashi presents a fairly middle-of-the-road answer to the tradeoff between free input and restricted input. Data and relationships are predominantly being collected from pre-set recognized knowledge base items. This method provides the analysis system with both recognized data elements and recognized relation types that allow comparison with the EKB. Due to the more strict pedagogical requirements, the students have completely free input when creating hypotheses. This combination results in a system that recognizes a large portion of student work but still has a crucial free-text-entry matching task to perform before it can recognize hypotheses and offer an assessment of a student solution.

This hypothesis-recognition task simplifies to the following problem. The system has a set of hypotheses typed by users and attempts to find matching propositions from the knowledge base. As NLU techniques mature, off-the-shelf software will become more and more useful for situations such as this and provide us with easy tools to accomplish such a task. Some researchers, particularly in the area of computer supported collaborative learning, have been developing middle ground techniques to balance the needs and issues of complex NLU techniques with simpler systems to provide generic, practical, and usable NLU technology (Rosé et al., 2008). However, as our main research focus is not in language understanding, we substitute a simple tactic involving software that is currently freely available. Specifically, we index all propositions from the knowledge base, along with associated keywords and stop words, in the search engine library Lucene (lucene.apache.org). We then run each student statement as a query and use the results to match student propositions to propositions in the EKB.

We have conducted iterative design on several interface possibilities to identify a way to confirm with the student that these hypothesis-knowledge base matches are correct. These attempts were not accepted well by students, and so this matching interface went largely unused. Attempts to force matching by students were deemed intrusive and provided too much information to students (by exposing knowledge base elements), and therefore also rejected.

Our current working solution is to automatically match items, by using the top-rated result from the search results. This clearly increases the risk of incorrect matches, yet through investigation and testing, we generated encouraging results showing that the top match from the algorithm was an appropriate match in 82.8% of instances from our test data, and that the match was appropriate in 98.2% of situations where feedback might have been offered based on this match (see Section 4.4.2.1). We consider this accuracy to be totally within acceptable range, but also realize we had a limited data set for analysis and therefore must consider that this number may vary. We carefully consider the implications of using this noisy matching scheme (as addressed in Section 3.3.4) but we also see our results as reason to proceed with our current method of matching.

To demonstrate the matching process, we provide an example hypothesis from the EKB, "Patient has Mono" (as seen in Figure 3.2 middle). "Patient has Mono" is the statement the SME chose to represent this knowledge base element, and also included the keyword "mononucleosis." Students typed many different statements using matching terms (e.g., "mononucleosis," "Fatigue is caused by mononucleosis," "This person may be suffering from mono," etc.). In each of these cases, the highest rated match to the knowledge base was the "Patient has mono" proposition, and therefore these student statements were matched to this EKB element. This matching can then be used to provide support to students, as discussed in Section 3.3.

Certain aspects of the biology domain are particularly suited to this recognition system. Discussion around medical diagnosis often employs precise and consistent terminology, and novices tend to repeat this terminology in their hypotheses and discussions verbatim. Similar efforts to apply the recognition system in forestry proved less fruitful with the same matching scheme due to the greater variety of phrases and the ease in which novices can phrase the observed phenomena in their won words. Such topics that employ a more generic or varied vocabulary might not be suited for such a matching technique, or at least may require better NLU techniques.

3.2.3 Recognizing Content within Collaboration

We have discussed our tactics for recognizing individual student solutions offered within Rashi for the purposes of evaluating student work, and we now consider collaboration features that provide additional information about the student state to potentially be used in recognizing a student's content focus. Current research provides insight as to how this information might be used for recognition, but also demonstrates some room for improvement.

The first task from a system design point of view is to revisit the input spectrum, and clarify where these given collaboration tools are located in respect to restricted input versus free input. In general, collaboration has a stricter requirement for freedom, as students are interacting with other humans rather than the system. As discussed in Section 2.4.1, one popular tactic for placing some restriction, or scaffolding, on statements and increasing the recognition potential is through adding a micro-script as some set structure to dialog tools (e.g., sentence starters, dialog phases, etc.) (Dillenbourg & Hong, 2008; Ravenscroft et al., 2010; Sabourin et al., 2012; Scheuer et al., in press; Soller et al., 2001). These structures can provide some information about the dialog without requiring NLU.

However, it should be noted that this structural recognition is subject to error when students improperly use the structure, or the content offered by students contradicts the structural implications (as noted in Section 3.2.1). A simple example of this structural contradiction is a sentence opener stating “I agree with you,” and the student continues by writing “about <point x> but I think you are wrong about <point y>.” A system based on understanding from sentence structures would interpret this statement as agreement when the student is actually introducing a disagreement in a gentle way. For these reasons, we seek to move beyond purely structural recognition, and again use our EKB to better understand collaborative contributions.

3.2.3.1 Implementing Content Recognition in Student Collaboration

In regard to collaboration, Rashi was designed to be a free input system. The chat tool (see Section 2.4.2.1) accepts plain text statements without requiring any other structured input for micro-scripts. However, there are two optional features of the Rashi collaboration system that provide some over-arching structure and also offer a method for the system to understand the content of student work. Both features utilize a concept we term *discussable objects,* which provides the ability to create direct links between pieces of content from the Rashi system (e.g., a student hypothesis, datum, etc.) and collaborative contributions of students.

The first feature that utilizes the concept of discussable objects is the ability for students to start a chat conversation about a specific notebook item (see Section 2.4.2.3). A student can begin a conversation by choosing an item to discuss from a Notebook (either his/hers or someone else’s). That specific chat message is then associated directly with that Notebook item. Additionally, if the Notebook item is associated with a knowledge base element, the chat message can then be directly related to a knowledge base element. For example, a student enters the hypothesis “I think she has mono,” which the system recognizes as “patient has mononucleosis” from the knowledge base. The student then chooses to “Discuss this.” The student types the statement “Does this make sense?” into the chat, and another student responds. Now, the students’ statements in the chat can be linked to the Notebook item “I think she has mono” and to the knowledge base item “patient has mononucleosis.”

The second feature employing the concept of discussable objects is even more direct. The critique-rebuttal system (see Section 2.4.2.3) inherently ties the statements being made (in critique or rebuttal text fields) to a Notebook item. In this way, similar to chat messages, any critiques or rebuttals can be linked to a knowledge base element when the Notebook item is matched.

Of course, students will never be expected to use these features when creating every message. Therefore, there will always be many chat messages that are not linked with Notebook items or knowledge base elements. In these cases, we use the same technique to match these statements to knowledge base elements that we apply to the hypotheses. We send the discussed item to the matching algorithm for suitable matches to the knowledge base content. We have shown this tactic to be successful at reasonably high rates (average of 70% recognition), with higher rates of recognition on cases that had more developed knowledge bases (Dragon et al., 2010). We also have demonstrated that we can increase the confidence of these recognitions through pruning of the knowledge base (Floryan et al., 2012). This evidence supports the ideas that the system can create useful connections between knowledge base elements and students’ collaborative contributions to help better understand the content focus of the students.

3.3 Providing Support

Section 3.2 has introduced both the concept and implementation of how ITSs for ill-defined problem spaces can understand student work. Our ultimate goal of providing an ITS with the means to understand student work is to provide methods by which the system can use this understanding to promote beneficial behaviors and in turn improve students' domain knowledge and learning skills. Students within these open-ended environments with varied learning goals require support (Kirscher et al., 2006).

In our case, we focus specifically on behaviors that relate to inquiry skills. As discussed in Section 2.3.1, researchers across inquiry learning agree on the main tasks involved in the process: creating hypotheses, gathering data, and relating the data with the hypotheses. However, studies have also shown that students learning these skills require support. For example, students tend to consider only a small number of hypotheses, ignore counter-examples presented in data, etc. (Collins & Stevens 1991). To improve student behavior, the system can provide advice that directly addresses such problems, offering support to help students expand the breadth their solutions, or helping to correct errors by highlighting counter-examples. To provide this support, the Rashi system has a *coach,* a component that uses the matching system described above to assess student solutions, and offers feedback that can help students engage in in the inquiry process in a fruitful manner that is clearly linked with relevant domain content.

We now present the various types of feedback the system may offer as related to the different analysis types described in Section 3.2: considering the structure of the argument (3.3.1), the content of the argument (3.3.2), or the inquiry process itself (3.3.3). After enumerating the types of possible feedback, we discuss how the system handles feedback delivery with respect to the uncertain nature of the analysis (3.3.4). Finally we discuss issues of visualization, timing and interruption (3.3.5), addressing the question of how/when feedback should be offered.

3.3.1 Providing Structural Support

We have discussed in Section 3.2.1 that analysis in ill-defined problem spaces is often limited to a structural level, rather than a content level. Rules that define appropriate structures are created and student solutions evaluated to determine if they adhere to these rules. Once this analysis is completed, the system can give students feedback in direct response to the violated rule, presenting the student with a statement about how to remedy the problem and adhere to the underlying theoretical principle of the rule. Several major systems implement this type of feedback (Baghaei et al., 2007; Paolucciet al., 1996; Pinkwart et al., 2007; Scheuer et al., 2009).

3.3.1.1 Implementing Structural Support

Section 3.2.1 presents the structural analysis within the Rashi system that can recognize fairly simplistic situations in which the system might provide feedback to the student. These situations often denote that the structure of a student solution is clearly lacking in some way, for example if student solutions lack certain types of contributions (hypotheses, data, relationships, etc.). A coaching system could merely point out the insufficiency. However, even in these straightforward cases, a coach utilizing a knowledge base can provide improved feedback. After consideration and testing, we narrowed the list of structural feedback the Rashi coach offers to three types. Below we list each type and identify how this type of feedback is enhanced by use of the knowledge base:

- **Lack of Data –**The student is advised that he/she has collected an insufficient amount of data, and should consider collecting more. *Knowledge base enhancement:* Upon request, the student can be brought to a location where data salient to the task at hand will be highlighted to be collected.
- **Lack of Hypotheses** – The student is advised that he/she is considering an insufficient number of hypotheses. *Knowledge base enhancement:* Upon request the student is provided with a list of possible hypotheses to be considered. The student is allowed to directly enter a given hypothesis into his/her argument editor if he/she chooses to do so.
- **Lack of Relationships** – The student is advised that he/she has created an insufficient number of relationships. *Knowledge base enhancement:* Data that are related to the case at hand can be highlighted as important data to relate with hypotheses.

We note the problematic ambiguity in the terminology “insufficient number” when defining these types of feedback. This ambiguity again highlights an issue with a purely structural approach to analysis. If based purely on structural information about a student solution, the system would have no case or domain level definition of how many hypotheses a student should be considering, how much data he/she should collect, or how many relationships he/she should be creating. This requires extra support from the SMEs, which can take several forms. First and simplest, the SME makes informed decisions on a case-by-case basis about these values. For example, the SME might consider that for a given case, considering three hypotheses is sufficient.

When the expert knowledge base is available, this type of decision can be defined in a more scalable way. Rather than giving set numbers, SMEs can decide on the relative percentage of expert knowledge that should be expressed by the students. In this way, as the knowledge base is iteratively improved, these numbers maintain their meaning. Rashi allows this number to be defined in either form, through a percentage or through a set number.

In pilot testing, we found that a dual approach was most successful, having a set number of three hypotheses required, and having a setting of 50% of EKB in terms of data collection and relationships. That is, a student should supply at least three hypotheses, and 50% of the data and relationships currently in the EKB. This configuration fit the SMEs expectations in two cases, and therefore we use this as a baseline, although it can be altered when authors consider it necessary.

Also in reference to structural support, we should recognize implementation decisions within Rashi that make certain types of structural feedback unnecessary. The interface of the Argument Editor (see Section 2.3.2.2) disallows certain relationships to be created. For example, the system does not allow data to support other data elements, since data should be the lowest-level observable facts and require no support or refutation. The system also prevents a student from creating a circular argument, meaning that a student cannot make this logical error using the tool. These preventative measures within the structure of the system itself can be viewed as restricting input, or as providing scaffolding that helps students understand how arguments should be formed.

3.3.2 Providing Content Support

Moving beyond structure, the Rashi coach offers more in-depth and specific feedback by examining the student solution as related to the knowledge base. The underlying principle is to encourage students to offer similar content and structure to that of the EKB. In other words, the way an expert describes the situation is the way we would like to students to learn to describe it (Collins, & Stevens 1991). Once we understand the content of the student argument in reference to the EKB (Section 3.2.2), several methods of intervention from inquiry literature can be used.

There are many examples of how a coaching system that relates a student solution to an EKB can offer feedback to promote inquiry behavior. The coach can encourage the student to provide alternative hypotheses, and offer hypotheses that the student does not already have. The knowledge base can also provide a way for the coach to give counter-examples for specific arguments, by finding a negative path in the EKB (a path that ends in a negative link) (Paolucci et al., 1996). The coach may also lead the student down this same path step-by-step to provide a method of following consequences to contradiction. The coach can prevent students from skipping critical steps by comparing a path in the student solution with a path in the EKB to identify missing nodes in the student solution, and can even present feedback when there are links in the student solution that are not present in the EKB (Suebnukarn & Haddawy, 2004). Through these types of actions, the coach starts to take on many of the crucial roles of a teacher pursuing the inquiry method: stimulating thought on alternative hypotheses, offering counter-examples, and correcting omissions and errors.

3.3.2.1 Implementing Content Support

The first step to producing content support within Rashi is for the coach to recognize the student solution in reference to the EKB. This is accomplished as described in Section 3.2.2; the system attempts to match each student hypothesis to an EKB element. When successful, the system then considers each hypothesis the student has stated, as well as any hypotheses for which the student has collected supporting or refuting data. It compares the relationships and data associated with the student's hypothesis to that of the EKB element. Then it formulates the following types of feedback:

- **Lack of Support or Refutation** –The student is advised that he/she should have more supporting or refuting evidence for a specific hypothesis that he/she has created. Upon request, the coach highlights the specific hypothesis in need of support or refutation, and finally the coach brings the student to the location of this supporting or refuting data in a data collection tool.
- **Missing Hypothesis** – The student is advised that he/she has all the necessary support for a hypothesis; however, the hypothesis was not added to the Argument Editor. Upon request, the coach provides the student with a list of possible hypotheses that includes the hypothesis for which he/she has support.
- **Missing Relationship** – The student is advised that he/she has collected data that is in fact related to a hypothesis he/she has created, but that he/she has not related this data with this hypothesis. Upon request, the coach highlights the hypothesis and data that should be related.

- **Wrong Relationship** – The student is advised of a relationship that is set up incorrectly (i.e., has been set to the wrong type). Upon request, the coach provides the correct relationship type between propositions in question.

By engaging in this feedback iteratively, the coaching system leads the students to create solutions similar to the expert solution, creating similar hypotheses, collecting and relating the data associated with those hypotheses. In this manner, Rashi could encourage students to recreate the expert solutions exactly. There are two issues that arise with this general approach. First, students should not be required to reproduce the entirety of the expert solution. Second, using such a coach, students could exploit the coach to reveal the answer to the problem without truly understanding the material or approach, an issue termed "gaming the system" (Baker et al. 2008).

The issue of requiring a student to enter the entire knowledge base contents in order to appease the coach was a serious concern in our early designs. Our knowledge bases are large enough that we found no real cases where teachers expected or desired that students collect *all* of the data, or consider *every* hypothesis present in the knowledge base. Experts often include repetitive data in order to create a complete representation, yet they do not want the student to necessarily be forced through the repetition as well. To avoid this problem, Rashi takes a unique approach that pushes the student to encapsulate a critical mass of the expert argument without requiring the entire argument or specified subset. This pruning is accomplished by adding extra knowledge to each proposition, indicating the quantity of support or refutation required. The author specifies for each node in the EKB how many of its children are necessary to make a reasonable argument about that statement. This can be considered similar to the approach we discussed on a macro-level to requiring a certain amount of data from the overall EKB (in Section 3.3.1). In this case, rather than specifying how much data is necessary for the entire solution, we are specifying the amount of data required for a specific hypothesis.

As to the second issue, gaming the system, we found this to be less of a concern, as we have not seen examples of this "gaming" in classroom experience. This is partially due to design considerations of the feedback itself. Each piece of feedback provides only one very small piece of the solution, and the student would have to iterate through dozens of pieces of feedback in order to create something similar to the expert argument. System feedback requires effort on the part the student, and therefore, in pilot testing, students actually tended to engage in the processes themselves after several uses of the coach, rather than continuing to use the coach for next steps. This is actually evidence that the coach was teaching some process skills, discussed in Section 3.3.3.

3.3.3 Providing Process Support

The final type of support to consider is the support the coach offers in learning meta-level, or process, skills. Inquiry systems are designed not only to teach students domain content more deeply or in context, but also to teach the process of inquiry investigation itself and to teach students *how* to learn (see Section 2.3). The Rashi coaching system does promote these higher-order learning objectives by teaching the inquiry process, but it does not explicitly instruct about the process with individual feedback messages. Rather, the coaching system supports these higher-level skills through the overall coaching experience, the flow and ordering of feedback offerings.

The higher-order skill we seek to teach is the inquiry process: gathering data, forming hypotheses and connecting the data to support or refute hypotheses (see Section 2.3). The coach encourages this behavior through the specific ordering of successive feedback messages. As the coach gives successive feedback messages, it leads students through the major phases of inquiry. If students start from a blank notebook, the system first encourages data collection (Lack of Data Feedback), then encourages hypothesis generation (Lack of Hypothesis Feedback), and finally encourages students to connect the hypotheses and data they have created/collected (Lack of Relationship Feedback). Once they have items in their notebook, the system encourages refinement of these items (Wrong Relationship, Missing Relationship, Missing Hypothesis, and Missing Support or Refutation Feedback). The ordering of this feedback is based on both inquiry learning theory, which states a general flow of activity, as well as basic learning theory on feedback, e.g., correcting errors before omissions (Collins & Stevens 1991, Shute & Glaser 1990). If a student is stuck, lost, or confused, the system implicitly teaches the inquiry process through the successive steps it offers as advice to the student. Meanwhile, each given step also highlights domain knowledge, creating an overall system that teaches both content and process.

3.3.4 Providing Support using Uncertain Assessment

Dealing with uncertainty is a classic problem in the ITS field (Jameson, 1995). The standard conceptualization of the uncertainty in well-defined domains deals with understanding students' true state of knowledge solely through their actions within the ITS. For example, many researchers use Bayes nets to estimate student mastery using their answers to tutoring questions as input (Gamboa & Fred, 2002; Jameson, 1995). However, a higher level of uncertainty exists when operating in ill-defined problem spaces (as presented in Section 2.2) and free input systems (as presented in Section 3.2.2.2). With these systems, we cannot be sure whether student input is correctly identified, let alone be certain about how this input represents their knowledge. Even with the most carefully defined knowledge bases, and the most cutting-edge NLU techniques, the system might be unable to identify student work, or to identify the student work correctly.

Therefore, as this problem is not likely to be solved in the foreseeable future, this uncertainty must be a factor in the support process, and one must offer feedback with careful consideration. This uncertainty forces us to limit and cater the type of feedback the system offers. We consider two aspects to be essential when offering feedback in a system that incorporates an uncertain understanding of student input:

- Do not comment directly on "incorrect" work.
- Offer feedback that might be useful even if analysis is mistaken.

The first aspect stresses that system should rarely, if ever, tell a student they are wrong. The second aspect reinforces the idea the system should "do no harm," if and when it incorrectly recognizes student work. To address both of these aspects, a useful feedback approach is to bring the points that are thought to be wrong into question, and call the student to re-analyze these points. This approach addresses the first concern by not explicitly correcting students, allowing for a situation where the system has misidentified some portion of the student work. Addressing the second concern, there is no explicit harm in reviewing work in general. In fact the self-explanation that might occur from this review has been shown to be beneficial in many situations (Aleven & Koedinger 2002; Conati & Vanlehn 2000). Overall, from a pedagogical standpoint, this type of feedback supports a more critical thinking approach, allowing students to explore their errors on their own rather than being told to take specified steps. In this way, the system can offer feedback that is essential when correctly recognized, and potentially useful, or at worst redundant, when recognition is incorrect.

To frame feedback in this manner, we consider the coaching system through the lens of identifying teaching opportunities and supporting students when these teaching opportunities occur, rather than a system that 'corrects mistakes.' In this way, the coach does not tell students what to do, but rather encourages students to continue in certain directions that experts consider fruitful. We see how these theories are implemented in the following section (Section 3.3.5).

3.3.5 Feedback Display: Interruption and Visualization

The questions of how much help to offer, and when to offer help, are examples of classic ITS concerns. Termed "the Assistance Dilemma" by (Koedinger & Aleven, 2007), we can view this problem on a pedagogical level. We consider a spectrum with little or delayed assistance on one end and plentiful or immediate assistance on the other end. Immediate and plentiful instruction has the benefit of preventing stuck states or floundering, and reducing cognitive load (Kirschner et al., 2006), but can potentially lead to shallow knowledge gain or "gaming" behavior (Baker et al., 2008). Systems that provide less or delayed assistance offer the other side of the tradeoff, possibly leaving students to flounder but potentially then allowing them to work through their confusion and issues themselves without excess support thus promoting deeper learning.

Narrowing the focus to ill-defined problem spaces and inquiry learning approaches, educators tend towards limited and delayed feedback. The pedagogical idea is to help students learn how to think through problems and learn alternative approaches on their own, when guidance is not available. At the far end of this spectrum is pure constructionism (Harel & Papert 1991), where the working theory is that no guidance should be given, and the only true learning comes from constructing knowledge on one's own.

We must consider not only 'how much feedback' to give, but also 'when to offer feedback.' Many systems provide feedback messages in list format (Gutierrez-Santos et al., 2010; Scheuer et al., 2009). This means that feedback messages can build up over time, and a system must decide on the timing and level of interruption of the feedback relative to the importance of the message. These same systems tend to take a simplistic answer to this question using the concept of on-demand coaching (Gutierrez-Santos et al., 2010; Scheuer et al., 2009). The idea is that coaching is only offered when students request help. This decision is made not only to simplify the problem, but also because help-seeking (knowing when and how to seek help) is a meta-level learning skill (Newman 1994) that students should acquire.

3.3.5.1 Implementing Feedback Display

The Rashi system's feedback system was originally designed with the tenets of non-interruption and on-demand help-seeking in mind. However, several issues were recognized with this original design, and the interface of this feedback has undergone several iterations to make the software easier and more likely to be used. We present the first implementation, the On-Demand Coach, and how it motivated effort to create the second implementation, the Suggested Links Sidebar.

Students activate the On-Demand Coach by choosing "coach" from the "help" menu on their notebook. Once activated, students are shown a coach (a separate window) that presents series of feedback statements (Figures 3.3 and 3.4, bottom). This Coach presents different types of structural and content-based feedback and in the order specified in Section 3.3.3. The user can iterate through the list to view different pieces of feedback. Each piece of feedback has several steps. First, is the general *identification phase*, where the coach points out a high-level potential issue and requests that the student either solves the issue his/herself, or asks for more assistance. If more help is requested, this feedback moves to the *specification phase*, where the system offers details about the current piece of feedback, including highlighting the entries in the notebook that are related to the problem at hand. Finally, if the student requests more assistance, the system enters the *solution phase*, stating explicitly the next step the student should take to solve the problem. This last phase could be considered the "Bottom-out hint," a term meaning a hint that gives the student at least a partial solution to the problem.

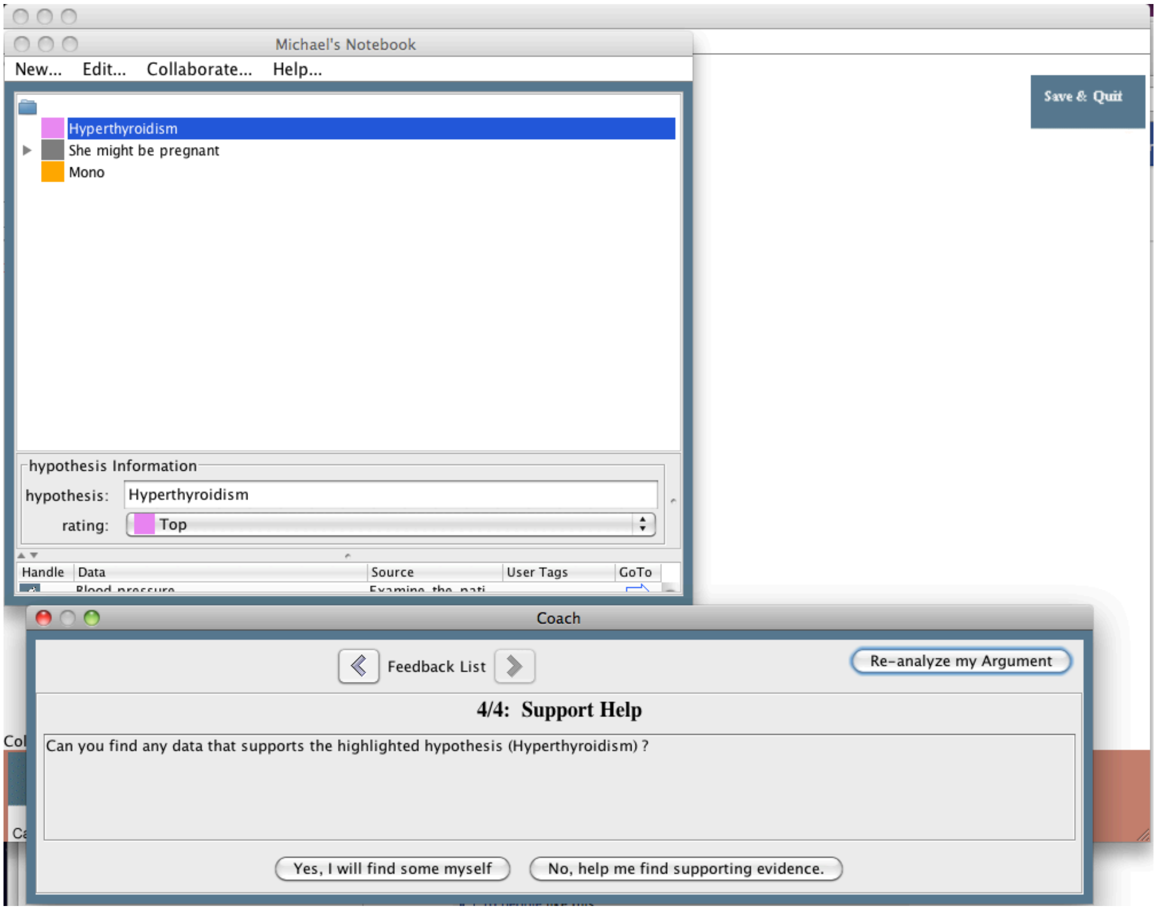

Figure 3.3. The On-Demand Prompting Coach in the *specification phase*. The system highlights a particular hypothesis, "hyperthyroidism," in the Notebook (top), and asks the student to find more support for this hypothesis in the coaching window (bottom).

We found issues with deploying the On-Demand Prompting Coach, as students rarely used the prompts in practice. Testing resulted in few students actually conferring with the coach, even when encouraged by their instructors (see Section 4.5.2 for further detail). Along with pedagogical changes (as suggested in Section 5.3.1) to encourage student use of the coach, we also considered technical aspects that could make the coach both more readily available and less complicated to use.

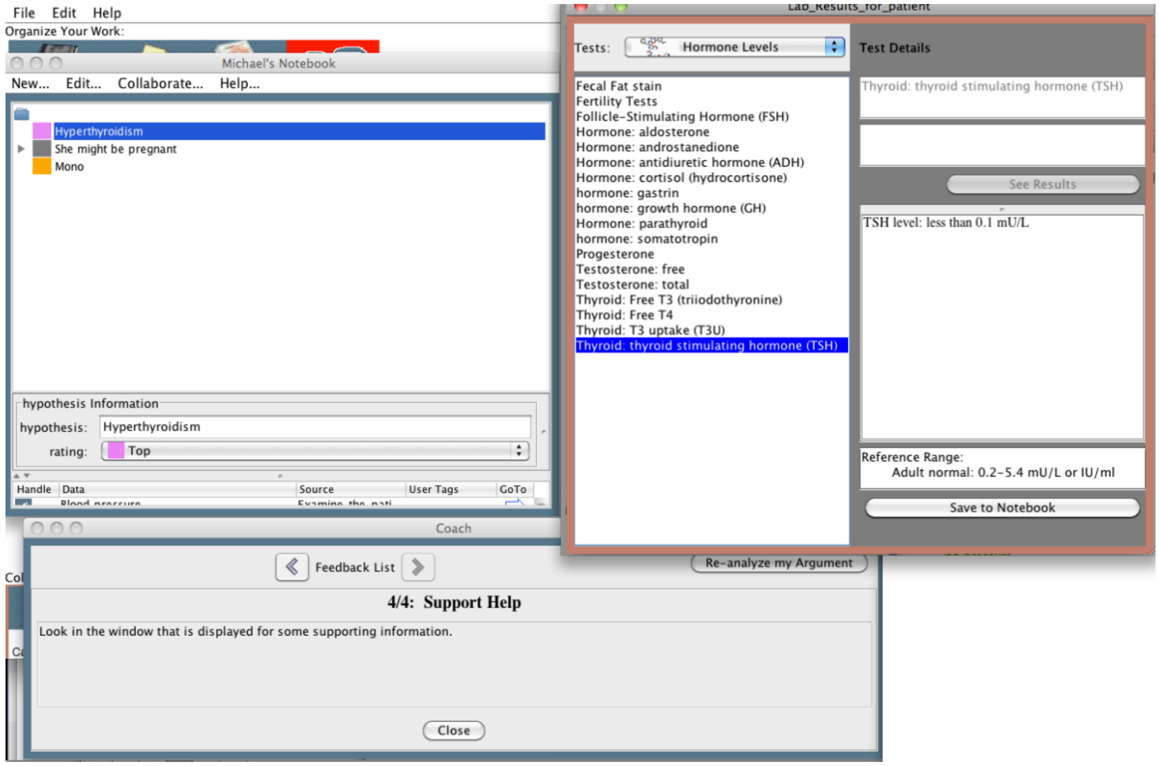

Figure 3.4: The On-Demand Prompting Coach in the *solution phase*. The coach system opens the Medical Lab environment (top-right), and highlights the supporting information while the coaching window (bottom) tells the student to look at this supporting information.

With these results in mind, the interface for the coach was redefined, attempting to maintain the non-invasive nature, but improve the chances of the coach being seen and used more regularly. From this process, we created the Suggested Links Coach. This interface enhances the Notebook interface by adding a separate pane that is present continually throughout the learning process. The coaching system auto-populates this pane with feedback as items are highlighted in the notebook, using the same analysis methods and feedback types as the On-Demand Prompting Coach. This implementation still does not interrupt students' work, allowing students to decide if/when to use the help. Yet students are constantly aware of the suggestions, making feedback more obvious and easily available. This system was pilot tested with small groups of students and students made greater use of the feedback. Some SMEs did express concern with the increased possibility of gaming the system with the increased availability of the EKB elements, but thus far we still have not seen such problems.

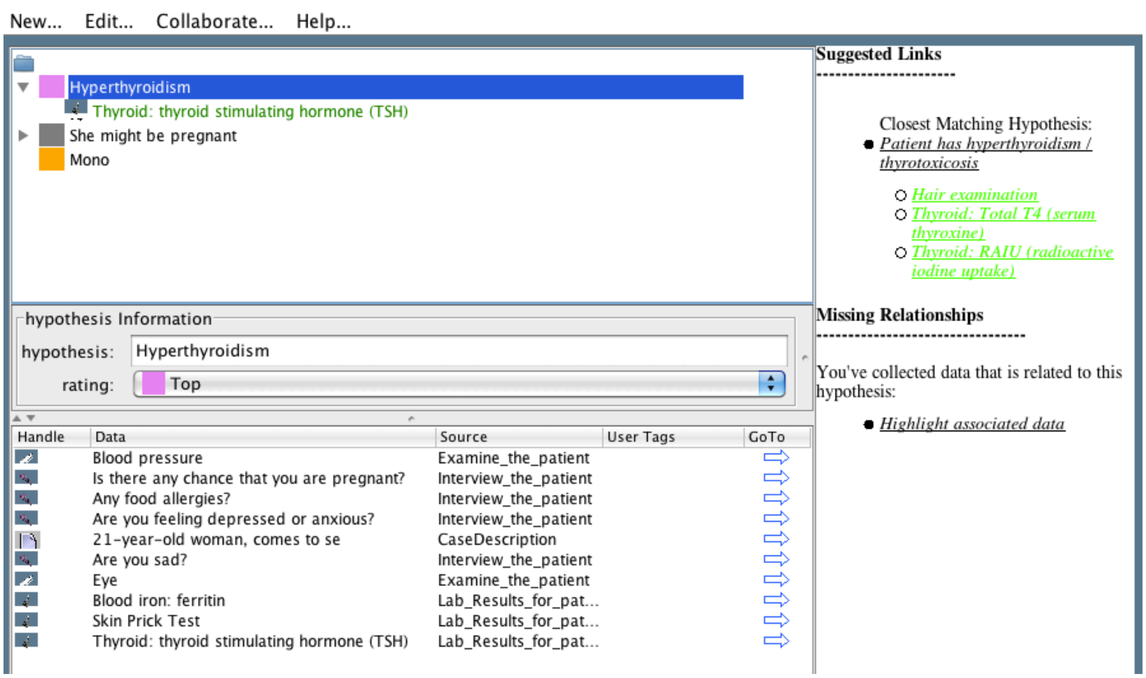

Figure 3.5. The Suggested Links Coach integrated with the Notebook. The user has highlighted "hyperthyroidism" in their Argument Editor (top-left), and the Suggested Links Coach (right) displays knowledge base elements that are associated with this hypothesis. Green highlighted links are supporting data elements, and the user can click these links to be shown the data in a data collection tool.

3.4 Providing Content-Focused Collaborative Support

Thus far we have only considered intelligent feedback on an individual level. In Section 3.2.3, we discussed that the addition of collaborative features provides new opportunities for content recognition and assessment. These same additions offer new opportunities for feedback and intervention. Many researchers attempt to take advantage of collaborative opportunities, with most research applying standard CSCL approaches to more open-ended environments. Magnisalis et al., (2011) provide a comprehensive overview of this work. These approaches focus on discovering how group interactions can be analyzed and improved, and often do not usually consider the content of students' work.

Several of the previously discussed systems for ill-defined domains have been extended to include collaboration, and subsequently researchers have also extended their analysis and feedback techniques to consider the students' collaborative efforts. Baghaei et al. have extended their system Collect-UML to support collaborative efforts (Baghaei et al., 2007). In a similar effort, Constantino-Gonzales et al. present the COLER system, which provides collaborative space for argumentation (Constantino-Gonzales et al., 2003). In both of these systems, collaborative support is offered to improve students' collaborative behavior. In other words, these systems do not use domain level knowledge about the students' arguments when considering collaborative feedback. Rather, these systems consider the facts about the collaboration itself (balanced contributions, even distribution of work, etc.) (Baghaei et al., 2007; Constantino-Gonzales et al., 2003). These systems monitor and instruct students on how to collaborate more effectively, but do not consider the understanding of the individual's recognized content from an expert model to foster or promote collaboration. A limitation of these approaches is that the coaching offered is not domain specific. These systems analyze and support the actual act of collaborating, and teach collaboration skills rather than focusing on learning skills associated with the task, such as inquiry skills or deep understanding of content knowledge. We suggest a more tight integration between collaboration and coaching. Magnisalis et al. phrase this concept as "collaborating to learn," as opposed to the coaching systems just discussed, where students were "learning to collaborate" (Magnisalis et al., 2011).

Specifically, two possible ways to integrate coaching and collaboration to promote "collaboration to learn" are to 1) *use collaborative contributions as coaching opportunities* by finding opportunities to introduce content support when student are collaborating, and 2) *use coaching as opportunity for collaboration* by finding opportunity to encourage targeted dialog based on the content of the students' solutions. Section 4.5.4 presents results demonstrating potential for both of these possibilities within our system. We now discuss the each concept separately.

3.4.1 Using Collaborative Contributions as Coaching Opportunities

As described in Section 3.2.3, several methods exist in Rashi to find connections between an EKB and student's collaborative contributions. Whether students offer their own connections by setting the subject of chat conversations or the system automatically matches the content of chat messages to knowledge base elements, the system can potentially recognize the content of chat messages. We suggest that this recognition presents an opportunity to offer additional coaching in reference to the chat messages rather than in reference to Notebook items.

This additional coaching is presented in Rashi by using the Suggested Links Coach as described in Section 3.3.5. The coach considers the hypothesis or datum being discussed in the Chat tool as well as the content of the student's Notebook to offer content feedback similar to that presented in Section 3.3.2. If a hypothesis is being discussed, the coach can offer Lack of Support or Refutation feedback, which offers links to supporting or refuting information about the hypothesis being discussed. If a datum is being discussed, the coach can offer Missing Hypothesis feedback, which offers links to hypotheses that are supported by the data being discussed but that the student has not yet entered into their Notebook. These suggested links are displayed to the right of the chat window (see Figure 3.6). In this way, we utilize the student chat as a new area to offer coaching suggestions, which can both prompt further conversation as well as help to focus that conversation around pertinent content.

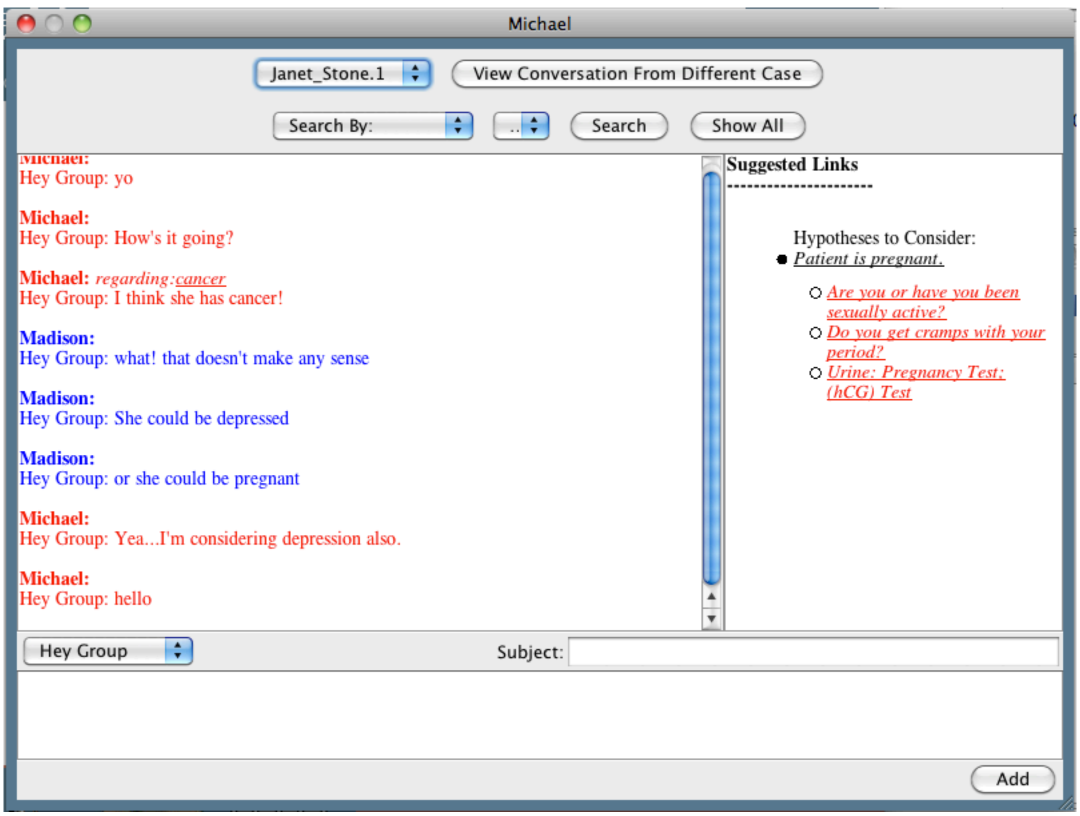

Figure 3.6. The Suggested Links Coach integrated with the Chat tool. One student (Madison) has mentioned "pregnant" in the chat (left), about which the system has much refuting evidence. The Suggested Links Coach (right) displays the knowledge base element about pregnancy, and links to the associated refuting data in red. The student can click these links to be shown the data in a data collection tool.

3.4.2 Using Coaching as an Opportunity for Collaboration

Using coaching to promote beneficial collaboration is a complex issue, but one that presents open potential in the field (Magnisalis et al., 2011). Researchers need to consider in a holistic sense both the function of the coach and the type of collaboration in which students should engage. On a high-level, we would like coaching to help students focus on appropriate content, and engage in inquiry activities.

When considering how to use coaching to support students, we can consider the range of software that pairs students for peer reviewing. In CSCL research, many such systems have been developed to identify appropriate pairs of students by different criteria (Christodoupoulos & Papanikolaou, 2007; Crespo et al., 2005; Garcia & Pardo, 2010; Pérez-Sanagustín, 2009). The basic idea across these different projects is to find automated ways of identifying pairs of students that will work successfully together. However, in these scenarios, domain content is not considered, only traits of students (Christodoupoulos & Papanikolaou, 2007; Crespo et al., 2005), and sometimes only practical stipulations of classroom groupings (Pérez-Sanagustín, 2009). We have a significant advantage in this light, because we can find just-in-time pairings based on the current focus of students.

In this light, we propose that coach's assessment capabilities can be used as a mechanism to prompt the same type of content-focused collaboration that our tools inherently promote, as discussed in in Section 2.4.2.3.

The coach matches content from a student solution to an EKB to offer evaluation or assessment of each student's work. This can be used to identify strengths and weaknesses of each student's solution. By comparing these assessments, the coaching system can identify pairs of students that might have reason for, or need to, interact with one another. These students can then be encouraged to collaborate for short, focused interactions about the specific topic, potentially fostering crucial dialog. We see this situation as promoting content-focused collaboratio*n*. Students are encouraged to focus their interaction around specific discussable objects that are pertinent to their current efforts. One simple example of such a situation where content-focused collaboration can be fostered is when the coach identifies a student that needs help, and, rather than offer them canned statements, the coach fosters a conversation between this student and a peer about the topic identified as needing support.

Other work in well-defined domains has shown the promise of fostering collaboration and peer-tutoring with intelligent identification of topic for discussion. Walker et al. demonstrate that this type of tutoring interaction is beneficial for both tutor and tutee (Walker, et al., 2008). Other current projects in ill-defined domains have attempted similar tactics of identifying key times for collaboration (Dragon et al., 2012), and using automated feedback to assist peer tutors (Ashley & Goldin 2011).

The idea of content-focused collaboration goes beyond this single example. We propose that developers can enhance the intelligence of their ITS and CSCL systems with collaborative tools that allow peers to have more fruitful collaborations, pairing students that are likely to be able to support one another and providing intelligent analysis of the student work that can help students give precise and helpful feedback to one another. A wide variety of situations exist in which content focused collaboration could be beneficial (Crespo et al., 2005; Garcia & Pardo, 2010; Pérez-Sanagustín, 2009). We consider the following situations as prime targets for a coaching system to encourage content-focused collaboration:

- **Same Topic**: When peers are working on the same subject matter, they can be brought together to discuss progress, difficulties, or solutions related to the topic.
- **Contrasting Opinions**: When peers have opposing solutions / conclusions to the same or similar problems, they can be brought together to discuss differences and evaluate opposing arguments.
- **Similar Errors**: When peers have similar errors or inefficiencies in their solution, they can both be presented with corrective feedback, and can be encouraged to discuss the feedback and potential solutions.
- **Tutoring**: When one student has a recognized correct solution and another student does not or has demonstrated signs of struggle, the two students can be brought together. The student with a correct solution can be presented with an assessment of the other student's potential error. The first student can then act as a peer-tutor and offer advice or support, guided by the system's automated assessment.

3.4.2.1 Implementing Content-Focused Collaborative Support

We developed two prototype functionalities to promote content-focused collaborative support in the Rashi system. Both utilize the coach's knowledge base assessment to recognize aspects of student solutions, and then suggest collaborations when coaching is requested. The two situations identified within Rashi are instances of the *Same Topic* and the *Tutoring* as listed above.

To implement Same Topic content-focused collaboration, the coach examines the current hypotheses on which a student is working, and then compares this assessment with other students to identify situations where other students have created similar hypotheses, or have collected data related to this hypothesis. Rashi then encourages chat between the students highlighting the related hypothesis and data.

To implement Tutoring content-focused collaboration, the coach again examines a student's hypotheses, but this time looking for hypotheses that the student has collected and connected sufficient data to be considered "sufficient" by the knowledge base evaluation. When this is true, the current student is a candidate for offering peer tutoring. In this case, the coach looks for other students who have the same hypothesis, but are missing data or relationships. The coach then asks the proficient student to use the "critique" functionality (Section 2.4.2) to offer advice to their peers to complete work on the hypothesis in question.

3.5 Summary

This chapter presents the rationale for, and the methods with which, an ITS can understand and support students during collaborative inquiry learning in ill-defined problem spaces. We focus specifically on how student work can be automatically recognized and understood in such systems using artificial intelligence techniques. We then present how this understanding gained through artificial intelligence techniques can be used to provide support to students on the structure and content of their solutions, as well as the process by which they create solutions. Finally, we discuss how this recognition and support can be extended beyond the individual learner to include collaborative work. We consider both how collaboration can provide additional opportunities to present content-related intelligent support, as well as how intelligent support can utilize collaboration as a means of intervention, identifying key points where collaboration is seemingly necessary. Throughout the chapter we present the specific example of our research on the ITS Rashi. We discuss how this system recognizes student input using an EKB, and provides an automated coach that offers these various types of feedback.

The first step for an ITS to provide intelligent feedback is for the system to understand student input in order to cater the learning experience to the specific student. We recognize that this understanding is more difficult in ill-defined problem spaces, where there are not necessarily correct solutions or solution paths, and also difficult in inquiry learning and collaborative environments where the input devices available to students create vast solution spaces. Some recognition of student work can be accomplished purely through recognizing the structure of student solutions, and we present examples from the Rashi system of how this can be accomplished. However, there are weaknesses to this approach that make it impractical as a full-fledged solution to understanding student input.

Moving beyond simple structural recognition, a system must somehow recognize the content of student contributions. To recognize the content of a student solution, we must first decide how students will input content, on a spectrum of free input to restricted input. When input is restricted it may be more easily recognized, but these restrictions can hinder the creativity and the active role of learners in the inquiry process. We present a hybrid method of understanding the student work within the Rashi system. Most data used by students is already represented within the knowledge base, and so data collection is generally easily recognized in reference to the knowledge base. Hypotheses are entered as free text and need to be recognized in some fashion. We create an expert knowledge base to represent the domain knowledge for a given topic; including expert hypotheses linked to the data present in the system. Finally, we use simple keyword matching techniques to link student entries to expert knowledge base elements in order to offer evaluation or assessment of the student work. We use similar tactics to recognize the content of chat messages. Employing this overall strategy, as we do in the Rashi system, one can gain an overall understanding of both student solutions and the content of students' collaborative efforts.

Once an ITS understands the student, the system should provide support catered directly to the situation at hand. The different methods of understanding work lend themselves to different types of feedback. Content recognition can provide more in-depth and specific feedback, which is preferable in many cases.

Many concerns arise when actually providing the feedback, such as how to handle uncertainty in the understanding of the student, when to interrupt to present feedback, and how to visualize the feedback. In inquiry learning systems in general, as in our case with Rashi, researchers tend towards conservative answers to these questions. They provide feedback that is not harmful if the system misunderstood student input, and time and present feedback in ways that limit the interruption of student work flow. We present examples of how the Rashi system implements feedback, and how it has been refined with these tenets in mind.

Finally, we present methods of combining automated support and collaboration. Although this is a complicated issue, it has the potential to improve both coaching and collaboration, promoting what we term as content-focused collaboration. We see how collaborative efforts can be used as additional opportunities for domain-related support. Most importantly, we discuss how collaborative elements can be harnessed as additional means of intervention for support. This is accomplished by recognizing certain situations where focused collaboration between two students could be highly productive based on the content of their solutions. We present different contexts in which this focused collaboration could be productive, and discuss how the Rashi system has been adapted to recognize and encourage such situations.

Overall, this chapter presents a holistic picture of utilizing artificial intelligence techniques, particularly expert knowledge bases, to provide automated support in the collaborative, inquiry learning environments for ill-defined problem spaces described in Chapter 2. Chapter 4 presents the evaluation of these tactics as related to the Rashi system, and its use in real classrooms.

CHAPTER 4

EVALUATION OF COACHING AND COLLABORATIVE FEATURES IN AN INQUIRY LEARNING ENVIRONMENT

4.1 Introduction

The overall purpose of this research is to offer an understanding of how both collaboration and coaching can improve student behavior within an inquiry learning system: *Does targeted use of both collaborative features and feedback from coaching software improve students' solutions?* Four hypotheses are tested by using data from studies conducted in actual classrooms.

These studies were carried out over a duration of four years (2007-2010). The development team used iterative design and implementation to introduce major updates to improve the usefulness and the usability of the system. Therefore, the data were collected over several years in which the system functionality and student populations varied. Because of this shifting system functionality and student population, combined with differences in data recording and analysis techniques, one unified dataset doesn't exist on which to test these hypotheses. Instead, different data sets were used to test four distinct hypotheses. The different scenarios and settings where each dataset was collected are described in the analysis of each hypothesis.

Section 4.2 describes the four hypotheses tested and a brief summary of results. Section 4.3 describes the classroom studies including target populations and demographics, classroom settings, and the various system features available for each study. Section 4.4 describes the methods for analyzing whether the data supports or refutes the hypotheses, describing in detail how we utilize data to support the given hypotheses. Section 4.5 examines the evaluation of each of the hypotheses, discusses the experimental results and implications of these results.

4.2 Hypotheses

This research focuses on two factors: *collaboration* and *coaching*. The research addresses individual hypotheses about each factor independently and combined. However, a third factor that is important enough to be considered in its own right, is the pedagogical approach within the classroom. This factor is considered explicitly controlled for in one of the studies. This made it possible to test the effects of defining and communicating the classroom pedagogy more clearly. This thesis addresses four hypotheses. This section reviews each hypothesis including the rationale for developing each, the experimental design used to test them, and a summary of the results.

4.2.1 Collaboration

H1: The addition of collaborative features improves student inquiry behavior, increasing the size and complexity of student arguments.

Collaboration was introduced into the Rashi system partly to address a major weakness recognized in student work by the teachers and researchers in early studies: namely failed to explicate their work. Students using Rashi were interacting with the system but not all were systematically collecting data, entering hypotheses, or building arguments in the volume that was expected and desired. Teachers reported that they discussed their hypotheses and reasons for rejecting them informally with other students, but once rejected they didn't see any point in entering that information into the system. Thus they didn't have all the data they needed to demonstrate the learning process they had used to others.

We defined an initial benchmark of improved inquiry behavior as an increase in the amount of data students collected and entered into Rashi, the number of hypotheses articulated and entered, and the number of relationships created between data and hypotheses. We found that students who were provided with collaborative features entered significantly more information into the system when they knew it was going to be shared and discussed with other students; they also gathered more data and created more arguments (see Section 4.5.1). The effect was significant, and this provides a solid understanding of the quantitative effects of the collaborative features of the software.

4.2.2 Coaching

H2: The addition of coaching components improves student behavior by helping students focus on essential information and increasing the creation of semantically meaningful and content-rich student solutions.

The coaching system within Rashi provides structural, content, and process support for students (see Section 3.3). To measure the effects of coaching, we look beyond the concept of a purely quantitative evaluation to consider qualitative measures as well. Automated qualitative evaluation in ill-defined problem spaces is difficult to measure, due to both the ill-defined nature of the problem space (Section 2.2) and the issues of understanding student work (see Section 3.2). We attempt to overcome these challenges by utilizing an automated system that understands student work through use of an expert knowledge base (EKB) (see Section 3.2.2), and in this way can offer some qualitative measure of student work (see Section 4.4.2).

We did not find direct statistical support for H2 across control and intervention group (see Section 4.3.2), leading to the conclusion that the existence of the coaching system as available during the experimentation did not improve student argument by the given measures. However, we did find results that show a correlation between the amount of coaching a student received and the quality of certain aspects of student solutions. This offers some evidence that as coaching is used more often, we might expect improved student solutions. The limited effects of the coaching component could be due to a number of factors, but is most likely attributable to the small number of interventions per student coupled with the short amount of time students operated within the system after coaching intervention was requested.

4.2.3 Effects of Improvement to Classroom Pedagogy

H3: Clarification of the pedagogical approach with both facilitators and students improves student behavior, increasing the creation of semantically meaningful and content-rich student solutions.

This hypothesis stems from changes to the classroom pedagogy, rather than system functionality changes. We observed continual problems with students' ability to develop large, content-rich arguments, particularly in the classes where students did not normally practice the inquiry method. To address this issue, we developed materials and provided additional information to facilitators both on the inquiry method and how it was realized within Rashi. We theorized that improvement in the pedagogical approach to the classroom would have an effect on the size and content of student solutions. This hypothesis was investigated in one year that brought multiple major developments: one technical (the addition of the coach), and one related to classroom pedagogy (the addition of introductory materials and training presentations). This hypothesis and the datasets used differentiate the effects, clarifying the change brought about by the technical and pedagogical developments, respectively.

We found statistically significant support for this hypothesis (see Section 4.5.3). We conclude that these improvements did encourage students to complete more work within the system (quantitative), and also to create solutions that were more accurate and relevant to the case as judged by our automated analysis (qualitative). We see several explanations for this improvement and offer insight as to how this improvement can be more explicitly taken into the research cycle in the future to make their contribution more explicit.

4.2.4 Combining Coaching and Collaboration

H4: An expert knowledge-based recognition system can identify opportunities to promote targeted content-focused collaboration.

Work on combining the coaching and collaborative abilities of the system is still in progress and so we can offer no formal evaluation of the effects of a complete system that offers integrated content-focused collaboration techniques, although prototypes of such a system have been developed and piloted (see Sections 2.4.2.3 and 3.4). Rather, we measure the potential opportunity to engage students in this content focused collaboration in several ways.

First, we consider whether collaboration can provide additional, relevant, knowledge-based feedback. We find clear evidence that the system can automatically identify the content of student discussion with reasonable accuracy, thereby allowing the system to offer additional and potentially useful feedback using the knowledge base (see Section 4.5.4.1). Second, we seek to understand how the coaching system can use collaboration as an opportunity to provide new types of feedback, specifically feedback promoting content-focused collaboration. In this way, the coach could enable students to benefit uniquely from targeted interaction at key points in the learning process (see Section 3.4). To evaluate this hypothesis, we identify how the analysis system can automatically recognize such opportunities in past data. We found ample opportunities in which a coach could have identified instances where promoting content-focused collaboration would be useful. We thereby conclude a system can identify key moments to encourage students to support each other's learning (see Section 4.5.4.2).

4.3 Classroom Studies

Over 3,000 students have used Rashi in the classroom between 2007 and 2012. These uses have all been in the biology domain and involved either introductory biology students in undergraduate college courses, middle school students using the software during the normal school year, or middle school students participating in a summer camp learning experience. The analysis performed to test the current hypotheses includes over 1,500 students who used the system between 2007 and 2010 at three different institutions.

The first setting of these studies is a small liberal arts college in Western Massachusetts. This college encourages new and innovative methods of instruction, promoting inquiry-based learning and constructive civic and social engagement. The openness to alternative teaching methods and innovation in teaching style provided a positive atmosphere for our early research, and methodologies from their existing classrooms provided guidance about how inquiry methods can be incorporated into current working classrooms. The second setting of these studies is a large, research-focused university in Western Massachusetts. This setting provided a much larger user base than any other setting. The number of the students involved in these studies allows us to test our system and theories on a larger scale, and demonstrate how our system can be used in more standard classroom settings. The last setting is a primary/middle school in Western Massachusetts, providing insight as to how the software might be applicable to a different age group in yet a different classroom setting. The software was well received and used repeatedly in each of these settings, providing evidence of general applicability of the software and the teaching methods.

Over the years of studies within these different populations, groups of students from these different settings have used various versions of the Rashi system. Typically each group used the latest deployment of the software available at the time of experimentation. The change in software between studies ranges from slight user interface modifications and bug fixes to drastic system overhauls. This variation introduces serious issues when considering cross-group analysis from an experimental design perspective. Due to the continual system changes occurring as we were simultaneously evaluating the effects of the system, the research presented here can be viewed as “design-based research” involving “quasi-experimental” evaluation (Brown, 1992; Wang & Hannafin, 2005). Pure “experimental design” does not include confounding variables as described here, and has well-matched control and experimental groups. As presented by Brown (1992), pure experimental design is virtually impossible in real classroom settings, and interesting (although potentially less powerful) results can also be gathered from experiments that have possible confounding variables, allowing for some integration of practice and research in the classroom. While we attempt for the most pure experimental design feasible in our conditions, we recognize that confounding factors involved (e.g., changing student population, software updates, etc.). To minimize the effects of these confounding factors, we analyze different data sets when considering different hypotheses, attempting to identify the data most pertinent to the specific research question at hand. We also choose the specific datasets with the most consistent student populations and the least amount of confounding factors presented by software updates. We will examine 11 data sets representing students from a broad range of experiments run between 2007 and 2010, as presented in Table 4.1.

We now offer a description of each classroom study, detailing both the population of students and the state of the software at the time of use. Finally, we describe how the dataset is used to investigate specific hypotheses.

Table 4.1. Detail of the classroom studies.

Ref .#	Date	Class	Rashi Case	User Accounts	Coaching	Collaboration
1	Fall 2007	Small College Intro. to Biology	Anemia 1, Hyperthyroidism	14	none	none
2	Fall 2008		Anemia 1, Hyperthyroidism	22	none	view/copy
2.1	Fall 2009		Anemia 1, Hyperthyroidism	14	none	View/copy/chat
3	July 2007	Middle school Summer Camp	Anemia 1	37	none	none
4	July 2008		Anemia 2	33	none	View/copy
5	July 2009		Bee Allergy	49	none	View/copy/chat
6	July 2010		Anemia 2	50	Suggested links	View/copy/chat
7	March 2009	Large University Biology 101	Hyperthyroidism	41	none	View/copy/chat
8	Feb. 2010			76	none	View/copy/chat
9	Feb. 2010			194	On-demand prompts	View/copy/chat
10	Aug. 2010			21	Suggested Links	View/copy/chat
11	April 2010	Middle School	Hyperthyroidism	22	Increased On-demand prompts	View/copy/chat

4.3.1 College Introduction to Biology – Fall 2007, 2008

The Introduction to Biology course in the small college where our studies were run is comprised mostly of freshman, but also includes some sophomore undergraduates. The faculty instructing this course also acted as subject matter experts (SMEs) to help develop the knowledge base for the medical cases. These faculty members are familiar with the Rashi system and have actively engaged in the testing process and worked closely with the development team to formulate extensions to the software as well as to initially develop the pedagogical approach. This class is taught with an inquiry-learning based approach, and therefore students engage in inquiry-based, “Rashi-like” activities even when not using a computer system.

Both 2007 and 2008 studies were carried out using the same cases, Anemia 1 and Hyperthyroidism. The system did not have any coaching capabilities. Students used the software as a lab activity for two weeks, with one to two-hour weekly lab meeting times where they completed a large amount of the total Rashi work. However, the system was available online for home use, so students could use it as necessary during this two week period to complete their assignments.

2007 study – This study involved 14 students, each working on a separate computer. The system had no coaching capabilities and no collaborative tools. However, the faculty in the class did promote a certain amount of face-to-face collaboration when students were struggling or stagnant (e.g., students were encouraged to discuss their ideas about the case with one another).

2008 study – This study involved 22 students, each working on a separate computer, but divided into groups of three to four. The only major system enhancements were the development of the collaborative capabilities, namely the ability for students in a small group to view one another's notebooks and to copy specific entries from group members' notebooks (as described in Section 2.4.1). Students still had no chat capabilities and engaged in limited face-to-face collaboration at their own discretion. The system again had no coaching capability.

We analyzed the data across these studies to offer an understanding of how the collaborative capabilities of the system affect student work. Specifically, we tested to see if the addition of these collaborative features increased the size and complexity of student solutions, as discussed in Section 4.5.1.

4.3.2 Middle-School Summer Camp – Summer 2007 - 2010

Rashi was also used as one aspect of a middle-school summer camp held at the same college. This summer camp introduces seventh to ninth grade students to inquiry learning and problem-based approaches to learning science. The students in two studies spent a total of five to six hours working with Rashi over a span of five days within a classroom, where students worked in pairs on computers. The 2007 study involved 33 students using the system without the collaborative features or coaching. The 2008 study involved 37 students with limited collaborative capabilities (as described above), namely the ability for students to view one another's notebooks and to copy specific entries from group members' notebooks (see Section 2.4.1). The system had no coaching capabilities. The 2009 study involved 49 students using the system with full collaborative capabilities, including viewing and sharing notebook contents and chat functionalities, also without coaching capability. Finally, the 2010 study involved 50 students and included the Suggested Links coaching capabilities, as described in Section 3.3.5.

We use the 2007 and 2008 data to test the effect of the collaborative capabilities on student solutions in the same fashion as the 2007-2008 data from the biology course, as discussed in Section 4.5.1. Some confounding variables are involved with the comparison of these two middle-school studies. The student population has more room for fluctuation from year to year since the summer camp invites students from wider a range of communities and schools than does an introductory college biology class. Also, different but relatively comparable Rashi cases were used in the two sessions (Anemia 1 vs. Anemia 2). The 2009 and 2010 data were not used here for any of the given analyses, due to the confounding variables and the differences in logging of data for the Suggested

Links Coaching, which offered limited insight into how coaching might have affected student work.

4.3.3 University – Biology 101, 2009-2010

Biology 101 at the university is an introductory course comprised mostly of freshman students in their second semester of classes, but also includes some sophomores and juniors. Rashi is used as a two to three hour lab exercise and was used during one lab session where students were required to complete their work outside of class time (logging in either from their own machines or from university lab computers). To complete their assignments, students exported their information from Rashi using the Report Editor (see Section 2.3.2.2) and completed a written report to be graded by the teaching assistants (TAs). Due to the limited number of computer terminals, students worked in groups of two or three on a single computer registered as a single user. The course was broken down into sections, each section having between 14 and 20 students and eight computers available. We conducted three studies within this setting, offering four distinct groups for analysis:

March 2009 - Slightly more than 700 students used the Rashi system in this study. Accounting for the groups of two to three, there were 358 total registered users. However, many different cases were used in this experiment, and therefore to be consistent with 2010 data, we use only the data from those 41 users who worked on the Hyperthyroidism case.

In this study, TAs were given a brief introduction on how to log into the system, and a general overview of how to use Rashi (e.g. "ask questions in the interview," "put hypotheses in the notebook," etc.). However, there was little introduction to the classroom pedagogy of Rashi, (the inquiry process), and there was no live demonstration of how the system could be used or what students should gain from using the system.

February 2010 – This study included slightly over 800 students, in groups of three or four, creating a total of 270 registered users. This set of students was divided into a control group of 76 students who used the system without coaching capabilities, and an experimental group of 194 users who used the system with coaching abilities. The set of students was divided into sections, each consisting of 14 to 20 students (as describe above). To randomize the experimental group versus the control group, we randomly assigned coaching capabilities to certain sections of the course. It was necessary to assign entire sections and not individuals to the control and experimental groups because special instructions were necessary to alert students of the presence of the coach and encourage students to use the capabilities.

The February 2010 study included a much more focused presentation on classroom pedagogy from the Rashi team. A Rashi team member met with all the TAs several times before the study to introduce general pedagogical idea of inquiry learning, describe how the system implemented that learning theory with its functionality, and finally discussed how students should ideally be using the system.

August 2010 – This study was run as a university summer course, and so only included 20 students. Each student worked at one computer, a total of 20 registered users. The full capabilities of the coaching system were enabled.

The data from these studies are used to test three hypotheses. First, the 2010 control versus experimental groups are analyzed to understand the effect of coaching. We analyze the data to see if students' solutions are improved with the availability of coaching, see Section 4.5.2. Second, the 2009 group is compared with the 2010 control group to understand the effect of the improved classroom pedagogy, specifically to understand the effects of the explication of pedagogy on the quality of student solutions, see Section 4.5.3. Finally, the 2009 data are also analyzed to understand the potential for combining collaboration and coaching. Specifically, we analyze both student's solutions and their collaborative contributions to identify opportunities where coaching and collaboration could be mutually encouraged and enhanced (see Section 4.5.4).

4.3.4 Middle School – Spring 2010

This study involved 22 students from a seventh grade class. Rashi was used as part of a teacher initiative to integrate instruction on "how to learn" into middle school science curriculum. Rashi was used in the classroom for five periods, each being one and a half hours in length. Each student worked on one computer. Students were randomly assigned to groups of four to five. Rather than completing a report, a de-briefing exercise was carried out with the entire class for one period to complete the study. In this de-briefing, students summarized what they learned and shared their feelings and opinions about the system and how it supported their learning. As this was a small population and was not a comparable setting to other studies, this study was used only to pilot new functionality and better understand how Rashi might perform in different educational settings. The data are not used in the formal evaluations.

4.4 Data Analysis Techniques

Automated analysis of student performance in ill-defined problem spaces is generally a difficult problem. As described Section 3.2, freedom of exploration and particularly free input text systems limit the use of more traditional modeling approaches (Mitrovic & Suraweera, 2001). Therefore, design of the software must include innovative approaches for assessment that harness the power of the limited information available to offer indications about student behavior from the given "solutions" to the task at hand.

We define *student solutions* within the Rashi system to be the contents of their Notebook (see Section 2.3.2). The notebook contents are a fair representation of student effort within Rashi as they represent all of the data collected, the hypotheses created, and the relationships established by the student. We evaluate these solutions in different ways to test different hypotheses. The choice of evaluation metric reflects both the specific focus of the research questions as well as the means of data collection and analysis that were available during the specific studies. The analysis system evaluates student solutions in terms of both quantitative measures (linked largely to structural understanding, see Section 3.2.1), and qualitative measures (linked largely with content recognition, see Section 3.2.2). We now briefly describe methods used for the evaluations offered in Section 4.5.

4.4.1 Quantitative Assessment

To understand students' work, we can first examine purely quantitative data. While such an assessment does not recognize the content of student work (see Section 3.2.1), it has been widely demonstrated that quantitative analyses are useful indicators in assessment, particularly in the community of Computer Supported Collaborative Learning (CSCL) (Kay et al., 2006; Soller et al., 2005). In the Rashi system, we consider many such quantitative indicators to provide insight into student behavior and help assess whether students interact with the system successfully. Below is the set of quantitative indicators used in our evaluations.

Table 4.2: Quantitative indicators to evaluate student solutions in Rashi

	Name	Description
Per student	Data Entries	Pieces of data present in student's notebook.
	Hypotheses	Hypotheses present in student's notebook
	Relationships	Relationships present in student's notebook.
	Arguments	Hypotheses and relationships present in student's notebook.
Per hypothesis	Related Support	Supporting data entries offered by a student for the given hypothesis
	Related Refutation	Supporting data entries offered by a student for the given hypothesis

These quantitative measures were used specifically to test hypothesis H1 (collaboration improves solutions) (Section 4.5.1), as this hypothesis is based on the quantity of work created by the student and not the content of the specific contributions (see Table 4.2). These quantitative measures were also used as a part of the metrics when considering H2 (coaching improves solutions) and H3 (clarification of the classroom pedagogy improves solutions)(see Section 4.5.2 & 4.5.3).

4.4.2 Qualitative Assessment

To offer qualitative assessment, we take advantage of the EKB recognition approaches described in Section 3.2.2. In this section, we described how student solutions could be understood by the system through relations to the EKB. Section 3.3 describes how this recognition can be used to offer feedback. We can also use this as a type of assessment, as it provides a measure of how closely a student's solution relates to the given expert solution. This comparison gives us some objective assessment of student solutions, given that the matching algorithm is reasonably successful (see Section 4.4.2.1). Specifically, we consider several key indicators that can be computed through the knowledge base evaluation that offer insight into the content of student work(see Table 4.3).

Table 4.3: Qualitative indicators to evaluate student solutions in Rashi

	Name	**Description**
Per hypothesis	Support Data	The percentage of data entries that a student has gathered in the notebook that have been designated by an expert to be supportive of the given hypothesis.
	Related Support Data [1]	The percentage of supporting data entries that a student has correctly related in the notebook that have been designated by an expert to be supportive to the given hypothesis.
	Refutation Data	The amount of data entries that a student has gathered in the notebook that have been designated by an expert to be refutation of the given hypothesis.
	Related Refutation Data[2]	A subset of Refutation Data, the amount of refuting data entries that a student has correctly related in the notebook that have been designated by an expert to be refutation of the given hypothesis.
Per student solution	Student Hypothesis Data	The number of hypotheses in the student's notebook that are recognized

[1] The Related Support Data will always be equal or smaller than Supported Data, because data must be first gathered before it can be related the correct hypothesis.
[2] The Related Refutation Data will always be equal or smaller than Refutation Data, because data must be first gathered before it can be related it to the correct hypothesis.

		with matching hypotheses in the expert knowledge base.
	Student Support Data	The addition of all Support Data for all hypotheses present in the student notebook.
	Student Related Support Data	The addition of all Related Support Data for all hypotheses present in the student notebook.
	Student Refutation Data	The addition of all Refutation Data for all hypotheses present in the student notebook.
	Student Related Refutation Data	The addition of all Related Refutation Data for all hypotheses present in the student notebook.

As can be seen in the definitions, the indicators per hypothesis are aggregated to create the indicators per student solution. This aggregation offers five indicators for each student solution, assessing the hypotheses, data, and the relations among them. We consider these measures to indicate a student's focus on essential information, and also identify whether student solutions are content-rich and semantically meaningful. By measuring the overlap between expert solution and student solution in this way, we understand a) some measure of the student's focus on information deemed important by the expert (noted in Support Data and Refutation Scores and Hypothesis Score, Table 4.3), and b) some measure of the correctness of the structure and content of the relationships students establish (noted in Support Data and Related Refutation Scores, Table 4.3).

We also have quantitative indicators discussed in the previous section, e.g., data entry, hypotheses. Together, these quantitative and qualitative indicators offer a somewhat holistic assessment of a student solution. To convert this into a single, comparable assessment metric, we normalize the values of these indicators, and sum over them to create an overall score called the *Solution Score*. We use the Solution Score to test hypotheses H2 and H3, as we judge the effects of both coaching (H2, see Section 4.5.2) and improvements to the classroom pedagogy (H3, see Section 4.5.3) by this overall score metric.

4.4.2.1 Validation of Matching Algorithm

As described in 3.2.2, one critical challenge to recognizing student work is the algorithm used to match student hypotheses with EKB elements. We therefore must manually validate this process in order to ensure the usefulness of our approach, both as a basis for coaching (as described in Chapter 3) and as a basis for assessment (as described in Section 4.4.2). We used all 872 of the hypotheses across all datasets to validate the matching algorithm, as these are the same hypotheses used to evaluate qualitative assessment for hypotheses H2 and H3. A human judge considered each hypothesis along with its respective match to better understand the successes and failures of the matching algorithm. Each hypothesis was marked not only as correct or incorrect, but also according to the following categories:

- Types of **correct** matches for the algorithm:
  - **Successful match** – The matching algorithm correctly identified the best match to the knowledge base.
  - **No match, success was impossible** – The matching algorithm found no match, and indeed there was no appropriate match in the knowledge base.
- Types of **incorrect** matches for the algorithm:
  - **No match, success was possible** – the matching algorithm failed to identify the correct match when the hypothesis was present in the database.

- **Wrong match, success was impossible** – the matching algorithm identified the wrong hypothesis. However, there was no matching hypothesis in the knowledge base.
- **Wrong match, success was possible** - the matching algorithm identified the wrong hypothesis, even though the correct hypothesis was present in the knowledge base.

Overall, the algorithm showed an 82.8 % success rate, as can be seen in the breakdown of the correct and incorrect matches shown in Table 4.4. There are several useful results to note here. First, when considering the use of this matching system in assessment, we can see the algorithm has a reasonably high rate of success in matching items with the existing knowledge base (82.8%). In order to make the assessment for our analyses presented in Section 4.5, only these successfully matched hypotheses are considered for analysis, meaning that we will only be considering 82.8% of our data for analysis, the 82.8% that was correctly identified by the system. The other 17.2% were discarded from consideration in the qualitative analysis, although still used in quantitative analysis, as the qualitative assessment over these incorrectly identified nodes would not be relevant.

Table 4.4: Results of the evaluation of the matching algorithm for hypotheses.

Match Evaluation	Hypothesis Count	Hypothesis percentage	Specific Match Evaluation	Hypothesis Count
Correct Matches	722	82.8 %	Successful match	458
			No match, success impossible	264
Incorrect Matches	150	17.2%	No match, success possible	134
			Wrong match, success impossible	14
			Wrong match, success possible	2

The second aspect of this validation process is to consider whether the coaching can be considered useful when it relies upon the given algorithm. The coach offers assessment only when it has found a match, and so to consider when the coach may give erroneous feedback, we need to consider the percentage of cases where the system has an incorrect match. In this case, that number is 16 (wrong match, success impossible + wrong match, success possible), which represents only 1.8% of the matches. We find this to be well in the range of acceptable error, and consider coaching upon this matching scheme to be useful in that regard.

4.5 Data Analysis Results

In the following section, we use the data analysis techniques and metrics described in Section 4.4 to offer empirical evidence to support or refute the hypotheses described in Section 4.2. After stating the hypothesis, we describe the specific scenario in which the hypothesis was tested and the results of the data analysis.

4.5.1 Effects of Adding Collaboration

H1: The addition of collaborative features improves student inquiry behavior, increasing the size and complexity of student arguments.

We seek evidence to test H1 by comparing data from studies given with and without collaboration features in 2007 and 2008. In 2007, students at the college in both Introduction to Biology (Table 4.1 row, 1) and in the summer camp (Table 4.1, row 3) used Rashi without collaboration features. In 2008, the same classes (with a different population of students) used the Rashi system with specific collaboration features enabled (Table 4.1, rows 2 and 4). Specifically, in the 2008 studies, students were able to view and copy from others' notebooks (see Section 2.4.1). Coaching was not enabled for any these studies.

To operationalize H1, we use only quantitative measures (see Section 4.4.1), as we were collecting and assessing this data before qualitative measures using the EKB were possible. Specifically, to assess student solutions we consider the key variable of data entries, as this is a solid, low-level indicator of the amount of work that students conduct within the system. To consider complexity, we look at the overall argument size (the total number of hypotheses and relationships present in a student's notebook) to understand how much students engaged in the more complex, higher-level tasks of creating hypotheses and identifying relationships. We look for differences between the non-collaborative and collaborative groups by comparing the mean of these indicators across the groups, comparing dataset 1 with 2 and 3 with 4 respectively (see Table 4.5). Overall, we see a clear increase in the size of student solutions, and in the case of datasets 1-2, we see also an increase in the complexity (the argument size). We now consider the results within the individual comparative studies.

Table 4.5: Effects of adding collaborative features on data collection and argument creation. Significant effects are observed in each case except for the mean or arguments in the middle-school data**.**

	College-level Studies (1&2)		**Middle School Studies (3&4)**	
	Mean of Arguments	Mean of Data Entries	Mean of Arguments	Mean of Data Entries
With Collaboration (2 & 4)	**60**	**75**	**8**	**44**
No Collaboration (1 & 3)	**29**	**30**	**8**	**27**
P-value[3] (Significance)	**0.04**	**0.004**	**0.9895**	**0.039**

[3] Calculated using the Wilcoxon signed-rank test.

4.5.1.1 College Student Evaluations

Although the sample size was relatively small for this study (14 students in dataset 1, 22 students in dataset 2), the results were statistically significant, and fairly dramatic (see Table 4.5, Figure 4.1). Students in the collaborative subset of the college level study had an average of 60 propositions in their arguments, with a median of 45 propositions, compared to 29 propositions, with a median of 19 for the non-collaborative subset of the college-level group. This is a statistically significant difference (p-value = 0.04), demonstrating that the collaborative group had more complex arguments than did the non-collaborative group (identifying more hypotheses and finding more relationships between data and hypotheses). These results were mirrored in the data collection aspect of the system; an average of 75 pieces of data for the collaborative subset of the college-level group, with a median of 53 as compared to an average of 30 pieces of data, with a median of 27 for the non-collaborative subset of the college-level group. Again, this is statistically significant (p-value = 0.004), demonstrating that students generally engaged in more data collection within the collaborative system. From this data, we see supporting evidence for H1, that the collaborative efforts increased both size and complexity of arguments according to the given metrics (Figure 4.1).

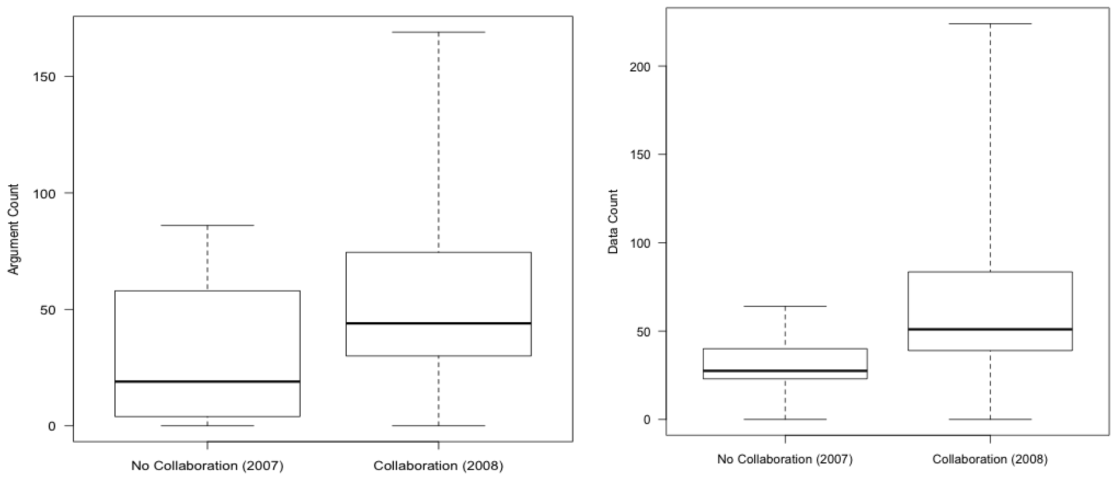

Figure 4.1: Box plots of the argument size (left) and the data collection size (right) for the college-level study. The difference between both groups is significant as judged by the Wilcoxon signed-rank test.

4.5.1.2 Middle School Evaluations

In reference to datasets 3 and 4, we see that subset of middle school students who had collaborative tools collected significantly more data (an average of 44 pieces of data with a median of 33) than the non-collaborative than subset of middle school students (an average of 27 pieces of data with a median of 22). This is a statistically significant difference (p-value = 0.039). However, the groups in the middle-school study had equivalent sized arguments, both had an average of eight argument propositions and a median of five. So, in this regard, we see limited but significant support for our hypothesis, namely that the size of student solutions was increased, but not the complexity (Figure 4.2).

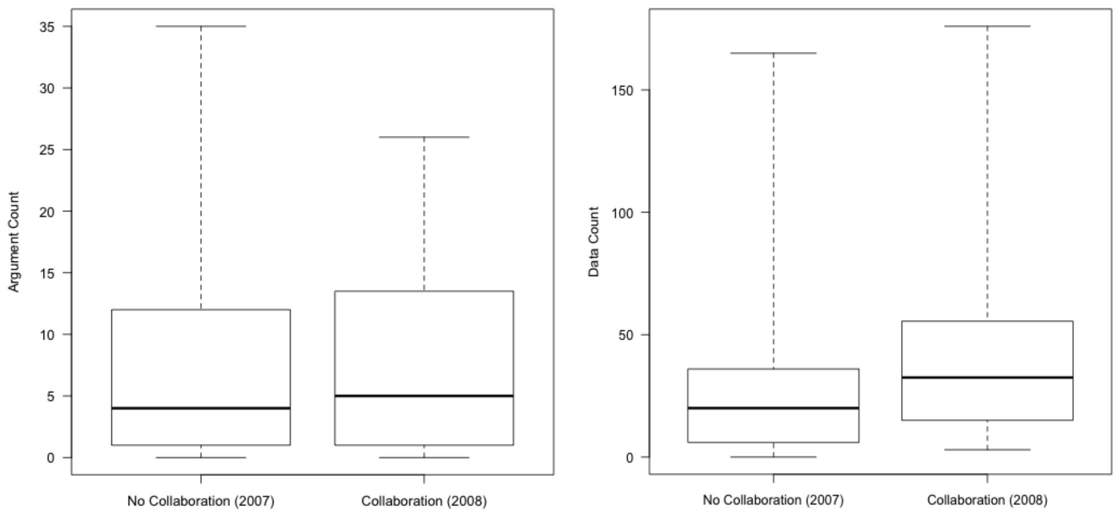

Figure 4.2: Box plots of the argument size (left) and the data collection size (right) for the middle-school level study. Argument size did not vary between groups (left), but data collection showed a significant increase (right), as judged by the Wilcoxon signed-rank test.

4.5.1.3 Discussion

In both middle school and college-level experiments, we see evidence supporting our hypothesis and conclude that the addition of collaborative features does tend to increase the size and complexity of student solutions (Dragon et al., 2009).

As to the lack of differences in complexity with regard to the middle school evaluations, we believe this might relate to the student population. Middle school students are possibly less skilled than college students in scientific inquiry. This would be particularly relevant when considering complex tasks such as creating a cohesive argument from data. The middle school students tended to form more simplistic arguments and a type of floor effect for their age might explain the lack of improvement with collaborative tools. Additionally, instructors in the middle school classes reported anecdotally that, as students were introduced to the collaborative tools, they thought it sounded like “cheating” and were afraid that people “would steal all their hard work.” This type of fear could limit the positive effects of collaborative tools for those that consider them to be “unfair.” However, teachers also reported that this fear generally dissipated after a brief period, and the students began to share more openly and explore possibilities brought up by other students.

It is also important to note that we have considered potential confounding factors in the set-up of the study and the metrics. First, we consider the populations and the state of the system between these two datasets, and find them to be the most valid datasets available for comparison of collaboration features. The student populations and teaching methods are similar, both being taught by the same teacher and the only major system update being the collaborative features that are the subject of study. As for teaching materials and content (the cases being studied), the college students worked on identical cases, while the 2008 middle school students worked on a case that was quite similar to the 2007 case (employing mostly the same data from the knowledge base and covering the same disease).

Second, we consider that the new collaborative tools allowed copying from one notebook to another. This should cause concern that the larger arguments and data collection could be merely an artifact of students copying work from one another. However, the data showed that the total increases in work was far greater than the number of copied entries, demonstrating that students in the collaborative setting did more independent data collection, hypothesis formation, and relationship creation than the subset without collaborative features.

4.5.2 Effects of Adding Coaching Tools

H2: The addition of coaching components improves student behavior by helping students focus on essential information and increasing the creation of semantically meaningful and content-rich student solutions.

We seek support for this hypothesis using the data from the university classes. We consider two different approaches to finding evidence of coaching effect. First, we consider an experimental design comparing datasets 8 and 9 from Table 4.1. Dataset 8 represents a control group that used the system with no coaching abilities, and dataset 9 represents an experimental group of students using a version of the system that provided on-demand coaching abilities. Students were randomly assigned to these control and experimental groups per section, as described in Section 4.3.3. Second, we consider the effect of coaching within dataset 9 (the experimental group that has access to coaching), analyzing the relationship between increased amounts of coaching and student solutions.

To operationalize H2, we need to apply some qualitative measures. Therefore, we use the qualitative approach of comparing solution scores as presented in Section 4.4.2 across groups to understand the potential effects of coaching on student solutions. When we did not find support for our hypothesis in these cross-group comparisons, we looked deeper to understand the more specific effects of coaching when coaching did occur. For this we consider only the group of students who could potentially receive coaching, and look for correlations between specific metrics and the amount of coaching received. Here we see some limited amount of support for our claims that coaching has an effect on student solutions.

4.5.2.1 Across Group Analysis

For this analysis, we consider datasets 8 and 9 from Table 4.1. Dataset 8 represents a control group of 76 students that used the system with no coach available, and dataset 9 represents an experimental group of 194 students that used the system with on-demand coaching prompts as described in Section 3.3.5. These students were divided into control and experimental groups by randomly assigning entire sections of the larger class, as described in 4.3.3. No effect was observed between the control and experimental groups in this analysis of overall Solution Score as can be seen in Figure 4.3.

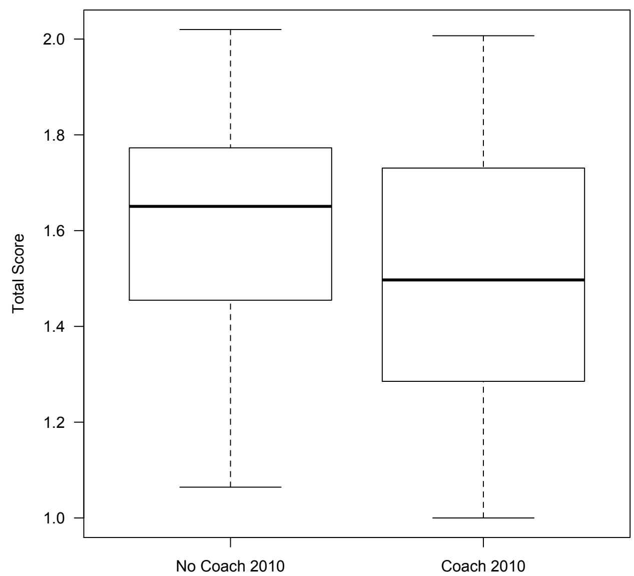

Figure 4.3:No positive effect of the coaching on solution score when comparing control and experimental subsets. Control represents students who had no access to coaching (dataset 8 from Table 4.1). Experimental represents students that had access to coaching (dataset 9 from Table 4.1)**.**

4.5.2.2 Within Group Analysis

Considering the results from the across group analysis, we see no evidence that the coach had an effect on student solutions. Considering the data further, we also observe that while the coach was present for all students included in DS9, only a small subset of students actually used the coaching capabilities compared with those that had the capabilities available (47 out of 194, or 24%). As most students in the experimental group did not use the coach, the comparison between the group with access and without access to this coach (control vs. experimental) is less meaningful. With this in mind, we limit our investigation to only the data from the experimental group that received coaching, dataset 9 from Table 4.1, and attempt to understand the effect of actual coaching instances. Specifically, we look for correlations between the number of times coaching was received by each student, and different qualitative measures of that student's solution.

Similar to the between-group analysis, we do not see evidence that coaching had an overall effect on solution score. Specifically, when looking at a linear regression of solution score and the amount of coaching received by students, we see no significant relationship (R-squared = 0.010, p-value 0.193). From this we see that the amount of coaching does not relate well with the students' solution scores.

However, analyzing on a finer-grained level, we do recognize some correlation between certain key indicators and the amount of coaching provided. Specifically, we analyzed both the Student Support Data and the Student Support Related Data (as presented in Table 4.3). These variables were chosen because they represent the most likely advice of the coach (the first type of advice the coach will offer in most situations), and as such are the most likely actions about which students received feedback.

Within dataset 9, we find that both Student Support Score and Student Support Related Score are correlated with the amount of coaching received (see Figure 4.4).

While the R-squared value is low for these regressions, they are statistically significant (Student Support Score R-squared = 0.045, p-value = 0.005, Student Support Related Score R-squared = 0.036, p-value = 0.012). This indicates that while the relationship is not predictive (using the coach does not guarantee any given student will have success), there is a statistical relationship (those who have used the coach tend to have higher scores in these assessment criteria).

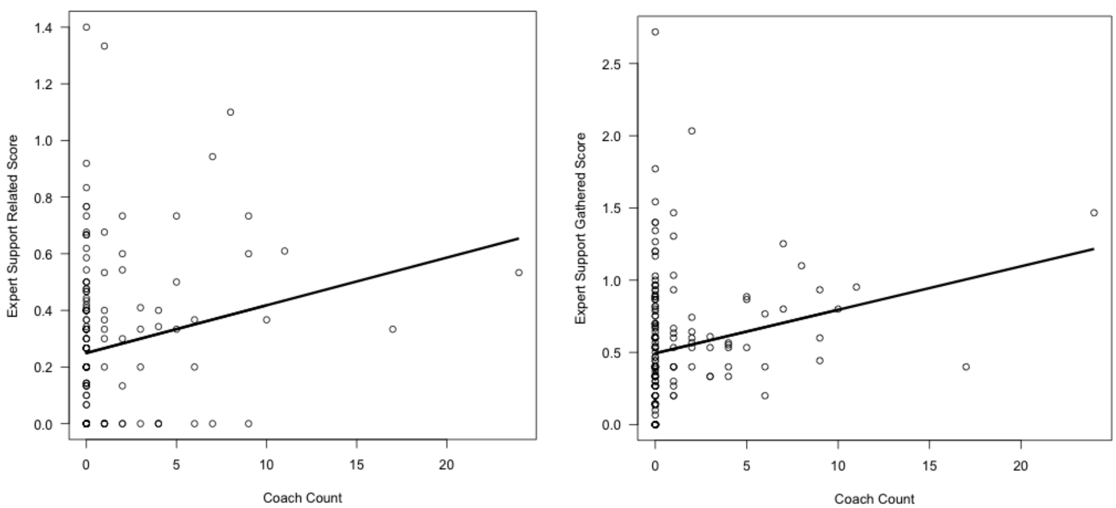

Figure 4.4: Linear Regressions with best-fit line showing a significant relationship between Student Support Related Score and coaching (left), and Student Support Score and coaching (right) for dataset 9 from Table 4.1, the experimental group of students who had access to coaching.

4.5.2.3 Discussion

Overall, we have seen no results to indicate that access to the coach as presented within the study will improve student overall solutions or motivate behavior change (Section 4.5.2.1). We theorize that this lack of effect may not be directly attributable to the underlying recognition mechanism that attempts to understand the student and identify opportunity for coaching, but rather the lack of effect might be due to a lack of use. The theory is supported by the low numbers of students who actually used the coach when available (24%), and the fact that more close study reveals that coaching does have some correlation with positive aspects of student solutions. Specifically, we see that increased coach usage is correlated both with students gathering appropriate data to support relevant hypotheses (Student Support Score), and with students relating that data with the appropriate hypotheses (Student Support Related Score).

We theorize that the coach was not used appropriately for several reasons. First, we recognize in retrospect that success with such a tool relies heavily on having an appropriate pedagogical model of coaching. Specifically, students should be provided with proper classroom support, demonstration, and rationale for using the coaching functionality. We see evidence for this case in Section 4.5.3, where we see demonstrable effects of properly defining and supporting the general classroom pedagogy. This type of pedagogical framework and introduction should be extended beyond the basic functionality of the system to also offer students and facilitators guidance in using both the coaching and collaborative features of the system. By informing users about the purpose and best practices of the various functionalities provided by Rashi, we can expect much better uptake and more focused use of the functionality. We offer concrete solutions in this respect in Section 5.3.1, namely considering the use of the coach in relation to the classroom pedagogy; presenting students and facilitators with a concept of when and how to seek help while using the Rashi system.

We also recognize a potential weakness of the user interface in our coaching system as used during the study. This prompted us to redesign the interface to take into account the issues observed. Specifically, students tended to ignore the coach completely, and anecdotally, some students using the coach reported that the coaching process was too long and complicated. Addressing these concerns, we developed a more continually present system of delivering coaching content that is still non-interruptive. We term the new interface "Suggested Links" coaching,(see Section 3.3.5).

Based on our lack of results from our between-group comparisons, and our more positive indications seen from within group analysis, we find that the coaching system has not been proven effective but still has the potential to benefit students. We suggest that a more clearly defined use of coaching within the classroom pedagogy and a more easily-used interface would prompt students to use the coach at a more appropriate level, allowing for a more realistic test of its potential.

4.5.3 Effects of Improvement to Classroom Pedagogy

H3: Clarification of the pedagogical approach with both facilitators and students improves student behavior, increasing the creation of semantically meaningful and content-rich student solutions.

This hypothesis tests the second significant change in classroom usage between 2009 and 2010. In 2009, instructors were given the Rashi system with little to no instruction on its use. In 2010, the Rashi team significantly enhanced and defined the specific pedagogical approach in the classroom. This involved defining and explicating the pedagogical model underlying our work in Rashi. In 2010, the Rashi team collaborated with the Biology instructors to understand and present a more coherent, unified classroom pedagogy, or set of instructional principles. The Rashi team produced supportive materials for the TAs and instructors of the class, and offered a one-hour training session presenting the abstract pedagogical goals of inquiry learning and the specific methods of approaching inquiry within the Rashi system. During this time, the TAs were also given materials and instructed on how to introduce the system to the students in the first 10 to 15 minutes of the students' first sessions. Finally, the TAs were instructed on how to interact during the sessions, what to watch for as students worked, and generally what to expect students to do during their time using Rashi.

With this major update to pedagogical approach in the classroom, we seek support for hypothesis H3 using datasets from 2009 and 2010. Specifically, we consider datasets 7 and 8 from Table 4.1, representing two groups of students using Rashi with the same functionalities and attempting to solve the same case. Collaborative features were available to both groups and coaching was not enabled for either group. The only significant difference between these groups was the presentation of the pedagogy. We use the same operationalization for H3 as we do for H2, presented in Section 4.5.2. Specifically we use the solution score as calculated from comparison with the EKB to demonstrate across-group differences in student solutions.

4.5.3.1 Across Group Analysis

When considering this solution score, which accounts for both quantitative and qualitative indicators of student solutions, we see drastic differences between the datasets (see Figure 4.5). The difference between the groups was clearly significant (p-value < 0.001).

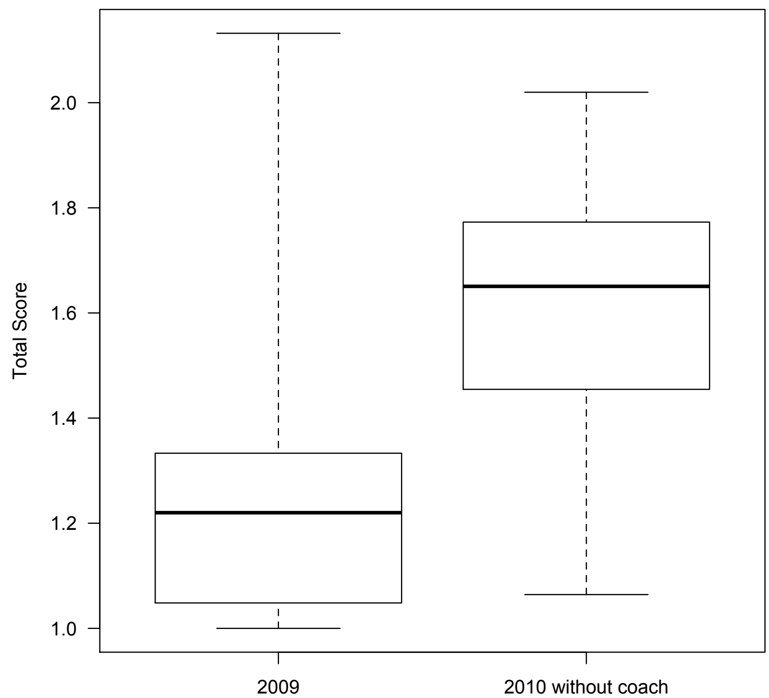

Figure 4.5: Solution scores as compared between dataset 7, Table 4.1 and dataset 8, Table 4.1. Dataset 7 represents students from the 2009 group having little pedagogical explanation and dataset 8 represents the 2010 group with a clearly defined and promoted classroom pedagogy. The groups are significantly different according to the Wilcoxon signed-rank test.

4.5.3.2 Discussion

This demonstrable effect of clarifying and explicating pedagogy is an important effect that needs to be considered generally when developing ITSs. In these results, we see clear evidence that building new, potentially productive functionality in learning systems is not sufficient to enhance the learning experience. Researchers in the fields of CBI and AIED are aware of this fact. Yet the central focus on technological development can lead sidelining the pedagogical approach in the classroom. Researchers must not disregard the importance of explicit pedagogy, and the importance of the facilitators who present this pedagogy to the students. Our situation with the Rashi system is especially susceptible to this problem because, in the university setting, we are introducing both facilitators and students to the often unfamiliar or unpracticed pedagogical concept of inquiry learning. In such a situation, facilitators and students need to leave old styles of learning behind and develop new practices. This process requires more explicit instruction.

In the field of AIED, research can tend towards emphasis on how the computer alone can understand and support students (as we present in Chapter 3). However, we cannot consider this factor in isolation, unless we have well-established pedagogical models. This creates a situation where we must always consider both the tools available to students (Section 4.3.1 and 4.3.2) and the classroom pedagogy with which the tools are being used (Section 4.3.3).

4.5.4 Combining Coaching and Collaboration

H4: A coaching system can identify opportunities to promote targeted content-focused collaboration.

We approach this section of evaluation differently than prior sections. As the work is more preliminary, we analyze student data in order to understand the potential opportunity for future interventions, rather than having the opportunity to study the effects of these interventions on actual students. Specifically, we look for supporting evidence that a coaching system can promote content-focused collaboration (as described in Sections 2.4 and 3.4) using a combination of the coaching system and the collaboration tools. We seek to identify these opportunities in two different ways (as described in Section 3.4):

- *Collaborative contributions as coaching opportunities*: finding opportunities to introduce content support when student are collaborating.
- *Coaching as opportunity for collaboration*: finding opportunity to encourage targeted dialog based on the content of the students' solutions.

4.5.4.1 Using Collaborative Contributions as Coaching Opportunities

One means of collaboration within Rashi is through a chat tool (Section 2.4.2.1). In order to take advantage of the chat system to help us understand student work and provide pertinent information from the EKB to students engaged in chat, we need only recognize the content of their chat messages and apply our standard coaching techniques (as presented in 3.4.1). In order to illustrate the usefulness of this technique, we seek to demonstrate the ability to correctly recognize the content of chat messages. To accomplish this automatic content recognition, we use the same approaches as we used on hypothesis matching, only now applied to chat messages. This procedure provides a matched EKB entry for each chat message. Similarly to the validation of hypothesis recognition (see Section 4.4.2.1), we need only to confirm that matches can be identified with reasonable accuracy to confirm that additional relevant content from the knowledge base can be offered during chat interactions. This relevant content includes data and hypotheses related to the concepts mentioned in the chat messages (see Section 3.4.1).

So, our validation of the concept rests on the idea that we can correctly match student chat messages with elements from the EKB. As presented in Dragon et al. (2010), we test the validity and usefulness of our automated matching techniques as applied to chat by analyzing the datasets from two studies that made most use of the chat tool, dataset 2.1 and dataset 5 from Table 4.1. During these two sessions, a total of 796 non-blank individual chat messages were sent. Each student, however, could only view the chat happening within his or her group of three to five students. Groups were generally self-selected, with teachers making suggestions when individuals were without a group.

An independent judge (a member of the software team not involved or familiar with the matching scheme or the knowledge base) created a set of comparison data over which the efficacy of an automatic matcher could be assessed. The judge rated all 796 chat messages by comparing the student's statement with this matched content and marked whether the knowledge base entry was appropriate, Table 4.6.

Table 4.6: Results of the automated chat-matching algorithm in correctly identifying the specific content of each student message.

Data Set	Automated Content Matches	Judged Correct Content Matches	% Content Match
DS 5	63	44	70%
DS 2.1 – Hyperthyroidism	69	52	75%
DS 2.1 – Anemia 1	45	25	56%
Total	177	121	70%

4.5.4.2 Discussion

When considering these percentages, we must also account for two additional factors. First, we are matching free input text from students with a large set (hundreds) of knowledge base elements. Therefore, even fairly low percentages (e.g. 50%) are far better than random. Second, we consider our principles put forward in reference to providing support with uncertain assessment (Section 3.3.4) that help us understand how to provide feedback that is not harmful even if the recognition is mistaken. Accounting for these factors, we see reasonable recognition rates to presume that coaching using recognition in chat could be useful. This is particularly true in regard to the Hyperthyroidism case, on which the Rashi team focused their efforts of EKB formation and refinement.

The success of identification of dialog utterances using the knowledge base also indicates that building knowledge bases is a viable option for the intelligent tutoring community, when researchers seek to understand the domain content of student discussion. Related to this approach, we also see that building a small incremental knowledge base focused on the content pertinent to the cases at hand (Section 3.2.2.1) can offer useful results in this regard. This result is demonstrated clearly by the contrast between the different cases in Table 4.6. The SMEs and developers put considerable effort into enhancing the EKB for the cases used in dataset 5 and dataset 2.1-Hyperthyroidism. The dialog recognition process on these portions of the study was more likely to be correct than the dialog recognition for dataset 2.1-Anemia 1. While this leaves us with less-than-ideal results for dataset 2.1-Anemia 1, it reinforces the idea that our knowledge base structure and creation process are working successfully. Added effort led to direct improvement of matching capabilities

We do recognize that these significantly lower rates on the recognition of the Anemia 1 case are far from optimal. However, beyond manual improvement of the knowledge base, there are more automated approaches to improving recognition within the knowledge base as well. We have investigated automated ways of pruning the knowledge base such that we make fewer matches overall, but also increase the confidence in the match procedure (Floryan et al., 2012). Using such techniques, these matching rates could be improved to even higher levels of confidence. Through these different methods of incremental improvement, we see even more promise to the concept of recognizing chat using a knowledge base, and using this recognition as an opportunity to offer related content support.

4.5.4.3 Using Coaching as an Opportunity for Collaboration

Here we consider support for H4 by looking for ways to promote targeted collaboration using the coaches' abilities to recognize student work in relation to the knowledge base. We can use our coaching abilities to promote collaboration, rather than using collaborative contributions to offer more coaching, as presented in Section 4.5.4.1. We seek evidence for this opportunity by evaluating logged student solutions from the same datasets (2.1 and 5, Table 4.1). Across the student solutions from these studies, we sought to identify a specific opportunity where content-focused collaboration could be promoted, namely the tutoring concept (presented in Section 3.4.2). To instantiate this tutoring concept, the system identifies opportunities where one student has correctly created a hypothesis, gathered supporting data, and related the data to the hypothesis. Then the system considers the other students within the group, looking for another student who has collected the same data, but has not yet added the correct hypothesis or related the data to it. While this focuses on only one instance of one type of content-focused collaboration, we believe it is a useful indicator of the opportunities available to promote content-focused collaboration.

In dataset 2.1, we found ample opportunities to promote this 'tutoring' type of content-focused collaboration. For each student, when reviewing their finished work in the system, we found an average of 13 instances (median 11) where an intelligent agent could have brought students together to discuss the differences according to the definition of the tutoring type of collaboration above. This means that for each student, there was an average of 13 pieces of data related correctly to a hypothesis in their own solution, which some other student had collected but had not related correctly. In dataset 5, we see the same potential for content-focused collaboration. Overall, we find an average of 20 instances (median 17) where the system could recognize a collaboration opportunity for any given student within the class.

4.5.4.4 Discussion

By considering these datasets, we see ample opportunities to promote the tutoring concept of content-focused collaboration as presented in Section 3.4.2. One weakness of the analysis is that it was done post-hoc, and therefore the analysis considers only final student solutions rather than work-in-progress student solutions as would be available during real-time interactions. However, while the number of opportunities will clearly be lower at some early stages before many hypotheses are formed and data are collected, overall this analysis demonstrates that opportunities for content-focused collaboration (in this instance peer tutoring) are plentiful within the Rashi system.

4.6 Summary

The research presented here investigates the general question; *does targeted use of both collaborative features and feedback from coaching software improve students' solutions?* This chapter presents the methods and results of our evaluation of this general question. We present the specific hypotheses that break the general question into components to be individually addressed. We then present the classroom studies where we have collected the data in order to test the hypotheses. We present the techniques used to analyze this data with respect to the different hypotheses, and finally present the results of this analysis upon each hypothesis.

Classroom studies were conducted over a number of years involving thousands of students. These studies were carried out in different classroom settings involving both college and middle-school students. Different studies were conducted with different versions of the software at various stages of development. As such, we select among these datasets carefully to find the most appropriate data with which to test each hypothesis, minding both variance in software features, classroom setting, and system functionality.

To analyze the data in order to test our hypotheses, we used several techniques to create numerical indicators assessing student solutions. These involve both structural, quantitative measures (such as the number of hypotheses created by a student) and content-based, qualitative measures (such as the number of hypotheses that are recognized as relevant to the case according to the EKB). Quantitative measures are used to test H1, whereas both quantitative and qualitative measures are combined to test H2 and H3. We consider existence evidence to test H4, namely we search past data for the evidence that the desired type of intervention could have been offered. This existence proof was used because no completed system was tested thoroughly in the classroom to provide insight on the effect the system has on actual users.

In summary, we find sound support for hypothesis H1; *the addition of collaborative features improves student inquiry behavior, increasing the size and complexity of student arguments.* We demonstrate that students having access to collaborative features provided significantly larger and more complex solutions than that of a control group.

We did not find sound support for H2; *the addition of coaching components improves student behavior by helping focus students on essential information and increasing the creation of semantically meaningful and content-rich student solutions.* There was no observable positive effect of the coaching capabilities across the groups. However, we consider that the coach was not used by the majority of participants in the experimental group, and upon inspection of more specific indicators within this experimental group, we do see correlations between coaching use and positive aspects of student solutions.

We find sound support for H3; *clarification of the pedagogical approach with both facilitators and students improves student behavior, increasing the creation of semantically meaningful and content-rich student solutions.* We demonstrate that students presented with a more well-defined and explicated pedagogical approach produce significantly better solutions, as judged by the overall metric of solution score.

Finally, we find sound support for H4; *an expert knowledge-based recognition system can identify opportunities to promote targeted content-focused collaboration.* We demonstrate that these opportunities can be recognized in two different directions, utilizing collaborative contributions to offer content support, and using content recognition in student solutions to promote collaboration. Each of these scenarios offers opportunities to combine coaching and collaboration in interesting ways that could potentially harness the power of both approaches.

Overall, we consider the investigation of these different hypotheses to offer solid support to answer our main research question positively. We have demonstrated the positive effects of collaboration (H1) and seen some indication of the positive effects of coaching (H2). We also have demonstrated the potential of combining these approaches (H4), which has the potential to improve both results, targeting collaborative efforts around content upon which students should be focused according to assessment. Additionally, outside the main research question, we also illustrate the importance of developing and explicating specific pedagogical models when using the software in the classroom (H3). In Chapter 5, we consider the overall implications and conclusions that can be drawn from this data, and how future work can utilize these results to refine and further this overall research agenda.

CHAPTER 5

CONCLUSIONS & FUTURE WORK

5.1 Introduction – Hypotheses Revisited

Here we present the over-arching conclusions of this dissertation research. First, we present a brief summary of the evaluation presented in Chapter 4, to clarify the specific hypotheses and their respective results. Three hypotheses were investigated as part of our main research question: *Does targeted use of both collaborative features and feedback from coaching software improve students' solutions?* We found mostly supportive evidence of this claim, although we had some mixed or negative results with respect to coaching.

We found solid support for hypothesis H1, which links the availability of collaborative features with the creation of larger and more complex student solutions. We did not find the same level of direct support for hypothesis H2, which attempts to link the availability of coaching with improvements in both size and content of student solutions. However, we did find some correlation between the amount of coaching received and certain specific metrics that are more directly linked with feedback advice. Finally, we demonstrated the potential for coaching and collaboration to be used in new and powerful ways through their combination. Hypothesis H4 demonstrated that collaborative contributions could provide an opportunity for content coaching, and that coaching techniques could identify opportunities to promote content-focused collaboration. With these results combined, we have shown compelling support that collaboration and coaching can have positive effects on student solutions, particularly if we can harness the potential power of their combination.

Aside from our main research question, an important result also emerged due to our experimental design; namely the positive result associated with H3, which linked the explication of classroom pedagogy with improvements in both size and content student solutions. From this we see clear support for the idea that building a functional tool is not sufficient to improve student behavior; students must be made aware of how and why to use the tool appropriately.

Now we present the summative results and lessons from the previous chapters. We describe, in a holistic sense, the outcomes and implications of this research when considering coaching, collaboration, and the combination of these techniques to further the intelligent support of a tutoring system specifically geared towards inquiry learning and ill-defined problem spaces. Following naturally from this summation, we present the main foci of suggested future work, and initial steps towards these future goals.

5.2 Lessons Learned

Chapters 1-4 presented the theoretical background, implementation details, and evaluation of a collaborative inquiry-learning system for ill-defined problem spaces. While the discussions in Section 4.5 focused on the detailed implications of each specific hypothesis, we must also consider the more broad and holistic implications of this research. Specifically, we consider the implications of our work on Intelligent Tutoring Systems (ITSs) for ill-defined problem spaces with respect to classroom pedagogy (Section 5.2.1), coaching (Section 5.2.2), collaboration (Section 5.2.3), and the combination of coaching and collaboration (Section 5.2.4). For each of these topics, we discuss the implications of our specific efforts, and lessons that the research community can learn from our experience.

5.2.1 On Classroom Pedagogy

Our research clearly demonstrates that explicating the pedagogical model for classroom use and ensuring that both the facilitators and the students understand this model is of the utmost importance to productive use of inquiry learning software in the classroom (H3, Section 4.5.3). While this idea is recognized by other research, particularly researchers more focused on classroom dynamics than technological aspects, the data presented here provide solid evidence of the importance of classroom pedagogy. These aspects should be explicitly recognized and accounted for all in future work.

We consider this problem to be even more important when the classroom pedagogy promoted by the software differs drastically from standard classroom experience, as was the case in our main university studies in 2009 and 2010 (Table 4.1, rows 7-10). The Rashi inquiry learning exercise replaced a standard lab session, where students were accustomed to receiving a specific procedure (lab protocol) and were expected to complete that procedure according to the predefined steps. To have students engage productively in a more open-ended experience with an unfamiliar tool requires clear instruction as to the student's task within Rashi, the functionality that Rashi provides, and the ways in which the student can use Rashi specifically to accomplish the task.

In Section 4.5.3, we demonstrate that explicating and communicating this pedagogical model for the classroom to facilitators and students showed immense effects on student work, eliciting student solutions that were larger, more complex, and more similar to the expert solution. This concept needs to be applied not only to the overall system (as in 4.5.3), but to any enhancements of the system, especially those that are pedagogically driven (i.e., enhancements that attempt to influence the way that students learn from the tool). This lesson applies to all of the other major efforts within this research (Sections 5.2.2-5.2.4), and we discuss methods of utilizing this result to improve future work in Section 5.3.1.

5.2.2 On Collaboration

Our research demonstrates that collaboration is both well suited to inquiry learning in ill-defined domains, and has potential to increase students' productivity (collaborative features resulting in generally larger and more complex student solutions). As discussed in Section 2.4, collaboration has been shown to be useful to students engaging in complex tasks. This implies that inquiry learning in ill-defined problem spaces is uniquely suited to benefit from collaborative efforts. These collaboration opportunities can be supported through both dialogs and/or through shared workspaces. In Section 2.4.2, we present how the Rashi system implements these standard forms of collaboration.

We offer direct evidence that these collaboration features have a positive effect on student solutions. With hypothesis H1 (Section 4.5.1), we present evidence that the addition of these collaborative features to the system had a demonstrable effect on the size and complexity of student solutions. We also showed clearly in Section 4.5.4 that the system can recognize domain content in student's collaborative work and can provide extra coaching based upon this recognition.

A common weakness of these standard collaborative approaches is the strong division between the collaborative contributions and the domain content around which the collaboration should be occurring (as discussed in Section 2.4.1). Moving beyond these standard approaches, we describe in detail how collaboration can be improved to focus more clearly on, and situate discussion alongside, relevant domain content. We describe key features and functionalities that can help students focus on the content at hand while collaborating. We introduce the concept of discussable objects to conceptualize exactly how a system can bring a tighter coupling between the learning artifacts (in our case the student solutions in Rashi) and their collaborative contributions (dialog). We present our implementation based on these more advanced functionalities of discussable objects (Section 2.4.2.3). We also describe how these discussable objects can actually offer improved recognition of student work (Section 3.2.3). We consider this ability of Rashi to promote content-focused collaboration to be a model for future work based on collaboration in ill-defined domains.

This concept is gaining popularity, and is being used in other large-scale research projects. Metafora (Dragon et al., 2013) is a web-based Computer-Supported Collaborative Learning (CSCL) system that brings together multiple learning environments to support students engaging in inquiry behavior. A central technical piece of this project is the *referable object* (Dragon et al. 2011), an extension of the idea of discussable objects as implemented within Rashi. This system enhances the idea of discussable objects by supporting not only discussion, but also manipulation of these shared objects by other team members. The Science Created by You (SCY) project, presents a similar concept termed emerging learning objects (ELOs). ELOs are specific objects within the learning environment that students create, share, and around which discussion and group work is organized (De Jong et al., 2010). Both of these systems address the difficulties with separation of content and collaboration in a similar fashion to the approach of discussable objects within Rashi.

Overall, we present compelling evidence that collaboration is a useful approach within our system, that it has positive effects on student solutions, and that there are concrete ways in which it can be improved to further the state of the art. However, accounting for our results from Section 5.2.1, we consider that our classroom pedagogy about collaboration was not defined to an appropriate level of detail, and was not communicated in an effective way to facilitators or students. We provided students with collaborative features that were generically useful across many use cases (Section 2.4.2), and did not provide students with instruction or guidance as to when, why, or explicitly how to use the features. We saw a distinct lack of use of some of the new collaborative features (particularly the critique/rebuttal feature, Section 2.4.2.3) during later pilot studies. We theorize this lack of use might be due to a lack of pedagogical grounding (i.e., we did not define for the students when or why they should use these features). Much in the same way as the overall student performance can be improved with explicit classroom pedagogy (Section 5.2.1), we theorize that the collaboration could also be improved through explicit instruction. Students need to understand the general collaboration principles, and how they are expected to enact those principles within the Rashi system.

5.2.3 On Coaching

Our research demonstrates no convincing evidence that our implementation of on-demand coaching has an overall positive effect on students, yet we do see some correlation between coaching and specific behaviors important to inquiry learning. As prior evidence provides us with some confidence that the overall concept of coaching is both necessary and useful, we consider ways in which our system or study might not have been fit to demonstrate these positive effects, and how this weakness in our approach might be remedied.

We present details on how an ITS operating in an ill-defined domain (in this case medical diagnosis) can both recognize student work (Section 3.2) and use that recognition to offer feedback to students (Section 3.3). For evaluation, in Section 4.5.1, we analyze the effects of our specific implementation of the on-demand prompts coach. The analysis shows no overall effect on student solutions in a cross-group study when comparing students who had a coach available with those who did not have a coach available. However, there is some limited evidence that increased use of the coach was correlated with an increase in certain behaviors that are vital to the inquiry process (namely collecting and relating data relevant to the case at hand). This demonstrates some potential positive influence of the feedback given.

Even with these results, we argue that coaching in ill-defined domains is necessary and useful. As we discussed in 2.2, ill-defined problem spaces provide opportunities for deep learning and development of higher-level learning skills. Yet these problem spaces also require special methods of learning, such as inquiry learning presented in 2.3. Both of these pedagogical concerns, ill-defined problem spaces and inquiry learning, differ from traditional classroom methods, and even traditional ITS methods, as they offer more open-ended approaches to learning. This student freedom creates a stronger need for some type of guidance and support for students (Kirschner et al., 2006). Finally, many external sources provide evidence that offering support and scaffolding to students in similar scenarios or other ill-defined problem spaces has been successful in promoting student learning (Baghaei et al., 2007; Pinkwart et al., 2007). From this theoretical background and sound scientific studies conducted across the field, we are confident the type of coaching we offer can be useful, if implemented correctly both in the system and in the classroom.

Considering the above evidence suggesting that coaching in ill-defined domains is both necessary and useful, we must consider why our study did not demonstrate similar results. A predominant issue with studies involving new, complex learning environments is the relatively short duration of the studies. As these systems present new approaches to learning, there will only be real improvement with extended use. Beyond this point, the main issue with our approach is likely related to our integration with the facilitators. The coaching system was not introduced to facilitators or students with explicit classroom pedagogy, and facilitators were not involved in the coaching process. To remedy this, problem, we must find ways that classroom pedagogy and technology can help facilitators to support both rapid prototyping and testing of coaching tactics during development, we also need to define a clear role for facilitators in the final system. We briefly describe these two issues below, and describe potential solutions in Sections 5.3.1 and 5.3.2, respectively.

Considering again the results from Section 5.2.1, we hypothesize that the root cause of our lack of coaching use, and therefore evidence of its efficacy, lies in a lack of explicit classroom pedagogy surrounding the coaching functionality. Again referring to the results presented in 4.5.3, the importance of explicit classroom pedagogy is clear. From this we know that students and facilitators should be instructed about the purpose and functionality of the system, explaining when and why students should be using the system in certain ways. This lesson can be transferred to the coach, explaining clearly the role, usefulness, and function of the coach in the learning process.

This coaching aspect of the classroom pedagogy is all the more necessary because our coach is designed to be non-interruptive (see Section 3.3.5), meaning that students needed to request help in order to receive any guidance. While requesting help may seem an intuitive task, literature review in related work does demonstrate that the awareness of when help is needed (or not needed) and emotional preparation to ask for help are not readily available skills that all students have, and actually can require specific meta-level instruction to acquire (Aleven et al., 2004, Newman 2000). We see this lack of coaching/help-seeking pedagogy as the main detractor from use of the coaching system, and therefore the major roadblock in detecting the efficacy of the larger recognition and coaching system. We describe the future work of implementing this classroom pedagogy in Section 5.3.1.

The second issue we flag (that is somewhat related to the first) is the lack of a role for the facilitator in both the production of and the use of the coach. The coaching system was brought into the classroom in its fully functional, completely automated form. There was no initial study of how students might perceive feedback, or how the facilitators in the classroom might play their role within the learning process. This setup requires an enormous upfront effort and provides only one avenue for success; whereas coaching can be introduced incrementally with prototypes that involve facilitators. These prototypes can use facilitators to mimic the costly efforts to produce intelligent feedback. In this way, lower-cost, higher return studies can be done rapidly to clarify strengths and weaknesses of an approach and allow far greater chances of producing results in final studies. Such studies also help refine and solidify a successful coaching pedagogy, which helps to remedy the first issue we described as well. We discuss some tactics that improve coaching success by introducing more explicit roles for facilitators in Section 5.3.2.

5.2.4 On Integrating Coaching and Collaboration

As the most far-reaching goal of our research, we argue that the intelligent analysis features of a coaching system can be used in combination with collaborative features. By combining these two distinct aspects, intelligent systems can understand collaborative contributions better, and can enhance collaboration efforts by identifying opportunities to bring students together to collaborate for specific reasons (as presented in 3.4).

First, we argue that by employing the methods of understanding students' individual work, a system can also gain some understanding of the content of the students' collaborative contributions. This understanding provides new opportunities for coaching. We present evidence of this process in Section 4.5.4.1 by demonstrating that our system can recognize student chat contributions with reasonable accuracy (70%). We then demonstrate how this understanding could be used to offer additional suggested links within the chat tool, providing a means of co-locating content and collaboration.

Next, we argue that by utilizing the analysis of students' individual work, a system can identify key moments when it should prompt students to collaborate. These key moments can prompt students to enact different collaborative roles at different times, co-constructing knowledge, debating concepts with peers holding different viewpoints, or being either a tutor or a tutee. All of these roles are potentially fruitful collaborative tactics, as described in Section 3.4.2. We present evidence that these situations can be identified in Section 4.5.4.2, where we demonstrate that one such key moment (peer tutoring), can be identified with relative frequency in the dataset studied.

Overall, we see great promise in the combination of intelligent content support and collaboration to promote content-focused collaboration. However, as stated in regards to both coaching and collaboration, the explicit classroom pedagogy plays a crucial role in these situations. Such on-the-spot prompts for specific types of collaboration require a sound pedagogical model of classroom peer learning. Modern literature presents many such models of peer learning (Crespo et al., 2005; Garcia & Pardo, 20102; Topping, 2005). Students will need both early instruction on the expected types of collaboration, and timely, appropriate information as to how to collaborate, what role they might consider, and how to partake in that role (Choi et al., 2005).

We can see an example of such a specific pedagogy when considering the concept of peer tutoring presented in Section 3.4.2. In this situation, two students are brought together so that they may help each other. The system can provide the helper with information about what topic they should discuss, and specific information about why the student might need help. Related research in well-defined domains on peer tutoring has demonstrated that this “helpers interface” can create productive learning outcomes for both the peer tutor and the tutee in such a situation (Walker et al., 2008). This idea of providing support by prompting peers to help one another is recognized as an open and interesting research direction in the field of intelligent collaborative systems (Magnisalis et al., 2011). Defining the specific classroom pedagogy of this prompted content-focused collaboration in specific scenarios is still an area for future research on the Rashi project (see Section 5.3.1).

5.3 Summary and Future Directions

The work presented in this dissertation represents early steps towards using coaching and collaboration tools within inquiry learning systems for ill-defined problem spaces. As both technology and teaching strategy are advanced and refined, we are likely to see more systems utilizing these alternative teaching methods and practicing in ill-defined problem spaces. The need to provide automated support and guidance will also increase as such systems are used more often.

We present work that demonstrates the potential usefulness of both coaching and collaboration as support systems for tutoring in ill-defined problem spaces. We also present the idea of combining these two different methods in strategic ways that could potentially improve the contributions of both the coaching and the collaboration to the learning process.

Several steps are necessary to solidify the ideas put forth in this research and demonstrate the unified vision of coaching and collaboration in order to produce content-focused collaboration. Specifically, we consider three separate aspects that would offer further evidence of the applicability and efficacy of our contributions. First, explicit classroom pedagogy must be developed and explained in greater detail for the different functionalities of the system (Section 5.3.1). Then, along with this pedagogy, coaching and feedback should involve more human effort, and be introduced in a more incremental, iterative fashion (Section 5.3.2). Finally, the focus and evaluation of the system should include higher-order learning skills, to identify the more complex and far-reaching learning gains that are theoretically attributed to these types of learning scenarios (Section 5.3.3). Through these three tactics, the research can move forward in providing new methods of teaching, successfully providing intelligent feedback that can support these new teaching methods, and further demonstrating the usefulness of these teaching methods. We now discuss each specific recommendation in detail.

5.3.1 Develop Explicit Classroom Pedagogy

Reviewing the conclusions from all aspects of the system, we recognize a recurring theme, the lack of explicit classroom pedagogy (Section 5.2.1). This is true across our coaching tools, collaborative tools, and our collaborative coaching tools. The technological focus of the project inherently meant that classroom pedagogy was less developed. However, we recognize that, in the end, the impact of these external pedagogical factors was greater than we assumed, and that students and facilitators need more external guidance to use the system in a productive manner. Students and facilitators require further instruction on the tasks and goals involved with the learning methods, and how to use the system features to complete those tasks and achieve those goals (as presented in 5.2.1). This explicit instruction is particularly important when deploying a classroom pedagogy that is unfamiliar to students and facilitators (e.g., inquiry learning). We demonstrate in Section 4.5.3, that when an explicit classroom pedagogical was presented to teachers and students, it drastically improved use of the system. We hypothesize that the same improvements geared towards coaching, collaboration, and collaborative coaching features would yield similar positive results.

First, for coaching, one must define specifically when and how the coach should ideally be used. Then one must convey this information to students both when introducing the system, and also at crucial reminder points during use of the system. Similarly, we must define exactly how students should collaborate. Should they be working continuously in pairs, co-constructing knowledge, or should they be working independently and only communicating with others when they are stuck or confused? Our technology supports a wide variety of the collaborative processes (as described in Section 2.4.2). We suggest that researchers and teachers decide upon the type of collaboration they seek to promote in the classroom, and explicitly encourage the appropriate collaboration method. Finally, when considering content-focused collaboration, teachers again need to define and encourage a specific pedagogical approach. Students need to understand their specific roles and how and when they should enact these roles. When the system is prompting for content-focused collaboration, students should understand, both from their teachers and from the system, the different reasons they have been brought together to discuss a topic, and how they are expected to proceed.

One clear source for inspiration and specific strategies for explicating such classroom pedagogy can be found in the CSCL research area of scripting. A *script* as defined in CSCL is "a set of principles that designers may apply to help scaffold specific classes of interaction" (Dillenbourg & Hong 2008). In this sense, the term script is used in CSCL to mean precisely the type of pedagogical model we require: defining learning objectives, activity types, sequencing, role distribution, etc. (Koller, Fischer & Hesse 2006). While concept of scripts in CSCL is only applied to collaborative efforts, we see also the need to define pedagogical models, or scripts, that define the individual interaction with the system and the coach. The CSCL research on scripting offers possible basis for grounding the future work on pedagogical models in theory and providing first steps towards implementing such models in a more formal way. Some set scripts are already defined and considered "best practice learning designs," such as the Collaborative Learning Flow Patterns (Magnisalis et al., 2011). Such scripts might be altered to work with the overall Rashi design, rather than creating new scripts.

5.3.2 Increase Human Involvement in Coaching

Specific to coaching, we suggest that future efforts should involve human facilitators more directly in the process. In general, the most low-cost and high-yield solutions to improving coaching functionality involve using human effort. This can manifest in two ways. First, humans can be involved in the development of coaching capabilities by taking on the role of a potential AI system, which we present in the form of "Wizard of Oz" studies (Mavrikis & Guiteirrez-Santos 2010). Using humans in this process can decrease programming costs and increase the return on experiments by allowing for a more incremental development approach where individual features (e.g., specific types of feedback as discussed in Section 3.3, specific methods of intervention as discussed in 3.3.5, etc.) can be studied and understood in more detail and with more control.

The second consideration is the facilitators' role in the finished system and finalized classroom pedagogy. There is potential for facilitators to make use of analysis information provided by the system to offer support to the student (Mavrikis, Dragon & McLaren 2012), even to the point where the system can recommend feedback for the facilitator to provide (Keshtkar et al., 2012). We now describe how each of these modes of human involvement can be utilized in future work to improve the efficacy of our process.

5.3.2.1 Implement Wizard of Oz Studies

An emerging field of research is how authors can design and test the functionalities of intelligent coaching software without expending the full resources to implement such functionalities. One particular focus of this work is the *Wizard-of-Oz* (WOZ) study, referring to the famous book and then Hollywood movie in which the "wizard" is in fact only a "man behind the curtain." While this name has been applied to a wide variety of studies, we consider the definition offered by Mavrikis and Gutierrez-Santo (2010) where they distinguish the iterative process of development applied by the approach. During this type of WOZ study, a human facilitator is given some ability to choose and send feedback to students. This feedback is presented to the student by the system, in the standard interface in which the system would normally send automated feedback. In this way, the students are not necessarily aware that the feedback is coming from a person, but rather often assume it is created by the system itself. The "wizard," or facilitator, acts in place of the intelligent system, viewing student work and providing feedback that appears to come from the system.

This type of experimentation can provide beginning or intermediate trials of feedback systems, and also support the testing of partially implemented approaches. In this way, researchers learn the potential benefits of, or issues with, feedback without the up-front cost of building a fully functional system. Lessons learned during these intermediate studies can then be applied directly to the system as it is developed, rather than waiting for completion before testing. These WOZ experiments have mostly been used with single students in relatively well-defined domains (Fiedler et al., 2004; Maulsby et al., 1993). However, new research efforts are pushing this tactic into more open, exploratory domains (Mavrikis & Gutierrez-Santos 2010).

In addition, this work has been further augmented in recent projects to offer facilitators access to higher-level analysis information from the system. For example, facilitators can be given access to analysis of the structure or content of student contributions (Mavrikis et al., 2012). This method has multiple advantages. First, this information can simply ease the facilitator's role, providing higher-level, more abstract understanding of the current students' state. The facilitator has the difficult task of offering feedback over a broad range of topics and across possibly many students. Having access to the analysis information provides concise, poignant, summarized information about the situation in which they must decide to give feedback. For example, the analysis components might report that one part of a student solution is deemed acceptable, while a related part of a different student solution is deemed unacceptable (the base components to recognize the potential for tutoring content-focused collaboration as presented in

Section 3.4.2). The facilitator can use this analysis information to pair students for content-focused collaboration if appropriate. The analysis information makes the pairing process much more streamlined for the facilitator.

The second major benefit of using this analysis information within the WOZ process is to make the feedback given by facilitators increasingly easier to automate directly within the system. As the facilitator relies more on the analysis information and less on direct observation, the algorithm with which to automate the facilitator's efforts becomes increasingly clear. As this process reaches its endpoint, facilitators gather the information necessary to trigger feedback solely from the analysis system. When facilitators are following direct logical rules to offer feedback based purely on this analysis information, that behavior can be coded into automated coaching rules. In this way, the WOZ experiments begin in a more exploratory fashion, and then refine the analysis components and the rules followed by facilitators until specific behaviors can be confirmed as useful and encoded. This process of refining the facilitator's role to allow for automation of their tasks is termed Iterative Communication Capacity Tapering (Mavrikis & Gutierrez-Santos 2010).

We see promise in the idea of WOZ-style experimentation in Rashi. The system already has a well-defined and implemented method of analyzing the student work in comparison to the Expert Knowledge Base (EKB) (see Section 3.2). This system could offer the analysis information and prompts for potential interventions to facilitators. The facilitators could then be given direct control over the feedback mechanisms currently present. This would allow experimentation with both timing and interruption levels of this feedback (issues presented in 3.3.5).

More importantly, it would also allow for experimentation with direct intervention, rather than waiting for students to request help. As discussed in Section 5.2.3, we saw very little use of the coach. We suggest this can be remedied through improved classroom pedagogy, but can also potentially be improved through interruptive feedback at crucial points, given that those crucial points are correctly identified. WOZ-style experimentation could provide the means to engage in such experimentation, allowing researchers to rapidly test different intervention techniques. This could potentially increase the rate of use the coach, and increase the rate at which we learn about the coach's abilities to influence student behavior.

5.3.2.2 Define the Facilitator Role in the Final System

Section 5.3.2.1 discusses the facilitator's role in the development process. However, future work should also consider the role of a facilitator using the finished system. As part of our general classroom pedagogy, we have defined the role of facilitators within our pedagogical framework. Specifically, at the start of the Rashi use, facilitators introduce the concept of inquiry, introduce the Rashi system and interface, and introduce the case at hand. During Rashi use, facilitators encourage the students to stay focused, and offer some broad inquiry advice, such as investigating multiple hypotheses and searching for both supporting and refuting evidence.

Moving beyond this type of generic classroom pedagogy external to the system, the facilitator's role in the Rashi system itself should be defined and implemented. Tools and data views should be made available to ease this role and to sharpen the facilitators' abilities to engage with students as deemed pedagogically appropriate. First steps could include providing basic statistical information about student work, as is available in many systems (Harrer et al., 2008; Kay et al., 2006). Beyond this basic quantitative information, we should consider the different higher-level analysis available, and decide if this should be given only to the facilitator, used only to create feedback for the students, or both. Tools similar to those necessary for the WOZ experiments (Section 5.3.2.1) could serve this purpose, providing a means for facilitators to have access to analysis information that is not provided to students.

Introducing a facilitator tool that provides specific analysis information not only supports facilitators in their roles, but also broadens the possibilities for using less-certain analysis information. Information involving a higher rate of uncertainty can be used as hints or clues to the facilitator, and double-checked before feedback is given. For instance, when the system recognizes a student's potential misunderstanding, the facilitator can be directly alerted to independently judge whether the student does in fact have a misunderstanding and feedback is necessary.

When facilitators receive information from the finished system and choose the interventions individually, we see the analysis system acting as somewhat like a recommender system for feedback. This occurs outside of a narrower, pre-defined, automated feedback system as described in Section 3.3. The facilitator's role can be to review analysis information and recommended feedback related to student work, and to decide the feedback to be given. This has been demonstrated in other ill-defined, collaborative problem spaces (Keshtkar et al., 2012), and we consider it a promising direction for future work in Rashi.

5.3.3 Focus on Higher-Order Skills

Our final recommended direction for future work is to focus upon the higher-order and meta-cognitive skills learned and practiced in the inquiry learning and collaborative processes. As discussed in Sections 2.3 and 3.3.3, the Rashi system teaches not only domain knowledge, but also teaches higher-order skills through inquiry. Additionally, the system has potential to teach higher-order skills about collaboration (sharing, help-seeking and help-giving, etc.) with the introduction of appropriate classroom pedagogy for collaboration (Section 5.3.1). These learning goals are the center of a large body of current research and play a central role in the conceptualization of 21st Century learning (Kellner 2002; Rotherham & Willingham 2010).

The first step in this direction is to better understand the current system's actual impact on higher-order learning skills. While these skills are challenging to measure (Clarke-Midura et al., 2011), a common approach is to monitor log traces of actions as students interact (with the system or with each other), and attempt to identify patterns that are indicative of the higher-order skills (Clarke-Midura & Dede, 2010). To implement this type of analysis, action logging must be implemented to leave a trace of all student actions as they interact with the system. These traces then need to be analyzed to identify specific patterns of behavior, either through simple rule-based approaches (Dragon et al., 2012), or with more complex machine-learning algorithms (e.g., clustering (Mondolnado et al., 2012), supervised learning (McLaren et al., 2010), etc.). From the identification of these log patterns, we gain information about how students are currently engaging in higher-order skills, and can estimate their uptake of these skills by recognizing change in behavior over time.

Considering lessons from Section 5.3.1, it is important to define a pedagogical model of the expected higher-order skills linked with specific expected behaviors within Rashi, and specific feedback or interventions designed to promote those skills. This pedagogical model can then be briefly introduced to students before and during their use of Rashi. Considering lessons from 5.3.2, WOZ studies can be carried out to discover how these behaviors might be recognized through analysis information, and also to study the effects of the given feedback, even before the analysis system is fully functional. In this way, we can learn the holistic impact of Rashi, including coaching and collaboration, on higher-order learning skills.

5.4 Summary

In this chapter, we presented a summary of our research, focusing on lessons learned and potential avenues for continuation. We discussed in detail the take-away messages about classroom pedagogy, collaboration, coaching, and their combination. We considered future research efforts in three distinct areas: defining classroom pedagogy, increasing human involvement in coaching, and focusing on higher order skills. Through the lessons learned we described the final results and conclusions of our work, and in so doing we define next steps to enhance or further our current results.

Our research demonstrates the positive effects of explicating and communicating classroom pedagogy. This concept should be taken into account in all future endeavors with ITSs, particularly those that bring different pedagogical approaches into classrooms (as our research introduced inquiry learning to students). We demonstrate the effect of a classroom pedagogy for use of the basic features of the system, but we encourage as future work the development of similar models for collaborative efforts, coaching, and content-focused collaboration. In each of these situations, students need clear definitions of how these features are useful to them, and specifically how the features should be used. A major source to consider formalizing these models is the CSCL research area of scripting, which provides such models solely for collaborative concepts.

Our research also demonstrates that collaboration is a useful learning mechanism when approaching ill-defined domains. Our studies show an improvement in student solutions when collaborative features are available. We also present our solution to one of the main issues in collaborative work: the difficulty for students to collaborate in context and with reference to specific domain content. We present the concepts of discussable objects and content-focused collaboration as a means to tighten the link between content and collaborative contributions. We discuss our approach to supporting this behavior by using the automated analysis system. We offer evidence that this automated analysis can identify opportunities to promote content-focused collaboration. We also recognize that these situations require explicit classroom pedagogy as discussed earlier.

In relation to coaching, our research does not present clear evidence that the possibility to receive coaching has a positive effect on student solutions. However, we argue that literature demonstrates in both theory and practice that such a coaching system is necessary and can be useful. We consider two major problems with our studies, the lack of classroom pedagogy and the lack of explicit human involvement in the process. We previously discussed the solution to the lack of classroom pedagogy.

Addressing the concern of a lack of human involvement in the process, we consider two different aspects to this concern. The first involves the facilitator's role in the development process. If we involve facilitators in the development (for example in WOZ studies), we can iteratively test and develop our software in a way that greatly increases our odds of success by shortening development and evaluation cycles. Beyond this role in the development cycle, we also need consider the final role of facilitators in the classroom pedagogy. We suggest that facilitators can take a more active and productive role if they are provided with specific tools in the system; particularly tools that provide access to analysis information about their students and feedback mechanisms to connect with their students. From this angle, we can also see that a single set of tools can be developed that serves both of these purposes, first supporting WOZ experimentation, and finally acting a means for facilitators to interact with students in the finished system.

Finally, we suggest that the evaluation of the Rashi system be lifted to consider the higher-order learning skills that Rashi theoretically improves. While the recognition or evaluation of these skills is difficult, we see this as an important step towards proving the importance and efficacy of the teaching methods that define Rashi. We suggest a close review of recent literature that analyzes traces of student behavior to identify key behaviors indicative of higher-order learning skills.

In summary, we present our research on a broad set of topics: ill-defined problem spaces, inquiry learning, collaboration, and intelligent systems. To exemplify our work and offer a test-bed for experimentation, we offer Rashi, an intelligent, collaborative, inquiry learning system for ill-defined domains. Within this system we have run studies over several years to demonstrate the effects of collaborative features, coaching features, and the combination of these approaches within an inquiry learning experience. We argue that both of these approaches are valid and interesting for inquiry learning in ill-defined problem spaces, and that combining these approaches could offer improvements to both aspects.

APPENDIX

GLOSSARY OF TERMS

While attempting to remain generically applicable and true to the default meanings of these terms, these definitions also reflect the specific context in which the terms are employed within this document.

Artificial Intelligence (AI): an active area of research in which machines are imbued with the capability to imitate intelligent human behavior.

Artificial Intelligence in Education (AIED): an active area of research in which artificial intelligence techniques are applied to computer-based instruction to understand and support students as an individual human tutor might in a one-on-one tutoring scenario.

Computer Based Instruction (CBI): the use of computer systems to teach or provide instruction.

Computer Supported Collaborative Learning (CSCL): the use of computers to provide tools and shared workspaces that enable multiple users to learn together.

Constraint Based Tutor (CBT): An ITS that offers guidance base on constraints (or rules) about the space of acceptable solutions, rather than requiring a specific solution or a specific solution path.

Content-focused collaboration: when collaborative contributions are centered on and revolve around specific domain content.

Discussable objects: elements of the user environment that can (and should) be the subject of dialog between learners.

Environment: the virtual workspaces available to a user that enable exploration, manipulation, and data collection.

Expert system: A specific type of Artificial Intelligence that attempts to mimic the reasoning of a human specialist.

Expert model: a semantic model of the knowledge of a specialist that can be inspected in order to understand the area of specialization. Often used as a part of an Expert System.

Expert Knowledge Base (EKB): a semantic representation of individual knowledge components and their inter-relations created using expert knowledge in a field, often created by technical developers working with Subject Matter Experts (SME).

Facilitator: any human who is acting in a coordinating, organizing, and help-giving role within a learning scenario, most often teachers or Teaching Assistants (TAs).

Free input: the un-restricted end of the input spectrum, representing data entry that enables users to submit any manner of input without structural limitations.

Ill-Defined Problem Space: a problem space that does not necessarily have clear start states, goal states, and/or intermediate states, or where the transition between these states is unclear.

Inquiry learning: A method of learning where students actively engage in solving realistic problems, by following certain basic steps: form hypotheses about the problem at hand, engage in data collection, and relate observable facts collected to support or refute the hypotheses.

Intelligent Tutoring System (ITS): CBI systems that can understand and adapt to the user by applying AIED techniques.

Knowledge Base: see *Expert Knowledge Base.*

Micro-script: a specific type of script that scaffolds students individual contributions, usually manifested as structure imposed on contributions (categories, sentence openers, etc.).

Natural Language Understanding (NLU): and active area of research in which artificial intelligence techniques are applied in order to enable a computer system to understand human utterances or dialog, whether written or spoken aloud.

Problem-Based Learning (PBL): a method of learning where students actively engage in learning activities through problem-solving, often used synonymously with inquiry learning.

Restricted input: the limited end of the input spectrum, representing data entry that only allows users to submit input or manipulate input in a certain structure and/or from some pre-specified set of inputs.

Script: a set of principles that designers may apply to help scaffold specific classes of interaction in learning scenarios. See also *micro-script*.

Subject matter expert (SME): a person who has authoritative knowledge on a certain domain, and shares this knowledge in order to create an expert model.

Student Solution: a representation of the student's accomplishment on a given problem. In the Rashi system, this consists of the contents of their Notebook (both argument editor and data table).

Wizard of Oz Study (WOZ): a specific study design in which a human facilitator is given some ability send feedback to students thorugh the standard interface for automated feedback, such that the student is to assume the system itself offers the feedback.

BIBLIOGRAPHY

Adamson, D., & Rosé, C. (2012). Coordinating multi-dimensional support in collaborative conversational agents. In *Intelligent Tutoring Systems* (pp. 346-351). Springer Berlin/Heidelberg.

Aleven, V. A., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science,* 26(2), 147-179.

Aleven, V.A., McLaren, B., Roll, I., & Koedinger, K. (2004). Toward tutoring help seeking. In *Intelligent Tutoring Systems* (pp. 227-239). Springer Berlin/Heidelberg.

Anastopoulou, S., Sharples, M., Ainsworth, S., Crook, C., O'Malley, C., & Wright, M. (2012). Creating personal meaning through technology-supported science inquiry learning across formal and informal settings. *International Journal of Science Education*, *34*(2), 251-273.

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, *42*(1), 7-49.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Ray. Pelletier. (1995). Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2), 167-207.

Anjewierden, A., Gijlers, H., Kolloffel, B., Saab, N., & De Hoog, R. (2011). Examining the relation between domain-related communication and collaborative inquiry learning. *Computers & Education*, *57*(2), 1741-1748.

Ashley, K.D & Goldin, I.M. (2011) "Toward AI-enhanced Computer-supported Peer Review in Legal Education." *Jurix 2011: The 24th Annual Conference*, pp. 1-12 (Katie M. Atkinson ed.) IOS Press 2011.

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*, 2(2-3), 159-190.

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why students engage in “gaming the system” behavior in interactive learning environments. *Journal of Interactive Learning Research*, *19*(2), 185-224.

Baerheim, A. (2001). The diagnostic process in general practice: has it a two-phase structure?. *Family practice*, *18*(3), 243-245.

Beck, J., Stern, M., & Haugsjaa, E. (1996). Applications of AI in Education. *Crossroads*, *3*(1), 11-15.

Beers, M. H., Porter, R., Jones, T., & et al, T. (2006). The merck manual of diagnosis and therapy. (18th ed. ed.). West Point: Merck Research Laboratories.

Bernhard, D., & Gurevych, I. (2008, June). Answering learners' questions by retrieving question paraphrases from social Q&A sites. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 44-52). Association for Computational Linguistics.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4-16.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *The journal of the learning sciences*, *2*(2), 141-178.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In *Intelligent Tutoring Systems* (pp. 261-269). Springer Berlin/Heidelberg.

Chaudhuri, S., Kumar, R., Howley, I., & Rosé, C. P. (2009, July). Engaging collaborative learners with helping agents. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling* (pp. 365-372). IOS Press.

Chen, W. (2006). Supporting teachers' intervention in collaborative knowledge building. *Journal of Network and Computer Applications*, *29*(2), 200-215.

Christodoulopoulos, C. E., & Papanikolaou, K. A. (2007, October). A group formation tool in an e-learning context. In *proceeding of 19th IEEE International Conference on Tools with Artificial Intelligence* (ICTAI 2007). Vol. 2, pp. 117-123. IEEE.

Choi, I., Land, S. M., & Turgeon, A. J. (2005). Scaffolding peer-questioning strategies to facilitate metacognition during online small group discussion. *Instructional science*, *33*(5), 483-511.

Clancey, W. J., & Letsinger, R. (1982). *NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching*. Department of Computer Science, Stanford University.

Clark, D., Weinberger, A., Jucks, R., Spitulnik, M., & Wallace, R. (2003). Designing effective science inquiry in text-based computer supported collaborative learning environments. *International Journal Of Educational Policy Research And Practice*, *4*(1), 55-82.

Clarke-Midura, J., & Dede, C. (2010). Assessment, Technology, and Change. *Journal of Research on Technology in Education*, *42*(3), 309-328.

Clarke-Midura, J., Dede, C., Norton, J. (2011). Next Generation Assessments for Measuring Complex Learning in Science. *The Road Ahead for State Assessments.* MA: Rennie Center for Education Research & Policy.

Collins, A., & Stevens, A. L. (1991). A cognitive theory of inquiry teaching. *Teaching knowledge and intelligent tutoring*, *203*.

Conati C., Vanlehn K. (2000). Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. *International Journal of Artificial Intelligence in Education,* 11, 398-415.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*(4), 253-278.

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intel- ligent tutoring systems. In M. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction (2nd ed.)* (pp. 849–874). Amsterdam: North-Holland.

Constantino-Gonzales, M. A., Suthers, D., & Escamilla, J. (2003). Coaching Web-based Collaborative Learning based on Problem Solution Differences and Participation. *International Journal of Artificial Intelligence in Education*, 13(2-4), 263-299.

Crespo, R., Pardo, A., Pérez, J., & Kloos, C. (2005). An algorithm for peer review matching using student profiles based on fuzzy classification and genetic algorithms. *Innovations in Applied Artificial Intelligence*, 111-115.

Crowley, R. S., & Medvedeva, O. (2006). An intelligent tutoring system for visual classification problem solving. *Artificial Intelligence in Medicine*, *36*(1), 85-117.

De Jong, T., Van Joolingen, W. R., Giemza, A., Girault, I., Hoppe, U., Kindermann, J., ... & Van Der Zanden, M. (2010). Learning by creating and exchanging objects: The SCY experience. *British Journal of Educational Technology*, *41*(6), 909-921.

Dillenbourg, P., Baker, M. J., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. *Learning in Humans and Machine: Towards an interdisciplinary learning science.*, 189-211.

Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning*, *3*(1), 5-23.

(2011). Dorland's illustrated medical dictionary. (29 ed.). W B Saunders Co.

Dragon, T., Floryan, M., Woolf, B., & Murray, T. (2010). Recognizing Dialogue Content in Student Collaborative Conversation. *Proceedings of the 10th International Conference for Intelligent Tutoring Systems, ITS 2010.* Pittsburgh: Springr-Verlag.

Dragon, T., Gutierrez-Santos, S., Mavrikis, M., & McLaren, B. M. (2011). Analysing and Supporting Students' Interactions in Synthesized Learning Environments: A Case Study with a Microworld and a Discussion Tool. *Preceedings of the Workshop on Technology Enhanced Learning for Math and Science (TELMAS), hosted in conjunction with the Sixth European Conference on Technology Enhanced Learning (EC-TEL 2011)* . Palermo.

Dragon, T., Mavrikis, M., McLaren, B. M., Harrer, A., Kynigos, C., Wegerif, R., & Yang, Y. (2013) Metafora: A Web-based Platform for Learning to Learn Together in Science and Mathematics. *IEEE Transactions on Learning Technologies. , 10 Jan. 2013. IEEE Computer Society Digital Library*. http://doi.ieeecomputer society.org/10.1109/TLT.2013.4

Dragon, T., McLaren, B.M., Mavrikis, M., & Geraniou, E. (2012). Scaffolding Collaborative Learning Opportunities: Integrating Microworld Use and Argumentation. *Advances in User Modeling: Selected papers from UMAP 2011 Workshops* (pp. 18-30). Girona: Springer-Verlag.

Dragon, T., Woolf, B.P. (2007). *Proceedings of the Workshop for Inquiry Learning of the 13th International Conference of Artificial Intelligence in Education*. AIED 2007. Marina Del Rey, CA (2007).

Dragon, T., Woolf, B., Marshall, D., & Murray, T. (2006). Coaching Within a Domain Independent Inquiry Environment. *Proceedings of the 8th International Conference for Intelligent Tutoring Systems, ITS 2006.* Jhongli: Springer-Verlag.

Dragon, T., Woolf, B., & Murray, T. (2009, July). Intelligent Coaching for Collaboration in Ill-Defined Domains. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling* (pp. 740-742). IOS Press.

Fiedler, A., & Gabsdil, M. (2002). Supporting progressive refinement of Wizard-of-Oz experiments. In *Proceedings of Intelligent Tutoring Systems* (pp. 62-69).

Fiedler, A., Gabsdil, M., Horacek, H. (2004) A tool for supporting progressive refinement of wizard-of-oz experiments in natural language. In Lester, J.C., Vicari, R.M., Paragua ̧cu, F., eds.: *Intelligent Tutoring Systems* — 7th International Conference (ITS 2004). Number 3220 in LNCS, Springer 325–335.

Floryan, M., Dragon, T., & Woolf, B. (2012). When less is more: focused pruning of knowledge bases to improve recognition of student conversation. In *Intelligent Tutoring Systems* (pp. 340-345). Springer Berlin/Heidelberg.

Furberg, A. (2010). Scientific Inquiry in Web-based Learning Environments. Exploring technological, epistemic and institutional aspects of students' meaning making. Doctoral thesis at InterMedia, University of Oslo. Oslo: Unipub.

Gamboa, H., & Fred, A. (2002). Designing intelligent tutoring systems: a bayesian approach. *Enterprise Information Systems III. Edited by J. Filipe, B. Sharp, and P. Miranda. Springer Verlag: New York*, 146-152.

Garcia, R. M. C., & Pardo, A. (2010). A supporting system for adaptive peer review based on learners' profiles. In *Proceedings of Computer Supported Peer Review in Education Workshop* (pp. 22-31).

Gauthier, G. (2010). *Capturing and Representing the Reasoning Processes of Expert Clinical Teachers for Case-Based Teaching* (Doctoral dissertation, Ph. D. diss. Dept. of Educational and Counselling Psychology, McGill univ., Montreal, Canada).

Gauthier, G., Lajoie, S. P., & Richard, S. (2007). Mapping and validating case specific cognitive models. Artificial Intelligence in Education*, Los Angeles, CA*.

Geier, R., Blumenfeld, P. C., Marx, R. W., Krajcik, J. S., Fishman, B., Soloway, E., & Clay-Chambers, J. (2008). Standardized test outcomes for students engaged in inquiry-based science curricula in the context of urban reform. *Journal of Research in Science Teaching*, *45*(8), 922-939.

Gokhale, A. A. (1995). Collaborative Learning Enhances Critical Thinking. *Journal of Technology Education*, *7*(1) 22-30.

Gutierrez-Santos, S., Mavrikis, M., & Magoulas, G. (2010). Layered development and evaluation for intelligent support in exploratory environments: the case of microworlds. In *Intelligent Tutoring Systems* (pp. 105-114). Springer Berlin/Heidelberg.

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Ablex Publishing.

Harrer, A., Ziebarth, S., Giemza, A., & Hoppe, U. (2008, July). A framework to support monitoring and moderation of e-discussions with heterogeneous discussion tools. In *Advanced Learning Technologies, 2008. ICALT'08.* Eighth IEEE International Conference on (pp. 41-45). IEEE.

Jameson, A. (1995). Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction*, *5*(3), 193-251.

Johnson, R. T., & Johnson, D. W. (1986). Action research: Cooperative learning in the science classroom. *Science and Children*, 24, 31-32.

Kabanza, F., Bisson, G., Charneau, A., & Jang, T. S. (2006). Implementing tutoring strategies into a patient simulator for clinical reasoning learning. *Artificial Intelligence in Medicine*, *38*(1), 79.

Kay, J., N. Maisonneuve, K. Yacef and P. Reimann (2006). The Big Five and Visualisations of Team Work Activity. *Proceedings of Intelligent Tutoring Systems (ITS06)*, M. Ikeda, K. D. Ashley & T-W. Chan (eds). Taiwan. Lecture Notes in Computer Science 4053, Springer-Verlag, 197-206.

Kazi, H., Haddawy, P., & Suebnukarn, S. (2009). Expanding the space of plausible solutions in a medical tutoring system for problem-based learning. *International Journal of Artificial Intelligence in Education*, *19*(3), 309-334.

Kellner, D. (2002). New media and new literacies: Reconstructing education for the new millennium. *The handbook of new media*, 90-104.

Keshtkar, F., Hatfield, D., Morgan, B., Wang, J., and Graesser, A. (2012) Introducing an Intelligent Virtual Suggester and Assessment in Computer Educational Games. *In the 11th International Conference on Intelligent Tutoring Systems* (ITS-2012), Workshop on Intelligent Support for Learning in Groups, Chania, Greeece.

Ketelhut, D. J. (2007). The impact of student self-efficacy on scientific inquiry skills: An exploratory investigation in River City, a multi-user virtual environment. *Journal of Science Education and Technology*, *16*(1), 99-111.

Kim, J. M., Hill, Jr, R. W., Durlach, P. J., Lane, H. C., Forbell, E., Core, M., Marsella, S., Pynadath, D., & Hart, J. (2009). BiLAT: A game-based environment for practicing negotiation in a cultural context. *International Journal of Artificial Intelligence in Education*, *19*(3), 289-308.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist*, 41(2), 75-86.

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with Cognitive Tutors. *Educational Psychology Review*, *19*(3), 239-264.

Koedinger, K.R., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as Cognitive Tools*. Hillsdale, NJ: Erlbaum. 15-46.

Koedinger, K. R., Corbett, A. T., Ritter, S., & Shapiro, L. (2000). Carnegie Learning's Cognitive Tutor : Summary Research Results. White paper. Available from Carnegie Learning Inc., 1200 Penn Avenue, Suite 150, Pittsburgh, PA 15222.

Kollar, I., Fischer, F., & Hesse, F. W. (2006). Collaboration scripts–a conceptual analysis. Educational Psychology Review, 18(2), 159-185.

Krajcik, J., Blumenfeld, P. C., Marx, R. W., Bass, K. M., Fredricks, J., & Soloway, E. (1998). Inquiry in project-based science classrooms: Initial attempts by middle school students. *Journal of the Learning Sciences*, *7*(3-4), 313-350.

Kumar, R., & Rose, C. P. (2011). Architecture for building conversational agents that support collaborative learning. *Learning Technologies, IEEE Transactions on*, *4*(1), 21-34.

Lajoie, S. P., Guerrera, C., Munsie, S. D., & Lavigne, N. C. (2001). Constructing knowledge in the context of BioWorld. *Instructional Science*, *29*(2), 155-186.

Land, S. M. (2000). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, *48*(3), 61-78.

Loll, F. (2012). Domain-Independent Support for Computer-Based Education of Argumentation Skills. PhD Thesis. Clausthal University of Technology, Department of Informatics.

Loll, F., Pinkwart, N., Scheuer, O., & McLaren, B.M. (2012). How Tough Should It Be? Simplifying the Development of Argumentation Systems using a Configurable Platform. In N. Pinkwart, & B. M. McLaren (Eds.), *Educational Technologies for Teaching Argumentation Skills*, Bentham Science Publishers.

Lund, K., & Baker, M. (1997). Promoting reflective interactions in a computer-supported collaborative learning environment. *Journal of computer assisted learning*, (13), 175-193.

Lynch, C., Ashley, K. D., Pinkwart, N., & Aleven, V. (2009). Concepts, structures, and goals: Redefining ill-definedness. *International Journal of Artificial Intelligence in Education*, *19*(3), 253-266.

Magnisalis, I., Demetriadis, S., & Karakostas, A. (2011). Adaptive and intelligent systems for collaborative learning support: A review of the field. *Learning Technologies, IEEE Transactions on*, *4*(1), 5-20.

Maulsby, D., Greenberg, S. and Mander, R. (1993) "Prototyping an intelligent agent through Wizard of Oz." In *ACM SIGCHI Conference on Human Factors in Computing Systems*, Amsterdam, The Netherlands, May, p277-284, ACM Press.

Mavrikis, M., Dragon, T., & McLaren, B.M. (2012). The design of wizard-of-oz studies to support students' learning to learn together. In the *Proceedings of the Workshop on Intelligent Support for Exploratory Environments 2012: Exploring, Collaborating and Learning Together at the 11th International Conference on Intelligent Tutoring Systems (ITS 2012)*.

Mavrikis, M., & Gutierrez-Santos, S. (2010). Not all wizards are from Oz: Iterative design of intelligent learning environments by communication capacity tapering. *Computers & education*, *54*(3), 641-651.

Mavrikis, M., Gutierrez-Santos, S., Geraniou, E., & Noss, R. (2012). Design requirements, student perception indicators and validation metrics for intelligent exploratory learning environments. *Personal and Ubiquitous Computing*, 1-16.

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*, 20(3), 194-204.

Mclaren, B. M., Scheuer, O., & Mikšátko, J. (2010). Supporting collaborative learning and e-discussions using artificial intelligence techniques. *International Journal of Artificial Intelligence in Education*, *20*(1), 1-46.

Minsky, M. (1995). Steps to artificial intelligence. In G. F. Luger (Ed.) *Computation & Intelligence: Collected Readings* (pp. 47–90). Menlo Park, California: AAAI/MIT Press.

Mitrovic, A. (1998). Experiences in implementing constraint-based modelling in SQL-Tutor. In B. P. Goettle, H. M. Halff, C. L. Refield & V. J. Shute, (Eds,) *Proceedings of the 4th International Conference on Intelligent Tutoring Systems, 1998* (pp. 414-423). London, UK: Springer-Verlag.

Mitrovic, A., Martin, B., Suraweera, P. (2007) Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. *IEEE Intelligent Systems,* 22(4), pp. 38-45. http://hdl.handle.net/10092/823

Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001). Constraint-based tutors: a success story. *Engineering of Intelligent Systems*, 931-940.

Mitrovic, A., & Weerasinghe, A. (2009). Revisiting Ill-Definedness and the Consequences for ITSs. *Artificial Intelligence in Education: Building Learning Systems That Care: from Knowledge Representation to Affective Modelling*, *200*, 375.

Morgan, B., Keshtkar, F., Duan, Y., Nash, P., & Graesser, A. (2012). Using state transition networks to analyze multi-party conversations in a serious game. In *Intelligent Tutoring Systems* (pp. 162-167). Springer Berlin/Heidelberg.

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*, *2*(2), 247-272.

Mulholland, P., Anastopoulou, S., Collins, T., Feisst, M., Gaved, M., Kerawalla, L., ... & Wright, M. (2012). nQuire: Technological support for personal inquiry learning. *IEEE Transactions on Learning Technologies*, *5*(2), 157-169.

Mühlpfordt, M., & Wessner, M. (2009). The integration of dual-interaction spaces. *Studying virtual math teams*, 281-293.

Murray, T. (1998). Authoring knowledge-based tutors: Tools for content, instructional strategy, student model, and interface design. *The journal of the learning sciences*, 7(1), 5-64.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education* (IJAIED), 10, 98-129.

Murray, T., Woolf, B., & Marshall, D. (2004). Lessons Learned from Authoring for Inquiry Learning: A tale of authoring tool evolution. In *Intelligent Tutoring Systems* (pp. 113-121). Springer Berlin/Heidelberg.

Newman, R. S. (1994). Adaptive help seeking: a strategy of self-regulated learning. Schunk, D. H. and Zimmerman, B. J., editors, *Self- regulation of learning and performance: Issues and educational applications,* Hillsdale, NJ: Erlbaum, 283–301.

Newman, R. S. (2000). Social influences on the development of children's adaptive help seeking: The role of parents, teachers, and peers. *Developmental Review*, *20*(3), 350-404.

Paolucci, M., Suthers, D., & Weiner, A. (1995, May). Belvedere: stimulating students' critical discussion. In *Conference companion on Human factors in computing systems* (pp. 123-124). ACM.

Paolucci, M., Suthers, D., & Weiner, A. (1996). Automated advice-giving strategies for scientific inquiry. In *Intelligent tutoring systems* (pp. 372-381). Springer Berlin/Heidelberg.

Pérez-Sanagustín, M., Burgos, J., Hernández-Leo, D., & Blat, J. (2009, November). Considering the intrinsic constraints for groups management of TAPPS and Jigsaw CLFPs. In *Intelligent Networking and Collaborative Systems, 2009. INCOS'09. International Conference on* (pp. 317-322). IEEE.

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2007). Evaluating legal argument instruction with graphical representations using largo. *Frontiers In Artificial Intelligence And Applications*, *158*, 101.

Pinkwart, N., Lynch, C., Ashley, K., & Aleven, V. (2008). Re-evaluating LARGO in the Classroom: Are Diagrams Better Than Text for Teaching Argumentation Skills?. *Intelligent Tutoring Systems* (pp. 90-100). Springer Berlin/Heidelberg.

Problem Space (2002). *McGraw-Hill Dictionary of Scientific and Technical Terms* (6th ed). McGraw-Hill. Retrieved November 29, 2012, from http://www.answers.com/topic/problem-space

Ravenscroft, A., McAlister, S., & Sagar, M. (2010). Digital dialogue games and InterLoc: Deep learning design for collaborative argumentation on the web. *Educational technologies for teaching argumentation skills. Bentham Science E-Books*.

Ravi, S., Kim, J., & Shaw, E. (2007, July). Mining On-line Discussions: Assessing, Technical Quality for Student Scaffolding and Classifying Messages for Participation Profiling. In *Educational Data Mining Workshop for the Conference of Artificial Intelligence in Education* (pp. 70-79).

Rotherham, A. J., & Willingham, D. T. (2010). “21st-Century” Skills. *American Educator*, 17.

Rummel, N., Spada, H., & Hauser, S. (2006, June). Learning to collaborate in a computer-mediated setting: Observing a model beats learning from being scripted. In *Proceedings of the 7th international conference on Learning sciences* (pp. 634-640). International Society of the Learning Sciences.

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

Sabourin, J., Rowe, J., Mott, B., & Lester, J. (2012). Exploring inquiry-based problem-solving strategies in game-based learning environments. In *Intelligent Tutoring Systems* (pp. 470-475). Springer Berlin/Heidelberg.

Sabourin, J., Rowe, J., Mott, B., & Lester, J. (2012a). Exploring inquiry-based problem-solving strategies in game-based learning environments. In *Intelligent Tutoring Systems* (pp. 470-475). Springer Berlin/Heidelberg.

Sabourin, J., Shores, L., Mott, B., & Lester, J. (2012b). Predicting student self-regulation strategies in game-based learning environments. In *Intelligent Tutoring Systems* (pp. 141-150). Springer Berlin/Heidelberg.

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

Scheuer, O., McLaren, B.M., Loll, F., & Pinkwart, N. (2009). An analysis and feedback infrastructure for argumentation learning systems. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (AIED-09), Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling. (pp. 629-631). IOS Press.

Scheuer, O., McLaren, B. M., Weinberger, A., & Niebuhr, S. (in press). Promoting critical, elaborative discussions through a collaboration script and argument maps. Accepted for publication in *Instructional Science*.

Self, J. A. (1990). Bypassing the intractable problem of student modelling. *Intelligent tutoring systems*: At the crossroads of artificial intelligence and education, 107-123.

Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C. C., & Cohen, S. N. (1975). Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system. *Computers and biomedical research*, *8*(4), 303-320.

Shute, V. J., & Glaser, R. (1990). A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, *1*(1), 51-77.

Singh, R., Saleem, M., Pradhan, P., Heffernan, C., Heffernan, N., Razzaq, L. & Dailey, M. (2011). Improving K-12 Homework with Computers. *AIED'11 Proceedings of the Artificial Intelligence in Education Conference*. Springer. 328-336.

Slavin, R. E. (1992). When and why does cooperative learning increase achievement? Theoretical and empirical perspectives. *Interaction in cooperative groups: The theoretical anatomy of group learning*, 145-173.

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

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(4), 261-290.

Stahl, G. (Ed.). (2009). *Studying virtual math teams* (Vol. 11). Springer.

Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 409-426). Cambridge, UK: Cambridge University Press.

Suebnukarn, S., & Haddawy, P. (2004). A collaborative intelligent tutoring system for medical problem-based learning. In *Proceedings of the 9th international conference on Intelligent user interfaces* (pp. 14-21). ACM.

Suebnukarn, S., & Haddawy, P. (2006). Modeling individual and collaborative problem-solving in medical problem-based learning. *User Modeling and User-Adapted Interaction*, *16*(3), 211-248.

Suthers, D. (1999a) Representational support for collaborative inquiry. In *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on System Sciences* (pp. 14-pp). IEEE.

Suthers, D. (1999b). Representational bias as guidance for learning interactions: A research agenda. *Artificial Intelligence in Education. Open learning environments: New computational technologies to support learning, exploration and collaboration*, 121-128.

Suthers, D. (1999c). Representational and advisory guidance for learning: Alternate roles for AI. In *Proceedings of the IASTED International Conference Artificial Intelligence and Soft Computing. Honolulu Hawai'i.*

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

Toth, E. E., Suthers, D. D., & Lesgold, A. M. (2002). Mapping to know: The effects of representational guidance and reflective assessment on scientific inquiry. *Science Education*, *86*(2), 264-286.

Topping, K. J. (2005). Trends in peer learning. *Educational psychology*, *25*(6), 631-645.

Van Joolingen, W. (1998). Cognitive tools for discovery learning. *International Journal of Artificial Intelligence in Education (IJAIED)*, *10*, 385-397.

Walker, E., Rummel, N., & Koedinger, K. (2008). To tutor the tutor: Adaptive domain support for peer tutoring. In *Intelligent Tutoring Systems* (pp. 626-635). Springer Berlin/Heidelberg.

Walker, E., Rummel, N., & Koedinger, K. R. (2009a, June). Beyond explicit feedback: new directions in adaptive collaborative learning support. In *Proceedings of the 9th international conference on Computer supported collaborative learning-Volume 1* (pp. 552-556). International Society of the Learning Sciences.

Walker, E., Rummel, N., & Koedinger, K. (2009b, July). Modeling helping behavior in an intelligent tutor for peer tutoring. In *Proceedings of the 14th international conference on artificial intelligence in education* (pp. 341-349).

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4), 5-23.

Wegerif, R., McLaren, B. M., Chamrada, M., Scheuer, O., Mansour, N., Mikšátko, J., & Williams, M. (2010). Exploring creative thinking in graphically mediated synchronous dialogues. *Computers & education*, *54*(3), 613-621.

White, B. Y., & Frederiksen, J. R. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and instruction*, *16*(1), 3-118.

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education (IJAIED)*, *10*, 151-182.

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.